Minimum-cost control of directed complex networks

Ding, Jie

2018

Ding, J. (2018). Minimum-cost control of directed complex networks. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/74223>

<https://doi.org/10.32657/10356/74223>

Minimum-cost Control of Directed Complex Networks

Jie Ding

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2018

Statement of Originality

I hereby certify that the intellectual content of this thesis is the product of my original research work and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Jie Ding

Acknowledgements

First and foremost, I would like to express my most sincere appreciation and heartiest respect to my supervisor, Prof. Changyun Wen, for his invaluable professional guidance, endless patience and encouragement throughout my Ph.D. study. Prof. Wen is an awesome supervisor and provides me the best role model to be a scientist and researcher. Not only the knowledge, but also the right attitude as well as enthusiasm towards research I learnt from him, would definitely benefit in my whole life.

I also would like to thank my co-supervisor, Assoc. Prof. Chin Seng Chua, who gives me invaluable suggestions and offers me the balance of research and freedom. In addition, I would like to express my special gratitude to Dr. Guoqi Li of Tsinghua University for his careful guidance and continuous support of my research work. He provides me many constructive suggestions and comments during our discussion. Without his support, my graduate life will be tough.

Furthermore, it is my pleasure to thank my amazing colleges in Internet of Things Laboratory: Dr. Jinming Xu, Dr. Le Zhang, Dr. Zhenghua Chen, Dr. Hongmin Zhou, Dr. Zhi Zhou, Dr. Zuo Bai, Dr. Ye Ren, Rui Zhao, Yong Zhang, Qingchang Zhu, Dongzhe Wang, Xuhao Li and Qingxu Ding for their companion and discussion. Thanks to all my friends, it makes my graduate study and life in NTU joyful and colorful.

Last but not least, I wish to thank my dear parents for their unconditional love and everlasting support. It is their unconditional support and encouragement that make this work possible.

To my dear parents

Abstract

Complex networks emerge in diverse areas, ranging from social to biological, economic and technological systems. Among these applications, controlling the complex networks is one of the most important problem, which can guarantee reliable and efficient operations of the network. Specifically, according to control theory, a dynamic network system is controllable if it can be driven from any initial state to any desired final state with appropriate external inputs. Existing works mainly focus on the problem of finding the minimum number of nodes connected with external input signals under different conditions such that the resulting network system is controllable. However, as we know when the Gramian matrix tend to be singular, the control cost of the network will be prohibitively large, resulting in the network is theoretically controllable but practically uncontrollable. Therefore, controlling the network with minimum cost has been an important problem to be solved.

In this dissertation, we first consider the problem of locating a key node set of a network, which is a subset of nodes connected with a given number of external control sources to achieve minimum control cost in controlling the network. To solve this problem, projected gradient method (PGM) and projected trust-region method (PTM) are proposed with established convergence property. We show that PGM is more computationally efficient than PTM. Projected gradient method extension (PGME) is also proposed to identify the key node set with the definition of importance index. Simulation results demonstrate satisfactory performance of the proposed algorithms. It is found that the importance index of a node is strongly correlated to occurrence rate of that node to be selected as a key node in Monte-Carlo realizations. We also point out that key nodes tends to divide elementary topologies equally when the control cost reaches its minimum and low-degree nodes are more likely to be selected as key nodes.

Additionally, we further improve the results of key node selection based on degree distribution and proposed an algorithm which outperforms PGME. We first reformulate the minimum-control-cost model in Boolean constraints. By relaxing these

constraints into their convex hull, we introduce inexact alternating direction method of multipliers (IADMM). With the help of degree distribution, an extension algorithm, named degree-based IADMM (D-IADMM), is proposed such that key nodes can be pinpointed. We further improve the performance of D-IADMM by employing a simple greedy algorithm named local optimization. The proposed algorithms are validated by numerical examples ranging from Erdős-Rényi networks, scale-free networks to some real-life networks.

Moreover, we investigate the implicit linear quadratic regulation problem where a feasibility is provided to minimize the control cost with regards to input and state. Specifically, a quadratic term is added to the objective function such that the deviation of transient state is penalized. To obtain the optimal input strategy, orthonormal-constraint-based projected gradient method (OPGM) is proposed. The convergence of OPGM is established theoretically. The effectiveness and validity of OPGM is examined through extensive simulations. The variations of inputs and system states are observed under different cost functions.

The last significant issue we deal with is to design the network topology with given external control sources such that the control cost is minimum. A matrix function optimization model is proposed to study how the network topology evolves when the objective is to achieve an optimal control of the networks. By obtaining the direction of network topology evolution, a normalized and projected gradient-decent method (NPGM) based on the obtained gradient of the network topology evolution is developed to solve the proposed optimization problem. It is proven that NPGM linearly converges to a local minimum point. We also derive an optimality condition to determine whether a converged solution is global minimum or not, and such a condition is verified through numerous experimental tests on directed networks. We find that a network adaptively changes its topology in such a way that many sub-networks are gradually evolved towards a pre-established control target. Such a finding enables us to model and explain how real-world complex networks adaptively self-organize themselves to many similar subnetworks during a relatively long evolution process. We further extend the model to another case, i.e. to investigate how connection strengths between nodes vary in accordance with control cost for complex networks with fixed topological structure. We obtain the optimal connection strength matrix with the proposed NPGM. It is discovered that several control-flow subnetworks are self-formed. Moreover, the control cost with

optimal weight matrix is smallest when pre-located controller sources distribute evenly. These observations provide a comprehensive understanding of the impact of connection link weight on control cost for complex networks.

Contents

Acknowledgements	i
Abstract	v
List of Figures	xiii
Symbols and Acronyms	xvii
1 Introduction	1
1.1 Motivation and Overview	1
1.2 Our Contribution	4
1.3 Outline of This Thesis	6
2 Background	7
2.1 Controllability of Complex Networks	7
2.1.1 Linear time-invariant system	7
2.1.2 Controllability	8
2.1.3 Structural controllability	9
2.1.4 Minimum input theorem	11
2.2 Energy Cost in Controlling Complex Networks	12
2.2.1 Distribution of eigen-energies	14
2.2.2 Energy-related metrics	15
2.3 Network Models	16
2.3.1 Erdős-Rényi networks	17
2.3.2 Scale-free networks	17
2.4 Graph Theory	19
2.5 Matrix Derivatives	19
3 Minimum-cost Control of Complex Networks Based on Matrix Differentiation	23
3.1 Introduction	24
3.2 Problem Formulation	26
3.3 First and Second order derivatives of $\mathcal{E}(B)$	28
3.4 Algorithm Design	29

3.4.1	Projected gradient method (PGM)	29
3.4.2	Projected trust-region method (PTM)	30
3.5	Convergence Analysis	31
3.6	Simulation Results	32
3.6.1	PGME in three elementary topologies	34
3.6.2	PGME in Erdős-Rényi and scale-free networks	38
3.6.3	Comparisons between PGME and random allocation method (RAM)	42
3.6.4	Strength-based random allocation method (S-RAM)	44
3.6.5	Robustness test of PGME	44
3.7	Conclusion	46
4	Key Node Selection in Directed Complex Networks via Convex Optimization	47
4.1	Introduction	48
4.2	Problem Formulation	49
4.3	Convex Relaxation for Key Nodes Selection	50
4.3.1	Inexact alternating direction multipliers of method	50
4.3.2	Convergence analysis of IADMM	56
4.4	Determination of Key Nodes	61
4.4.1	Key node selection based on degree distribution	61
4.4.2	Local optimization	62
4.5	Simulation Results	64
4.6	Conclusion	67
5	Implicit Linear Quadratic Regulation for Directed Complex Networks	69
5.1	Introduction	70
5.2	Problem Formulation	72
5.3	OPGM for Solving Optimization Problem	78
5.4	Convergence Proof	81
5.5	Simulation Examples	83
5.5.1	Effect of α in ImLQR	83
5.5.2	ImLQR in Erdős-Rényi and scale-free networks	85
5.5.3	Key nodes selection in directed networks	87
5.5.4	Discussions	89
5.6	Conclusion	90
6	Controlling Directed Networks with Evolving Topologies	91
6.1	Introduction	92
6.2	Problem Formulation	94
6.3	Normalized and Projected Gradient-decent Method	96
6.3.1	The derivative of the objective function with respect to network topology	97

6.3.2	Algorithm	100
6.4	Convergence Analysis	101
6.5	Optimality Condition	104
6.6	Simulation Results	111
6.6.1	Convergence illustration of NPGM	111
6.6.2	Effective rank of the network topology matrix A^*	112
6.6.3	Control cost when N is a multiple of M	114
6.6.4	Control cost when N is not a multiple of M	116
6.6.5	Application to social networks	117
6.6.6	Robustness test of NPGM	118
6.7	Extension NPGM on Networks with Fixed Structure	119
6.7.1	Clustering the network into M control-flow subnetworks	120
6.7.2	Control cost with different key nodes	122
6.8	Conclusions and Discussions	123
7	Conclusions and Future Works	125
7.1	Conclusion	125
7.2	Recommendations for Future Works	126
7.2.1	Sparsity-promoting topology design for directed complex networks	127
7.2.2	Closed-loop optimal control of complex networks	127
7.2.3	Alternative control of complex networks	128
7.2.4	Distributed optimal control of large-scale complex networks	128
A	Proofs for Chapter 3	129
A.1	Proof of Lemma 3.1	129
A.2	Proof of Lemma 3.2	129
A.3	Proof of Lemma 3.3	130
A.4	Proof of Lemma 3.4	131
A.5	Proof of Theorem 3.1	131
A.6	Proof of Theorem 3.2	131
A.7	Proof of Theorem 3.3	133
B	Proofs for Chapter 5	135
B.1	Proof of Lemma 5.1	135
B.2	Proof of Lemma 5.2	135
B.3	Proof of Theorem 5.1	137
C	Proofs for Chapter 6	143
C.1	Proof of Lemma 6.3	143
	Author's Publications	147
	Bibliography	149

List of Figures

1.1	Minimum-cost control problems addressed in this thesis compared with LQR. (a) The traditional LQR problem in which A and B are pre-given. We just need calculate the input to achieve the pre-defined objectives. (b) The case with A pre-given. The input matrix B and input $\mathbf{u}(t)$ are designed by our proposed algorithms so as to minimize the control cost. (c) The case with B pre-given. The topology A is obtained by minimizing the control cost with the proposed algorithm NPGM.	3
2.1	Controlling a simple graph	9
2.2	Illustration for structural controllability	9
2.3	Networks with maximum matching. Matched edges/edges are shown in Green.	12
2.4	(a) ER network. (b) Degree distribution of ER network	17
2.5	(a) SF network. (b) Degree distribution of SF network	18
2.6	A network consisting of three elementary topologies with 9 external control sources. The red points are connected with external inputs such that control cost is minimum.	19
3.1	Key nodes in three elementary topologies.	35
3.2	Elementary dilations. (a) An elementary dilation with $N = 8, M = 4$. (b) An elementary dilation with $N = 100, M = 25$. Red points are key nodes with minimum control cost.	37
3.3	Importance and occurrence rate distribution of nodes in an elementary stem with $N = 50, M = 10$ based on 1000 independent simulations. The dotted lines present envelopes.	38
3.4	Importance and occurrence rate distribution of nodes in an elementary cycle with $N = 100, M = 25$ based on 1000 independent simulations.	39
3.5	Importance and occurrence rate distribution of key nodes in an elementary dilation with $N = 100, M = 25$ based on 1000 independent simulations. The dotted lines present envelopes.	40
3.6	The correlation between importance index r and occurrence rate p with different n	40
3.7	Key nodes (red color) located in an ER network.	41

3.8	The comparisons between PGME and random allocation method in both synthetic and real-life networks. (a) Elementary stem with $N = 100$. (b) Real-life electronic network	43
3.9	The comparisons between PGME and RAM on SF and ER networks. (a) SF networks $N = 100$, $M = 40$. (b) ER networks with $N = 100$	43
3.10	The robustness test when designing B with A pre-given. Key nodes are marked by red color.	45
4.1	Ω -pattern.	62
4.2	Control cost versus number of key nodes for directed ER and SF networks with $N = 100$, $\langle k \rangle = 7.5$. (a) ER network (b) SF network	65
4.3	Comparison between proposed algorithms for directed networks with different mean degree. (a) Directed ER network with $N = 50$, $M = 20$. (b) Directed SF network with $N = 50$, $M = 20$	65
4.4	In-degree distribution of key nodes in directed ER and SF networks with $N = 50$, $M = 15$, $\langle k \rangle = 5$. (a) ER network (b) SF network	66
5.1	Illustration of the optimization problem. The notations in this figure are given in (5.13)-(5.15) and Theorem 5.1.	78
5.2	The illustration for convergence of OPGM with different α. (a) Control cost with iterations when $\alpha = 0.01$. (b) Control cost with iterations when $\alpha = 0.5$. (c) Control cost with iterations when $\alpha = 1$	85
5.3	System inputs and states when $\alpha = 0.5$. (a) The comparison between system inputs $\mathbf{u}(t, B^*)$ with their initials $\mathbf{u}(t, B_0)$. (b) The comparison between system states $\mathbf{x}(t, B^*)$ with their initials $\mathbf{x}(t, B_0)$	85
5.4	The effect of α on designed inputs and network states. (a) The comparison of system inputs $\mathbf{u}(t, B^*)$ with different α . (b) The comparison of system states $\mathbf{x}(t, B^*)$ with different α	86
5.5	Control costs for different values of α. (a) $M = 1, N = 3$ (b) $M = 2, N = 4$	86
5.6	Key nodes in a directed stem. The key nodes found by OPGM equally divide the stem to achieve the minimum control cost. The highlighted node connected to an external source (“star”) is a key node.	88
5.7	Key nodes in a directed circle. The highlighted node connected to an external source (“star”) is a key node. Similar to the results observed in a stem, the key nodes found by OPGM also equally divide the circle. The only difference is that the key node set that equally divide the circle is not unique. For example, both sets $\{1, 6, 11, 16, 21, 26\}$ and $\{5, 10, 15, 20, 25, 30\}$ can be located as key node set when implementing OPGM.	88

5.8	Illustration of the selected key nodes in a small random network. (selected key nodes are indicated by highlighted nodes). .	89
6.1	Control cost when $M = 3$ and $N = 15$	112
6.2	Control cost when $M = 3$ and $N = 14$	112
6.3	Control cost when $\frac{N}{M}$ is an integer. By taking $M = 3$, $M = 4$ and $M = 5$ as examples, the minimum cost is computed by $\frac{M}{\lambda_i^{\frac{N}{M}}}$ and NPGM respectively.	115
6.4	Control cost computed by NPGM with $\frac{N}{M}$ fixed.	116
6.5	Control cost when $\frac{N}{M}$ is not an integer with $M = 3$. The bottom of the rectangle is the lower bound of the minimum energy cost while the top is its upper bound.	117
6.6	The “optimal” topology for cooperation in social networks.	118
6.7	The robustness test of NPGM.	119
6.8	Change of interaction strengths when the control sources are pre-allocated.	121
6.9	Interaction strength changes with iterations for network. A_{ij} is weight of the connection from node j to node i . (a) Elementary stem with $N = 6$, $M = 2$. Nodes $\{1, 3\}$ are connected to external controller sources. (b) Elementary cycle with $N = 6$, $M = 2$. Nodes $\{1, 3\}$ are connected to external controller sources.	121

Symbols and Acronyms

Algebraic Operators

X^T	Transpose of matrix X
X^{-1}	Inverse of matrix X
$\det(X)$	Determinant of matrix X
$\text{tr}(X)$	Trace of matrix X
$\text{diag}(\mathbf{x})$	Diagonal matrix with diagonal entries $x_1, \dots, x_n$
$\ \cdot\ $	ℓ_2 norm for vector or the induced norm for matrix
$\ \cdot\ _F$	Frobenius norm for matrix
$\langle \cdot, \cdot \rangle$	Inner product of two vectors

Symbols

$\mathcal{G}$	Graph
$\mathbb{R}^n$	n -dimensional Euclidean space
$\mathbb{1}$	All-ones column vector
$\mathbf{x}$	State of the dynamical system
$\mathbf{u}$	Input of the dynamical system
A	Adjacency matrix of the networked system
B	Input matrix of the networked system
$E(\cdot)$	Position-dependent control cost
$\mathcal{E}(\cdot)$	Average control cost

Acronyms

LCC	Longest control chain
-----	-----------------------

LTI	Linear time-invariant
MCP	Minimum controllability problem
IADMM	Inexact alternating direction multipliers of method
D-IADMM	Degree-based inexact alternating direction multipliers of method
LD-IADMM	D-IADMM with local optimization
NPGM	Normalized and projected gradient-decent method
OPGM	Orthonormal-condition-based projected gradient method
PGM	Projected gradient method
PGME	Projected gradient method
PTM	Projected trust-region method
RAM	Random allocation method

Chapter 1

Introduction

1.1 Motivation and Overview

In real life, many complex systems can be modeled as networks, including social networks [1, 2], sensor networks [3, 4], smart grid [5, 6], economic networks [7, 8], gene networks [9, 10], brain networks [11, 12], transportation networks [13, 14] and so on. Each network is a unit composed of nodes and edges, collectively exhibiting complex behaviours. The nodes in the network represent individual elements (e.g., genes, sensors, neurons) and the edges indicate connections or interactions among nodes. As a networked system evolves with time, the state of a node (e.g., protein being expressed, position of a sensor, activity level of neurons) normally depends not only on the node's intrinsic dynamics but also on the couplings with its nearest neighbors.

Naturally, a networked system is expected to achieve some ideal situation during the evolution. For example, in transportation networks, it is hoped that there is no traffic congestion and the throughput is maximized [15]. A distributed generator is required to coordinate with other existing power resources to improve the energy efficiency in smart grid [16]. Virus spreading needs be controlled by human when there is an epidemic outbreak due to highly dangerous viruses [17]. In fact, whether these objectives in above applications can be achieved firstly relies on the controllability of complex networks. In control theory, if a dynamical system can be steered from any initial state to any desired final state with a suitable choice of inputs, the system is controllable. Due to the couplings between nodes, it is actually

difficult to choose such suitable inputs. In 2011, a prominent work is presented in [18], to identify the minimum number of driver nodes/inputs that ensure the controllability of the network. The authors propose maximum matching algorithm based on the concept of structural controllability [19] by ignoring the weights of coupling between nodes. It is shown that the minimum number of driver nodes required for complete control mainly depends on network's degree distribution. And it only requires a few of driver nodes to control dense and homogeneous networks while sparse inhomogeneous networks are much difficult to control. Different from [18], in [20], the minimal controllability problem (MCP) of controlling the network with minimum state variables, i.e. actuated nodes is addressed. Shortly after [20], many extensions about MCP have emerged, see examples [21–24] and references therein. However, there is no consideration of cost in all of the aforementioned works, while their solutions assure controllability or structural controllability of the system. Actually when the Gramian controllability matrix is nearly singular, the control cost becomes excessively high, making it impossible to fully control the whole network. In addition, it is arguably even more important to evaluate the energy cost in controlling real-life networked systems due to limited and valuable resources. For example, in electronic networks, it may be important to minimize the energy consumed by all the integrated circuit chips to achieve a given control objective [25]. In social networks, we may want to identify how easy/difficult to change people's opinions or behaviors [26]. Motivated by this, in this dissertation, we mainly focus on the problem of controlling the complex networks from energy perspective, so as to minimize the control cost.

The minimum-cost control problem is widely considered in optimal control, dealing with the problem of finding a control strategy for a dynamical system such that a given optimality criterion is achieved. A special case of the optimal control problem is the linear quadratic (LQ) problem [27], which aims to minimize a quadratic continuous-time cost functional subject to a linear first-order dynamic constraints $\dot{x}(t) = Ax(t) + Bu(t)$. Here A is the directed network topology matrix describing the connection/interaction strengths between N nodes, and B is an $N \times M$ input matrix describing how M external control sources connect with each node. When A and B are pre-given [27–31], the solution of LQ problem is provided by linear-quadratic regulator (LQR) through solving the continuous time Riccati differential equation [32–36].

However, there are situations that we have freedom to choose the topology A or input matrix B . In this case, we may need to address the following issues. Which inputs should be injected to the networked system? Under optimal control scheme, how can we design input matrix when we only have the information of topology and what characteristics does the topology matrix have when the input matrix is pre-given? To this end, in this thesis we mainly study these two basic issues with freedom of choosing input matrix B or topology matrix A , as illustrated in Fig. 1.1.

FIGURE 1.1: **Minimum-cost control problems addressed in this thesis compared with LQR.** (a) The traditional LQR problem in which A and B are pre-given. We just need calculate the input to achieve the pre-defined objectives. (b) The case with A pre-given. The input matrix B and input $\mathbf{u}(t)$ are designed by our proposed algorithms so as to minimize the control cost. (c) The case with B pre-given. The topology A is obtained by minimizing the control cost with the proposed algorithm NPGM.

In the first case, the input matrix B is selectable and becomes a variable of a pre-defined cost function. One needs to determine whether a node is connected to an external control source and what the exact connection weight is if connected, through an optimal way of minimizing an appropriate cost function. In [37], Yan et al. start to take control energy into account but only by analyzing a single external input which is connected with one node [37]. Other interesting heuristic

results are available related to control cost in existing works including [38–40] by exploring the eigenvalue properties of the controllability Gramian matrix. It is shown that the control energy decays exponentially when the number of control nodes increases [38, 40]. However, to the best of our knowledge, this important yet challenging problem remains largely as an open issue. This gives an promising future work to analyze how the locations of input signals (input matrix B) affect the cost of controlling directed networks.

On the other hand, different from existing works, we instead consider the following scenario: if network topology A becomes changeable, i.e., A is matrix variable to be optimized, what kind of role does the changing topology play for achieving minimum cost control of directed networks. This problem is significant since the topologies of various real-life networks are indeed adaptively changing from time to time so as to minimize certain cost or maximize certain criteria [41–44]. For example, in [45], how large-scale transportation networks evolve so that a maximum transportation potential can be achieved is investigated. In the work [46], it is pointed out that small-world networks are adaptively formed due to its economical properties of tending to minimize wiring costs while supporting high dynamical complexity. Since the conservation of wiring costs is likely to be an important selection pressure on the evolution of brain networks [47], the evolution of small-world networks in the brain is essentially to minimize a selection pressure to deliver cost-effective information-processing systems. The work in [48] indicates that cellular networks as a whole are scale-free networks due to the evolutionary selection of a robust and error-tolerant architecture. Also, this evolutionary selection may characterize all cellular networks, for which scale-free topology with a conserved network diameter appears to provide an optimal structural organization [49]. Moreover, given a structural controllable network, what is the optimal connection strength when the control energy is minimized? It is interesting to explore these characteristics of topology A from the perspective of optimal control theory.

1.2 Our Contribution

The objectives are to solve the problems motivated in Section 1.1. The main contributions made in this thesis are summarized as follows.

1. In Chapter 3, we consider the problem of selecting a subset of nodes connected with a given number of external control sources to drive the network at minimum control cost. To solve this problem, two algorithms, named projected gradient method (PGM) and projected trust-region method (PTM) are proposed with established convergence property. We also show that PGM is of lower complexity compared with PTM. In addition, with the definition of importance index, PGM extension (PGME) in Monte-Carlo scenario is introduced to determine key node set. Through extensive simulations ranging from synthetic networks to real-life networks, it is found that 1) importance index of a node is strongly correlated to the occurrence rate of that node to be selected as a key node in Monte-Carlo realizations. 2) key nodes tend to distribute as evenly as possible in elementary stems/cycles/dilations when control cost attains minimum. 3) low-degree nodes are of higher probability to be selected as key nodes.
2. In Chapter 4, to identify key nodes more effectively, we further improve the results obtained in last chapter and propose a new algorithm through convex optimization based on nodes' degree distribution. We reformulate the minimum-control-cost model in Boolean constraints. By relaxing these constraints into their convex hull, inexact alternating direction method of multipliers (IADMM) is developed and convergence is analyzed. Through exploiting the information of nodes' degree, an extension algorithm, name Degree-based IADMM (D-IADMM), is proposed to identify the key node set. With the technique of local optimization, the performance of D-IADMM is greatly improved. To validate the proposed algorithm, simulations are performed on different types of networks ranging from Erdős-Rényi networks, scale-free networks to some real-life networks.
3. In Chapter 5, an implicit linear quadratic regulator (ImLQR) is developed which enables us design the input as well as input matrix. Specifically, similar to linear quadratic problem, we add a quadratic term that penalizes deviation of transient state to the input energy function. To address this issue, an algorithm called orthonormal-constraint-based projected gradient method (OPGM) is proposed with established convergence. Through extensive simulation results, the effectiveness of the proposed algorithm is demonstrated.

4. In Chapter 6, we start to investigate the role of topology that plays on control cost with input matrix B is given. Normalized and projected gradient-decent method(NPGM) based on the obtained gradient of the network topology evolution is developed to solve such a problem. We also establish the convergence property and convergence rate of NPGM. The optimality condition is derived since only local minimum can be achieved by NPGM. Through comprehensive simulations on directed networks, we uncover that a network adaptively changes its topology such that many sub-networks are gradually evolved under the optimal control scheme. In addition, we extend our idea on the fixed structural networks in order to explore the influence of connection strength.

1.3 Outline of This Thesis

Chapter 1 introduces the overview and the potential practical applications in terms of controllability of complex networks, which essentially motivates the problems studied in this thesis. The main contributions and the outline of this thesis are also summarized. In Chapter 2, recent advances in controlling complex networks are reviewed and some preliminaries about graph theory and matrix derivatives are given. In Chapter 3, we start to consider the problem of selecting a subset of nodes that ensure the controllability with minimum control cost. Two algorithms based on matrix differentiation are proposed. A scheme of determining key node set under the definition of importance index is introduced. In Chapter 4, we further improve the selection of key nodes for directed complex networks via convex optimization by exploiting the degree information. In Chapter 5, an orthonormal-condition-based algorithm is proposed to address the implicit linear quadratic regulation problem where a quadratic term of penalization of transient state deviation is added to the objective function. The role of topology of directed network from optimal control perspective is investigated in Chapter 6, where we uncover that several subnetworks are gradually self-formed. Final conclusions and recommendations for future works are given in Chapter 7.

Chapter 2

Background

In this chapter, we mainly review the recent prominent works with regards to controllability of complex works and related energy scale. Two typical network models are introduced. In addition, some preliminaries about graph theory and matrix derivatives are also presented in this chapter.

2.1 Controllability of Complex Networks

In the literature of complex network controllability problem, most studies utilize LTI system to describe real-life networks [18]. This is because linear controllability problem not only provides a general framework to study the non-linear dynamics in controlling complex networks, but also reflects some intrinsic characteristics to further understand the non-linear dynamics. In addition, through linearization, it is possible to deal with some non-linear dynamics in the systems. Therefore, understanding the effect of linear dynamics on controlling complex networks is a prerequisite before proceeding to the complicated non-linear dynamics.

2.1.1 Linear time-invariant system

In this thesis, we adopt linear time-invariant (LTI) dynamics [50] to model complex networks described by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.1)$$

where $\mathbf{x}(t) = [x_i(t)]_{N \times 1}$ is the state vector of network nodes capturing the nodes' dynamics at time t , such as people's opinion in social networks, the gene expression in the gene regulation networks, etc. $\mathbf{u}(t) = [u_i(t)]_{M \times 1}^T$ is a time dependent input vector of external signals with M ($M \leq N$) being the number of control sources where $u_i(t)$ may drive multiple nodes. The element A_{ij} of the $N \times N$ network (weighted) adjacent matrix A describes the connection/interaction (strength/weight) from node j to node i , and the element B_{im} of the $N \times M$ input matrix B is nonzero if external control source m is connected to node i and zero otherwise.

2.1.2 Controllability

Consider the LTI system in (2.1) where A and B are constant matrices. The definition of controllability is stated as follows.

Definition 2.1 (Controllability [50, 51]). The linear state equation (2.1) or a pair (A, B) is said to be controllable if for any initial state $\mathbf{x}(t_0) = \mathbf{x}_0$ and any final state $\mathbf{x}(t_f) = \mathbf{x}_f$, there exists a continuous input signal $\mathbf{u}(t)$ that drives the system from its initial state $\mathbf{x}_0$ to final state $\mathbf{x}_f$ in a finite time. Otherwise, (2.1) or (A, B) is said to be uncontrollable.

To test the controllability of system (A, B) , the following theorem is given.

Theorem 2.1. [51] Given a LTI system, the following statements are equivalent

1. The system or (A, B) are controllable.
2. The matrix $W = \int_0^t e^{A\tau} B B^T e^{A^T \tau} d\tau$ is positive definite for any $t > 0$.
3. The $N \times NM$ controllability matrix $\mathcal{C} = [A, AB, A^2B, \dots, A^{N-1}B]$ has full rank, i.e. $\text{rank}(\mathcal{C}) = N$.
4. The $N \times (N + M)$ matrix $[A - \lambda I, B]$ has full row rank at every eigenvalue λ of A .

When the system is controllable, the state of the system at time t is given by

$$\mathbf{x}(t) = e^{At} \mathbf{x}(0) + \int_0^t e^{A(t-\tau)} B \mathbf{u}(\tau) d\tau \quad (2.2)$$

FIGURE 2.1: Controlling a simple graph

FIGURE 2.2: Illustration for structural controllability

2.1.3 Structural controllability

Given a LTI system (A, B) , if the elements of A and B are either zeros or non-zero independent parameters, (A, B) is considered as a structured system. For example, in Fig. 2.1, as $a_{12}, a_{13}, b_{11}, b_{23}$ are free independent parameters, system (A, B) is a structured system. This concept is consistent with the fact that in real-life networks we can only know the adjacent relation between components of system but cannot know the exact values of such free independent parameters. A system is structurally controllable if it is controllable when the parameters are fixed in certain values [18, 19]. If a system is controllable for all parameters (other than zero), the system is strongly structurally controllable.

To clarify this, we list four examples in Fig. 2.2. In (a), the system is described as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 0 & 0 \\ a_{12} & 0 & 0 \\ 0 & a_{23} & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} b_{11} \\ 0 \\ 0 \end{bmatrix} \mathbf{u}(t)$$

Then the controllability matrix is calculated as

$$\mathcal{C} = b_{11} \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{12} & 0 \\ 0 & a_{23} & a_{12}a_{23} \end{bmatrix}$$

Obviously, $\text{rank}(\mathcal{C}) = 3 = N$. Therefore, the system is strongly structural controllable as long as a_{12} , a_{23} , b_{11} are non-zero. That is to say, strongly structural controllability is independent of the choice of parameters in the system.

In Fig. 2.2 (b),

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_{13} & 0 & 0 \end{bmatrix}$$

It is easy to obtain $\text{rank}(\mathcal{C}) = 1 < N$. Thus the system is not controllable.

In Fig. 2.2 (c), $\text{rank}(\mathcal{C}) = 3 = N$ as long as the parameters are non-zero values. The system is strongly structurally controllable.

In Fig. 2.2 (d),

$$\mathcal{C} = b_{11} \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{12} & a_{13}a_{32} \\ 0 & a_{13} & a_{12}a_{23} \end{bmatrix}$$

If $a_{12}^2 a_{23} \neq a_{13}^2 a_{32}$, $\text{rank}(\mathcal{C}) = 3 = N$. Otherwise, $\text{rank}(\mathcal{C}) = 2 < N$. That is to say, in most cases, the system is controllable. Such a system is called structurally controllable, but not strongly structurally controllable.

In addition, the following theorem in terms of structural controllability in a digraph is given by Lin in 1967 [19].

Theorem 2.2 (Lin's Structural Controllability Theorem [18, 19]). For a LTI system, the following three statements are equivalent:

1. The pair of (A, B) is structural controllable.
2. The digraph (A, B) contains no inaccessible nodes and no dilations.
3. The digraph (A, B) is spanned by a cactus.

2.1.4 Minimum input theorem

In work [18], Liu et al. derive a method to calculate the minimum number of external inputs that ensure the structural controllability of the network. By bridging the relation between the structural controllability and maximum matching method, the nodes connected with external inputs are pinpointed. Given a digraph $\mathcal{G}$, if no two edges in an edge subset $\mathcal{M}$ share a common starting or ending vertex, such a subset $\mathcal{M}$ is called a matching. A node is matched if a matched edge is pinpointed to it. Otherwise, the node is unmatched. Maximum matching is the matching set with the largest number of matched edges. If all vertices are matched, it is a perfect matching. Fig. 2.3 illustrates the maximum matching of graphs (a) (b) and (c), respectively. Among them, (b) is a perfect matching. To ensure the system is structural controllable, unmatched nodes are required to be connected with external inputs. Those controlled nodes that do not share input vertices are called driver nodes. To identify the minimum driver nodes set (MDNS), the minimum input theorem is given below.

Theorem 2.3 (Minimum input theorem [18]). Given a digraph, the minimum number of external inputs (N_I) required to fully control the network equals to the number of unmatched nodes. There is an exception that $N_I = 1$ if a graph is a perfect matching, where any single node can be a driver node. Therefore,

$$N_I = N_D = \max\{1, N - |\mathcal{M}^*|\}$$

where $|\mathcal{M}^*|$ denotes the cardinality of the maximum matching of the graph.

FIGURE 2.3: **Networks with maximum matching.** Matched edges/edges are shown in Green.

2.2 Energy Cost in Controlling Complex Networks

In [37], the authors are one of the first researchers to consider the energy cost in controlling complex networks. Assuming the networked system is controllable with one single node, the energy cost required to control the network is

$$E = \int_0^{t_f} \|\mathbf{u}(t)\|^2 dt \quad (2.3)$$

Generally, there could be multiple ways to choose the input $\mathbf{u}(t)$ to drive the system from any initial state $\mathbf{x}_0$ to any desired final state $\mathbf{x}_f$ during the time interval $[0, t_f]$ [37, 50]. Among all these possibility, the optimal control input which minimizes the control energy cost is given by

$$\mathbf{u}(t) = -B^T e^{A^T(t_f-t)} W^{-1} (e^{At_f} \mathbf{x}_0 - \mathbf{x}_f) \quad (2.4)$$

where $W = \int_0^{t_f} e^{At} B B^T e^{A^T t} dt$ is the controllability Gramian matrix. When the system is controllable, W is positive definite and invertible.

By subsisting (2.4) to (2.3), we can obtain that

$$E(t_f) = (e^{At_f} \mathbf{x}_0 - \mathbf{x}_f)^T W^{-1} (e^{At_f} \mathbf{x}_0 - \mathbf{x}_f) \quad (2.5)$$

For convenience, the desired final state is set as $\mathbf{x}_f = 0$ and the control cost is

$$E(t_f) = \mathbf{x}_0 H^{-1} \mathbf{x}_0 \quad (2.6)$$

where $H = e^{-At_f} W e^{-A^T t_f}$.

The normalized energy cost is formulated as

$$E'(t_f) = E(t_f) / \|\mathbf{x}_0\|^2 = \frac{\mathbf{x}_0 H^{-1} \mathbf{x}_0}{\mathbf{x}_0^T \mathbf{x}_0} \quad (2.7)$$

According to Ray-leigh-Ritz theorem, we can bound the normalized energy cost by

$$E'_{\min} = \frac{1}{\eta_{\max}} \leq E'(t_f) \leq \frac{1}{\eta_{\min}} = E'_{\max} \quad (2.8)$$

where $\eta_{\min}$ and $\eta_{\max}$ are the smallest and largest eigenvalues of H , respectively.

For undirected networks, $A = [a_{ij}]_{N \times N}$ is symmetric weighted adjacency matrix with linear nodal dynamics, i.e. $a_{ii} = -|\sum_{j=1, j \neq i}^N a_{ij} + a|$ where a is a tuning parameter to control A positive definite (PD) or negative definite (ND). Assume that the undirected network can be controllable by a single node and the eigenvalues of A in the descending order is $\lambda_1 > \lambda_2 > \lambda_3 \cdots > \lambda_N$. In addition, the single node connected with external input is denoted as node c . Under these assumptions, it can be obtained that

$$E'_{\min} \begin{cases} \approx t_f^{-1} & \text{small } t_f \\ \approx \frac{1}{[(A+A^T)^{-1}]_{cc}} & \text{large } t_f, A \text{ is PD} \\ \sim t_f^{-1} \rightarrow 0 & \text{large } t_f, A \text{ is semi PD} \\ \sim e^{2\lambda_N t_f} \rightarrow 0 & \text{large } t_f, A \text{ is not PD} \end{cases} \quad (2.9)$$

and

$$E'_{\max} \begin{cases} \approx t_f^{-\vartheta} (\vartheta \gg 1) & \text{small } t_f \\ \approx \varepsilon(A, c) & \text{large } t_f, A \text{ is not ND} \\ \sim t_f^{-1} \rightarrow 0 & \text{large } t_f, A \text{ is semi ND} \\ \sim e^{2\lambda_1 t_f} \rightarrow 0 & \text{large } t_f, A \text{ is ND} \end{cases} \quad (2.10)$$

where $\varepsilon(A, c)$ is a positive value that is dependent on matrix A and node index c .

For a directed wighted network, suppose that it can be controllable for one driver node. Note that the eigenvalue of A is non-degenerate although it may have complex eigenvalues. Similar to undirected networks, let $\text{Re}(\lambda_1) > \text{Re}(\lambda_2) > \text{Re}(\lambda_3) \cdots > \text{Re}(\lambda_N)$. The corresponding upper bound and lower bound of energy keep unchanged as (2.9) and (2.10) except that λ_1, λ_N are replaced by $\text{Re}(\lambda_1), \text{Re}(\lambda_N)$ respectively.

In the case that the system can be controllable by one single node, it is revealed that the complex system consumes different energy in two time scale. In small time regime, the control cost can be reduced by setting the controlled time t_f slightly longer. In terms of large-time scale, there exists a turning point that the cost energy is no longer reduced although the final time t_f is increased. In addition, the authors point out that controlling the hub center will lead more energy cost.

2.2.1 Distribution of eigen-energies

In general, real networks function near a stable state. In the LTI model (2.1), A is the weighted matrix with nodal dynamics, i.e. $a_{ii} = -|\sum_{j=1, j \neq i}^N a_{ij} + a|$. Through tuning the parameter a , A can be negative definite such that the control energy decays quickly and converges to a stable value. By setting $\mathbf{x}_0 = 0$ and $\mathbf{x}_f = \mathbf{x}_d$ in (2.5), it is obtained that

$$E_d(t_f) = \mathbf{x}_d W^{-1} \mathbf{x}_d \quad (2.11)$$

This indicates that the energy needed to control a system depends on the final state $\mathbf{x}_d$. When $\mathbf{x}_d$ is a unit eigenvector of W associated with eigenvalue λ_i , the control cost consumed by driving the network from zero state to this eigen-direction is called eigen-energies, denoted as E_d^i and $E_d^i = 1/\lambda_i$. Geometrically, the control

energy surface is a super-ellipsoid spanned by N eigen-energies. In [38], the authors study the relationship between control energy cost and the eigen-directions of controllability Gramian matrix W . For the network with degree distribution $P_{\text{deg}}(k) \sim k^{-\gamma}$, Table 2.1 shows the distribution of eigen-energies and maximum control cost (corresponding to the direction that is most difficult to control) with different number of driver nodes. Actually, there are three main cases, controlling the network through 1) a single node, 2) a subset of its nodes, and 3) all nodes.

In Table 2.1, N_{peak} denotes the number of peaks for the distribution of $p[\log(E_{\text{eigen}}^i)]$. When γ is large, the network tends to be homogeneous, exhibiting similar behavior of random networks.

TABLE 2.1: Controlling complex networks with different number of driver nodes

Num. of driver nodes	Distribution of eigen-energies	Maximum control energy
$N_D = 1$	$p(E_d) = E_d^{-1}$	$E_d^{\max} \sim e^N$
$1 < N_D < N$	$N_{\text{peak}} = \text{int}[N/N_D]$ for $p[\log(E_d)]$	$E_d^{\max} \sim e^{N/N_D}$
$N_D = N$	$p(E_d) = E_d^{-\gamma}$	$E_d^{\max} \sim e^{1/(\gamma-1)}$

2.2.2 Energy-related metrics

From (2.11), it is obvious that controllability matrix W has close relation with control energy. In [52, 53], the authors specify several energy-related metrics to measure the consumed control energy.

1. $\text{tr}(W^{-1})$. In Section 2.2.1, it is stated that the inverse of eigenvalue of W represents the required energy to drive the networked system in one unit eigen-direction. As $\text{tr}(W^{-1}) = \sum_{i=1}^N \lambda_i$, it is easy to defer that $\text{tr}(W^{-1})$ measures the average control energy. Note that for a uncontrollable networked system, $\text{tr}(W^{-1})$ tends to infinity as there is at least one direction that is impossible to be arrived at. In this case, one can consider $\text{tr}(W^\dagger)$, indicating the average energy by moving the system around controlled state space. Here $\dagger$ denotes the pseudoinverse.
2. $\text{tr}(W)$. The trace of controllability Gramian matrix W is inversely related to the average of control energy cost given that

$$\frac{N}{\text{tr}(W)} \leq \frac{\text{tr}(W^{-1})}{N}$$

It also indicates the average controllability of all directions in the state space. In addition, $\text{tr}(W)$ is closely related to H_2 form due to

$$\begin{aligned}\|H_2\|^2 &= \text{tr} \left(C \int_0^\infty e^{At} B B^T e^{A^T t} dt C^T \right) \\ &= \text{tr} (C W C^T)\end{aligned}$$

3. $\log \det(W)$. The determinant of Gramian is a volumetric measure of consumed energy spanned by one unit or less reachable eigen-direction. This is because it is related to the volume of the enclosed super-ellipsoid given by

$$\frac{\pi^{N/2}}{\Gamma(N/2 + 1)} \sqrt[N]{\det(W)}$$

Note that when the system is uncontrollable, $\det(W) = 0$ and $\log \det(W) = -\infty$. In this case, only the volume of controllable state subspace can be considered.

4. $\lambda_{\min}(W)$. As $\frac{1}{\lambda_{\min}(W)}$ represents the maximum control energy, $\lambda_{\min}$ is inversely related to the control energy required to drive the system to the most difficult control direction.
5. $\text{rank}(W)$. $\text{rank}(W)$ measures the dimensionality of controllable state space for a system.

In [52], the authors demonstrate that $-\text{tr}(W^{-1})$, $\log \det(W)$ and $\text{rank}(W)$ are submodular [54]. Based on the submodularity of these metrics, the actuator placement problem is solved by an greedy algorithm with application in power grids.

2.3 Network Models

In this subsection, two typical network models, named Erdős-Rényi network and scale-free network are introduced.

FIGURE 2.4: (a) ER network. (b) Degree distribution of ER network

2.3.1 Erdős-Rényi networks

Erdős-Rényi (ER) model [55, 56] is to generate a random graph, which is first introduced by Paul Erdős and Alfrd Rényi in 1959. Generally, there are two random models. One is to randomly generate a graph with N vertices where two nodes are connected by an edge with probability p . This is $G(N, p)$ model. Another model is to generate a graph with N vertices and S edges, denoted as $G(N, S)$. Each edge is included in the graph with the probability p . Equivalently, a graph consisting of N nodes and S edges is generated with the probability

$$p^S (1 - p)^{[N(N-1)/2 - S]}$$

Fig. 2.4 shows an example of ER network where the node size is proportional to the node degree. From 2.4(b), it is clear that its degree obeys binomial distribution.

2.3.2 Scale-free networks

For a scale-free network [57, 58], the degree presents a power law distribution. For an undirected network, the degree distribution can be described as

$$P_{\text{deg}}(k) \sim k^{-\gamma}$$

FIGURE 2.5: (a) SF network. (b) Degree distribution of SF network

where γ is a parameter whose value is generally within the range $2 < \gamma < 3$.

Obviously, P_{deg} will decrease as k increases, resulting in some large hubs in the network, as illustrated in the Fig. 2.5.

One of the most important algorithms to generate scale-free networks is Barabási-Albert model by relying on the technique of preferential attachment. Specifically, given an initial network with m_0 nodes, one node is added to the network at one time and connected to the nodes with a probability that is proportional to the degree of the existing nodes. Specifically, the probability that the newly added node is connected with the existing node i is

$$p_i = \frac{k_i}{\sum_j k_j}$$

where k_i is the degree of the node i . With the growing of networks, hubs are formed gradually due to their accumulation of more and more links while the nodes with fewer links are less likely to be attached with newly arriving nodes.

derivatives are used in the theories of nonlinear elasticity and plasticity, particularly in the design of algorithms for numerical simulations. In the following chapters, we will apply the index notation arrangement to derive the gradient of the cost function. Here, for convenience, we present some lemmas and formulas which are essential to do first/second order derivatives of a function with respect to a matrix in this subsection.

Denote $[\cdot]_{\star\circ}$ as the $(\star, \circ)$ th element of the matrix $[\cdot]$ where $(\star, \circ)$ can have different notations. For compatible matrices A, B, X such that $X = AB$, then $X_{ij} = \sum_k A_{ik} B_{kj}$. It can be written as $X_{ij} = A_{ik} \cdot B_{kj}$ by introducing $[\cdot]$ operator. In what follows, the summation notation is omitted over repeated indices (Einstein notation) for the convenience of representation. Similarly, for compatible matrices A, B, C, X such that $X = ABC$, then $X_{ij} = (ABC)_{ij} = \sum_k \sum_p A_{ik} B_{kp} C_{pj}$ can be written as $X_{ij} = A_{ik} \cdot B_{kp} \cdot C_{pj}$.

Lemma 2.1. [59] *For a matrix X with X_{ij} as its ij -th element, denote that $\delta_{ij} = 1$ iff $i = j$ (otherwise, $\delta_{ij} = 0$). If the elements of X are independent, we have:*

- 1) $\frac{\partial X_{kl}}{\partial X_{ij}} = \delta_{ki} \cdot \delta_{lj}$
- 2) $\frac{\partial X_{kl}^T}{\partial X_{ij}} = \delta_{kj} \cdot \delta_{li}$
- 3) $X_{kl} = X_{lk}^T$
- 4) $\delta_{ki} \cdot \delta_{ij} = \delta_{kj}$

Lemma 2.2 (I-Chain rule [60]). *Suppose that matrix $U \in \mathbb{R}^{N \times M}$ is a function of matrix X , i.e., $U = g(X)$, E and g are differentiable. The derivative of the function $E(U) = E(g(X))$ with respect to X is given by the chain rule as follows:*

$$\begin{aligned} \frac{\partial E(g(X))}{\partial X_{ij}} &= \sum_{m,n} \frac{\partial E(U)}{\partial U_{mn}} \frac{\partial U_{mn}}{\partial X_{ij}} \\ &= \text{tr} \left(\left[\frac{\partial E(U)}{\partial U} \right]^T \left[\frac{\partial U}{\partial X_{ij}} \right] \right) = \frac{\partial E(U)}{\partial U_{mn}} \cdot \frac{\partial U_{mn}}{\partial X_{ij}} \end{aligned} \quad (2.12)$$

where X_{ij} is the ij -th element of matrix X , and the summation $\sum_{m,n}$ ($1 \leq m \leq M$, $1 \leq n \leq N$) over repeated indices is implied (Einstein notation) for the convenience of representation (Index-notation-arrangement). Thus, the ij -th element

of matrix $\frac{\partial E(g(X))}{\partial X}$ which is the derivative of $E(g(X))$ with respect to matrix X , denoted as $\left[\frac{\partial E(g(X))}{\partial X} \right]_{ij}$, is obtained by

$$\left[\frac{\partial E(g(X))}{\partial X} \right]_{ij} = \frac{\partial E(g(X))}{\partial X_{ij}} \quad (2.13)$$

Let $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, $f(\mathbf{x})$ is a function of vector $\mathbf{x}$ (first-order tensor). The gradient of $\mathbf{x}$ is $\frac{\partial f}{\partial \mathbf{x}} = [\frac{\partial f}{\partial x_1} \quad \dots \quad \frac{\partial f}{\partial x_n}]^T$. The Hessian matrix of $\mathbf{x}$ is a second order tensor, which is given by

$$\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (2.14)$$

Now consider an $n \times m$ matrix $X = [x_{ij}]$. Let $\{\cdot\}$ denote a vec operator applied on matrix X stacking its columns into a vector $\{X\}$, that is $\{X\} = [\mathbf{x}_1^T \quad \mathbf{x}_2^T \quad \dots \quad \mathbf{x}_m^T]$. Then, $\{X\} \in \mathbb{R}^{p \times 1}$ where $p = n \cdot m$. As $f(X)$ is a function of a matrix X , which is a second order tensor, then the gradient of $f(X)$ is also a second order tensor given by

$$\frac{\partial f(X)}{\partial X} = \left[\frac{\partial f}{\partial \mathbf{x}_1}, \dots, \frac{\partial f}{\partial \mathbf{x}_m} \right]$$

Note that f can be transformed into a function of $\{X\}$, denoted as $\tilde{f}(\{X\})$. Thus $\tilde{f}(\{X\}) = \{f(X)\}$. We know that the derivative of a function with respect to a first/second order tensor is still a first/second order tensor. As $\frac{\partial \tilde{f}(\{X\})}{\partial \{X\}} = \left\{ \frac{\partial f(X)}{\partial X} \right\}$, now we assume f and $\tilde{f}$ are twice continuously differentiable to derive the relationship between Hessian matrix and the fourth-order tensor.

The Hessian matrix of $f(X)$ is a fourth order tensor given by

$$\frac{\partial^2 f(X)}{\partial X \partial X^T} = \begin{bmatrix} \frac{\partial^2 f}{\partial \mathbf{x}_1 \partial \mathbf{x}_1^T} & \frac{\partial^2 f}{\partial \mathbf{x}_1 \partial \mathbf{x}_2^T} & \dots & \frac{\partial^2 f}{\partial \mathbf{x}_1 \partial \mathbf{x}_m^T} \\ \frac{\partial^2 f}{\partial \mathbf{x}_2 \partial \mathbf{x}_1^T} & \frac{\partial^2 f}{\partial \mathbf{x}_2 \partial \mathbf{x}_2^T} & \dots & \frac{\partial^2 f}{\partial \mathbf{x}_2 \partial \mathbf{x}_m^T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \mathbf{x}_m \partial \mathbf{x}_1^T} & \frac{\partial^2 f}{\partial \mathbf{x}_m \partial \mathbf{x}_2^T} & \dots & \frac{\partial^2 f}{\partial \mathbf{x}_m \partial \mathbf{x}_m^T} \end{bmatrix} \quad (2.15)$$

where every block like $\frac{\partial^2 f}{\partial \mathbf{x}_m \partial \mathbf{x}_m^T}$ is an $n \times n$ matrix, and there are totally $m \times m$ blocks.

Suppose a fourth order tensor $\mathcal{H}(i, l, k, j) = \frac{\partial(\frac{\partial f}{\partial X})_{ij}}{\partial (X^T)_{kl}} = \frac{\partial F_{ij}}{\partial X_{kl}^T} = \frac{\partial \mathcal{F}_{ij}}{\partial X_{lk}}$ where $\mathcal{F}_{ij} = (\frac{\partial f}{\partial X})_{ij}$, $1 \leq i \leq n, 1 \leq l \leq n, 1 \leq j \leq m, 1 \leq k \leq m$. Note that $\mathbf{x}_1$ corresponds to the first column of X , which is denoted as $\mathbf{x}_1 = X(:, m = 1)$. Similarly, $\frac{\partial f}{\partial \mathbf{x}_1}$, $\frac{\partial^2 f}{\partial \mathbf{x}_1 \partial \mathbf{x}_1^T}$ and $\frac{\partial^2 f}{\partial \mathbf{x}_1 \partial \mathbf{x}_2^T}$ correspond to $\mathcal{F}(:, j = 1)$, $\mathcal{H}(:, :, k = 1, j = 1)$ and $\mathcal{H}(:, :, k = 2, j = 1)$, respectively. We can obtain the Hessian matrix of $\tilde{f}(\{X\})$ as follows:

$$\frac{\partial^2 \tilde{f}(\{X\})}{\partial \{X\} \partial \{X\}^T} = \begin{bmatrix} \mathcal{H}(:, :, k = 1, j = 1) & \mathcal{H}(:, :, k = 2, j = 1) & \cdots & \mathcal{H}(:, :, k = m, j = 1) \\ \mathcal{H}(:, :, k = 1, j = 2) & \mathcal{H}(:, :, k = 2, j = 2) & \cdots & \mathcal{H}(:, :, k = m, j = 2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}(:, :, k = 1, j = m) & \mathcal{H}(:, :, k = 2, j = m) & \cdots & \mathcal{H}(:, :, k = m, j = m) \end{bmatrix} \quad (2.16)$$

Note that Eq. (2.16) represents the relationship between Hessian matrix and the fourth-order tensor, which will be used in Chapter 3.

Chapter 3

Minimum-cost Control of Complex Networks Based on Matrix Differentiation

In this chapter, we start to present the main results of this thesis by designing the input matrix for controlling the complex networks at minimum control cost.

Finding the key node set to be connected to external control sources so as to minimize the energy for controlling a complex network, known as the minimum energy control problem, is of critical importance but remains open. We address this critical problem where matrix differentiation is involved. To this end, the differentiation of energy/cost function with respect to the input matrix is obtained based on tensor analysis, and the Hessian matrix is compressed from a fourth order tensor. Projected gradient method (PGM) and projected trust-region method (PTM) are proposed with established convergence property. We show that PGM is more computationally efficient than PTM. With the definition of important index, PGM extension (PGME) is proposed in Monte-Carlo realizations and key node set is identified. Simulation results demonstrate satisfactory performance of the algorithms, and reveal important insights as well. Several interesting phenomena are observed. We first point out that key node set tends to divide elementary paths equally. Then it is found that the importance index of a node is strongly correlated to occurrence rate of that node to be selected as a key node in Monte-Carlo realizations. Finally, the low degree nodes may be more important than hubs

from control point of view, indicating that controlling hub nodes does not help to lower the control energy. These results suggest a way of achieving optimal control of complex networks, and provide meaningful insights for future researches.

3.1 Introduction

In controlling complex networks, one of the most important and challenging problems is to locate a subset of nodes connected with a given number of external control sources such that the control cost is minimum. Generally speaking, given a number of external control sources to drive the network, there could be multiple possible ways to connect them with nodes in the network. Among these sets, the one which can control the network with minimum control cost is **key node set**. Due to limited and valuable resources, researchers start to investigate how to locate the “key node set” connected with external controllers with minimum energy cost [39, 52, 61, 62]. In this chapter, we try to solve this problem based on matrix differentiation. Specifically, to formulate an optimization problem, we need to determine how the cost function changes with respect to the matrix B . That is to say, we want to know the first order (gradient information) and second order (Hessian matrix [63]) derivatives of the cost function. However, this is generally quite challenging as it involves matrices-by-matrices derivatives. Although vector derivative has been well established, matrix derivative is difficult. Currently there is no unified framework that can completely solve this problem [64]. For example, when finding the derivative of a matrix function $F(X)$ that maps matrix $X \in \mathbb{R}^{n \times m}$ to matrix $F(\cdot) \in \mathbb{R}^{p \times q}$, denoted as $\frac{\partial F_{ij}}{\partial X_{kl}}$ for all i, j and k, l , which is a fourth order tensor [65]. Existing schemes mainly apply two basic ways to deal with this issue, one is the vec operator and Kronecker products arrangement [66], and the other is the index notation arrangement [67]. However, in implementations, there are a lot of intricacy and tedious calculations. This situation becomes worse when $F(\cdot)$ has a more complicated form. We find that index notation arrangement relatively simplifies the presentation and manipulation of differential geometry in the minimum-cost control problem. Thus, index notation arrangement for matrix differentiation is applied in this work.

Our approach to addressing the above long-standing issue is summarized as follows. Firstly, we formulate an optimization model for the minimum-cost control

problem where input matrix B is a variable. Secondly, we obtain the gradient and the Hessian matrices of the cost function based on matrix differentiation. By using projected gradient method (PGM) or projected trust-region method (PTM), the optimization problem can be solved. We point out that, in applications, PGM and PTM lead to similar solutions, while PTM requires computing a double integral to obtain the gradient and compressing a 4th order tensor to a Hessian matrix, which implies that the computational time is much higher than that of PGM. Thus, we usually employ the PGM method in applications. With the definition of importance index, PGM extension (PGME) is proposed to pinpoint the key nodes. By implementing PGME, we observe that key nodes tend to divide elementary topologies equally. Another interesting observation is that distribution of importance index is strongly correlated with the occurrence rate in Monte-Carlo realizations. It is also found that the mean degree of the selected key nodes is significantly below the mean degree. Extensive simulations have been finally carried out to show the performances of the proposed method with various synthetic and real-life networks. It is illustrated that for various cases, the proposed method significantly and consistently reduces the control energy compared with a randomly allocation method (RAM).

There are also other related works in existing works such as “pinning controllability” [68] or “network synchronizability” problems [69], and consensus or agreement problems in multi-agent systems. These works mainly addresses whether the system can exhibit specific spatiotemporal symmetries and the robustness of such symmetries. For example, there are some algorithms to select “leader agent” in stochastically forced consensus networks so as to minimize the mean-square deviation from consensus [70, 71]. The problem we addressed in this chapter is fundamentally different from existing works. We wonder whether the system can fully explore its state space and how much energy is necessary if possible.

The rest of this chapter is organized as follows. In Section 3.2, the optimization model for the minimum-cost control problem is formulated. The gradient and Hessian matrices are calculated in Section 3.3. In Section 3.4, two algorithms named PGM and PTM are proposed. Section 3.5, the convergence properties of PGM and PTM are analyzed. Simulations about synthetic and real-life networks are given in Section 3.6. Section 3.7 concludes this chapter.

3.2 Problem Formulation

Consider a directed network system consisting of N nodes and M external control sources. Its dynamics is described (2.1) where matrix $A = [a_{ij}]_{N \times N}$ is network adjacency matrix which describes the interconnections between nodes, i.e. $A_{ij} = 1$ if there is an edge from node j to node i ; otherwise $A_{ij} = 0$. $B = [b_{im}]_{N \times M}$ is the input matrix where $b_{im} \neq 0$ if external control source m is connected to node i ; and $b_{im} = 0$ otherwise; $\mathbf{u}(t) = [u_i(t)]_{M \times 1}$ is the vector of the input signals, and $\mathbf{x}_0$ is a random initial state. The control objective in this chapter is to design B and $\mathbf{u}(t)$ such that the system state can be driven from an randomly given $\mathbf{x}_0 = [x_1(0), \dots, x_N(0)]^T$ to the origin at $t = t_f$, i.e., $\mathbf{x}(t_f) = [0, \dots, 0]^T$, subject to the condition that the control cost is minimized. As there are multiple possible initial states, here we consider “average control energy” given by $\mathbb{E}[\int_0^{t_f} \|\mathbf{u}(t)\|_2^2 dt]$, where an expectation function $\mathbb{E}[\cdot]$ arguments over all realizations of the random initial state $\mathbf{x}_0$. Without loss of generality, it is assumed that each element of $\mathbf{x}_0$ is an identical independent distributed (i.i.d) variable with zero mean and variance 1. This assumption holds in various applications of science and engineering since the elements of $\mathbf{x}_0$ are the initial states of different nodes. When we sum all realizations of $x_i(0)$ and $x_j(0)$ together, we obtain that $\mathbb{E}[x_i(0)x_j(0)] = \delta_{ij}$ where $\delta_{ij} = 1$ if and only if $i = j$, otherwise, $\delta_{ij} = 0$. Thus, $\mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T]$ can be regarded as an identity matrix since

$$\begin{aligned} & \mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] \\ &= \begin{pmatrix} \mathbb{E}(x_1(0)x_1(0)) & \dots & \mathbb{E}(x_1(0)x_N(0)) \\ \mathbb{E}(x_2(0)x_1(0)) & \dots & \mathbb{E}(x_2(0)x_N(0)) \\ & \ddots & \vdots \\ \mathbb{E}(x_N(0)x_1(0)) & \dots & \mathbb{E}(x_N(0)x_N(0)) \end{pmatrix} \\ &= I_N \end{aligned} \tag{3.1}$$

where I_N is an identity matrix with dimension N . As mentioned before, the problem we shall tackle in this chapter is to locate M key nodes for a directed network that leads to the minimum control energy.

Assumption 3.1. (A, B) is controllable.

Assumption 3.2. The Frobenius norm of the input matrix $\|B\|_F^2 = \omega$ for some arbitrary parameter ω .

Remark 3.1. Assumption 3.1 implies that M should be greater than a lower bound value denoted as N_D , which can be determined by the maximum matching algorithm in [18]. For Assumption 3.2, it is easy to see that for two pairs $(\tilde{B}, \tilde{\mathbf{u}}(t))$ and $(B, \mathbf{u}(t))$ such that $\tilde{B} = 2B$ and $\tilde{\mathbf{u}}(t) = \frac{1}{2}\mathbf{u}(t)$, the system just has exactly the same behaviour. Thus, to formulate the minimum-cost control problem, the norm of B should be fixed. Owing to equivalence of all matrix norms, we choose the Frobenius norm $\|\cdot\|_F^2$ and set $\omega = M$ in presenting the optimization model later without losing generality.

Remark 3.2. In this work, we consider the networks with dynamics $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$ where topology matrix A is an adjacency matrix that is generally asymmetrical for directed networks. The matrix A could be also a Laplacian matrix when undirected networks is considered. Such a dynamic captures the fundamental/essential characteristics of various networks considered in existing literatures [1, 18, 37, 58]. We also agree that modeling the dynamics of more complicated real world networks is a promising yet changeling research topic.

We consider the problem of driving the system states to the origin in the time interval $[0, t_f]$ while minimizing the energy [37]

$$\min_{\mathbf{u}(t), B} \mathbb{E} \left[\int_0^{t_f} \|\mathbf{u}(t)\|^2 dt \right] \quad (3.2)$$

where t_f is a given time constant, and $\mathbf{u}(t)$ and B are the decision variables to be determined by the optimization process. Here $\mathbf{u}(t)$ specifies the values of external control source inputs at each operating time and B determines the nodes to be connected to the external control sources as well as the weights of these connections.

If B satisfies that (A, B) is controllable, $\mathbf{u}(t)$ can be designed as (2.4) for minimum-energy control, where $\mathbf{u}(t)$ is the function of B . To emphasize that the controllability Gramian matrix also depends on B , we denote controllability Gramian matrix as W_B here.

Substituting the optimal $\mathbf{u}(t)$ in (2.4) into (3.2), it yields that

$$\begin{aligned}
& \min_B \mathbb{E}[\mathbf{x}_0^T e^{A^T t_f} W_B^{-1} e^{A t_f} \mathbf{x}_0] \\
&= \min_B \text{tr}(W_B^{-1} e^{A t_f} \mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] e^{A^T t_f}) \\
&= \min_B \text{tr}(W_B^{-1} e^{A t_f} e^{A^T t_f}) \\
&\triangleq \min_B \mathcal{E}(B)
\end{aligned} \tag{3.3}$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, B is a matrix variable to be determined and $\mathcal{E}(B) = \text{tr}(W_B^{-1} e^{A t_f} e^{A^T t_f})$ is the defined cost function to be minimized.

Assumption 3.2 implies that the trace $\text{tr}(B^T B)$ is fixed as M . Thus, the optimization problem is formulated as

$$\begin{aligned}
& \min_B \quad \mathcal{E}(B) = \text{tr} \left(W_B^{-1} e^{A t_f} e^{A^T t_f} \right) \\
& \text{s.t.} \quad \text{tr}(B^T B) = M
\end{aligned} \tag{3.4}$$

where B becomes a matrix variable. The objective is to determine B such that $\mathcal{E}(B)$ could be minimized on the sphere surface $\text{tr}(B^T B) = M$. It is worth to mention that the proposed algorithms are still suitable to minimize the control cost with a specific initial state, i.e. minimize $E(B) = \text{tr} \left(W_B^{-1} e^{A t_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} \right)$ under the constraint $\text{tr}(B^T B) = M$. For the convenience of representation, we define a norm function $\mathcal{N}(B) = \text{tr}(B^T B)$.

3.3 First and Second order derivatives of $\mathcal{E}(B)$

To obtain the gradient and Hessian matrix of $\mathcal{E}(B)$, we present the following lemmas regarding the first and second derivatives of functions where matrix variable B is involved.

Lemma 3.1. *The gradient of norm function $\mathcal{N}(B)$ with respect to B is parallel to B .*

Lemma 3.2. *The gradient of the function $\mathcal{E}(B)$ in (3.4) is*

$$\frac{\partial \mathcal{E}(B)}{\partial B} = -2 \int_0^{t_f} e^{A^T t} W_B^{-1} e^{A t_f} e^{A^T t_f} W_B^{-1} e^{A t} dt B.$$

Lemma 3.3. *The second order derivative of $\mathcal{E}(B)$ with respect to matrix B is a fourth order tensor, which is given by*

$$\begin{aligned}
 \mathcal{H}(i, j, k, l) &= \frac{\partial \left[\frac{\partial \mathcal{E}(B)}{\partial B} \right]_{ij}}{\partial B_{kl}^T} \\
 &= 2 \int_0^{t_f} [e^{A^T t}]_{im} \left[\int_0^{t_f} [[W_B^{-1} e^{At}]_{ml} [B^T e^{A^T t} W_B^{-1}]_{kn} \right. \\
 &\quad \left. + [W_B^{-1} e^{At} B]_{mk} [e^{A^T t} W_B^{-1}]_{ln}] dt \right] [e^{At_f} e^{A^T t_f} W_B^{-1} e^{At} B]_{nj} dt \\
 &\quad + 2 \int_0^{t_f} [e^{A^T t} W_B^{-1} e^{At_f} e^{A^T t_f}]_{iz} \left[\int_0^{t_f} [W_B^{-1} e^{At}]_{zl} [B^T e^{A^T t} W_B^{-1}]_{kr} \right. \\
 &\quad \left. + [W_B^{-1} e^{At} B]_{zk} [e^{A^T t} W_B^{-1}]_{lr}] dt \right] [e^{At} B]_{rj} dt \\
 &\quad - 2 \int_0^{t_f} [e^{A^T t} W_B^{-1} e^{At_f} e^{A^T t_f} W_B^{-1} e^{At}]_{il} \delta_{jk} dt
 \end{aligned} \tag{3.5}$$

The proofs of Lemmas 3.1-3.3 can be seen in the Appendix A.

3.4 Algorithm Design

In this section, we propose PGM and PTM to solve the optimization model (3.4). We first define a projection operator in Lemma 3.4 which will be applied in the algorithm design.

Lemma 3.4. *For a matrix $B \in \mathbb{R}^{N \times M}$, define a projection operator $\mathcal{P}_{\frac{\partial \mathcal{N}(B)}{\partial B}}(v) = (I - \{B\}\{B\}^\dagger)(v)$ where † denotes the Moore-Penrose pseudoinverse and $\{\cdot\}$ denotes a vector form of an argument matrix. Then, $\mathcal{P}_{\frac{\partial \mathcal{N}(B)}{\partial B}}$ is a projection operator which projects an arbitrary vector $v \in \mathbb{R}^{N \cdot M \times 1}$ onto a space orthogonal to the gradient of the norm function $\mathcal{N}(B)$.*

3.4.1 Projected gradient method (PGM)

Now we present an algorithm solving the optimization problem (3.4). Define $\Gamma(B)$ as a normalization operator which normalizes B onto the surface $\mathcal{N}(B) = M$, given by $\Gamma(B) = \sqrt{\frac{M}{\text{tr}(B^T B)}} B$. The main idea of PGM is as follows. Firstly, do

the derivatives of functions $\mathcal{N}(B)$ and $\mathcal{E}(B)$ with respect to B , respectively. Secondly, project the gradient of $\mathcal{E}(B)$ onto the space perpendicular to the gradient of $\mathcal{N}(B)$ (projection). Lastly, drag the projected vector onto the surface $\mathcal{N}(B) = M$ (normalization). We repeat this process until the sequence converges.

- 1) Step 1. Initiate B randomly as matrix B_0 with a dimension $N \times M$ where $M \geq N_D$.
- 2) Step 2 (projection). At iteration k , calculate the gradients $\frac{\partial \mathcal{E}(B_k)}{\partial B_k}$ and its vector version $\left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}$ at $B = B_k$. Applying the projection operator defined in Lemma 3.4 on $\left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}$, obtain

$$\left\{ \widetilde{\frac{\partial \mathcal{E}(B_k)}{\partial B_k}} \right\} = \mathcal{P}_{\frac{\partial \mathcal{N}(B_k)}{\partial B_k}} \left(\left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\} \right) \text{ where } \left\{ \widetilde{\frac{\partial \mathcal{E}(B_k)}{\partial B_k}} \right\} \text{ is a } N \cdot M \text{ dimensional vector.}$$
- 3) Step 3 (normalization). Denote $\mathcal{U}(\cdot)$ as a column-wise operator which reshapes an $N \cdot M$ dimensional vector back to a $N \times M$ matrix, and do the gradient decent iteration:

$$B_{k+1} = \Gamma \left[B_k - \eta \cdot \mathcal{U} \left(\left\{ \widetilde{\frac{\partial \mathcal{E}(B_k)}{\partial B_k}} \right\} \right) \right]$$

where η is a chosen step length.

- 4) Step 4. Calculate the angle θ_{k+1} between $\left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}$ and $\left\{ \frac{\partial \mathcal{N}(B_k)}{\partial B_k} \right\}$ at $B = B_{k+1}$.
- 5) Step 5. Update $k = k + 1$. If $\cos(\theta_k) + 1 > \xi$ for a sufficiently small positive ξ , go to Step 2; otherwise, stop.

Denote the solution obtained by PGM as B^* . In the next section, we will prove that $\mathcal{E}(B_k)$ (for $k = 1, 2, \dots$) is a non-increasing sequence which converges to a nonzero local minimum point $\mathcal{E}(B^*)$ as $\cos(\theta_k)$ converges to -1 , or equivalently, as θ_k converges to $-\pi$.

3.4.2 Projected trust-region method (PTM)

Trust region is a term used in mathematical optimization to denote the subset of the region of the objective function that is approximated using a quadratic function

[72]. The only difference between PGM and PTM is in Step 3, the gradient decent iteration is given by

$$B_{k+1} = \Gamma \left[B_k - \eta \cdot \mathcal{U} \left((H_k + \zeta_k I)^{-1} \left\{ \frac{\partial \widetilde{\mathcal{E}(B_k)}}{\partial B_k} \right\} \right) \right]$$

where H_k is the Hessian matrix of $\mathcal{E}(\{B\})$ which can be obtained based on Lemma 3.3 and the formula (2.16) in Chapter 2, ζ_k is a positive parameter such that $H_k + \zeta_k I$ is positive definite, and η is a chosen step length.

3.5 Convergence Analysis

Theorem 3.1. For a controllable (A, B) , the gradient $\frac{\partial \mathcal{E}(B)}{\partial B}$ can never be a zero matrix as long as $B \neq 0$.

Remark 3.3. Theorem 3.1 implies that we cannot search for the minimum point of $\mathcal{E}(B)$ by only applying the classic gradient information, since $B = 0$ will be a trivial solution. This is also the reason why problem (3.4) has to be optimized by fixing the norm of B .

Theorem 3.2. For PGM, $\mathcal{E}(B_k)$ is non-increasing as k increases and it converges to a nonzero local minimum point B^* such that $\mathcal{E}(B^*) > 0$.

Theorem 3.3. The proposed algorithm PGM converges linearly.

The proofs of Theorem 3.1-3.3 are shown in Appendix A.

Remark 3.4. The convergence of PTM can be achieved similarly. We also would like to point out that for proposed PGM and PTM methods, like any non-convex optimization method with a reasonably low complexity, can only guarantee converging to a local minimum point. Interestingly, in our extensive implementations of PGM/PTM starting with different initial B_0 matrices shown in the next section, we observed that the solutions of different rounds of implementations typically lead to nearly the same control energy. Whether that means that the solution found by a single round implementation of PGM/PTM is steadily close to the global optimum requires further careful studies.

Remark 3.5. In implementations, the computation time of PGM/PTM depends on the convergence rate, computational and memory complexity of each iteration. From Theorem 3.3 and Remark 3.3, PGM/PTM converges linearly, and normally PTM requires fewer iterations by exploiting the second order derivative information of $\mathcal{E}(B)$ [72]. We know that the computational complexity of each iteration in PGM mainly depends on the calculation of $\frac{\partial \mathcal{E}(B)}{\partial B}$ involving the matrix multiplication and matrix inverse. Then, the computational complexity of PGM is $O(N^3)$ with memory complexity $O(N^2)$. For PTM, as it requires computing a double integral in Lemma 3.3 to obtain the gradient and compressing a 4th order tensor to a Hessian matrix with dimension $NM \times NM$, the computational complexity and memory complexity of each iteration become $O(N^3M^3)$ and $O(N^2M^2)$, respectively. This implies that the computational time of PTM is much higher than that of PGM. This is why we suggest to employ PGM rather than PTM in real applications.

3.6 Simulation Results

For the solution obtained by PGM, it may require each control source connect with many nodes. This is not practical for large-size networks. Therefore, based on the definition of importance index, PGME Extension (PGME) is now proposed to obtain a sparse input matrix B^s with one-to-one connections between key nodes and external control sources. As PGM converges to a local optimum, we apply PGM in Monte-Carlo realizations to further approximate the global solution. Detailed definition of importance index of a node is given below.

Definition 3.1 (Importance index). Note that the absolute value of link weight $|B_{ij}^*|$ reflects on the importance of the connection from the j -th control source to node i . Let r_i^t denote the importance index of node i in experiment t , which is given by

$$r_i^t = \frac{q_i^t}{\max(q_1^t, \dots, q_i^t, \dots, q_N^t)}$$

where $q_i^t = \sum_j |B_{ij}^*|$ for $i = 1, \dots, N$. When experiments are run n times independently in Monte-Carlo realizations, importance index of each node is defined

as

$$r_i = \frac{\sum_{t=1}^{t=n} r_i^t}{\max\{\sum_{t=1}^{t=n} r_i^t\}} \quad (3.6)$$

where $\max\{\sum_{t=1}^{t=n} r_i^t\} = \max\{\sum_{t=1}^{t=n} r_1^t, \dots, \sum_{t=1}^{t=n} r_i^t, \dots, \sum_{t=1}^{t=n} r_N^t\}$.

Clearly, we have $\max\{r_i\} = 1$. The importance index of node i evaluates the relative importance of this node in achieving the minimum-cost control objective in Monte-Carlo realizations. Let $\mathbf{r} = [r_1, r_2, \dots, r_N]^T$ be a vector including importance indices of all nodes in the network. Then nodes with the M largest importance indices are selected as the **key nodes** of the network. The corresponding connection matrix, denoted as B^s , can then be easily constructed:

$$B_{i,j}^s = \begin{cases} 1 & \text{if node } i \text{ is connected with control source } j \\ 0 & \text{otherwise} \end{cases}$$

Note that only one element in the column vector $B^s(:, j)$ is 1 while all the rest are 0. For the row vector $B^s(i, :)$, at most one element is 1 while the rest are 0. This means that, if $B_{ij}^s = 1$, node i can be connected and can only be connected to one control source j while control source j can directly control one node i . Combined with PGM, we term the whole approach for determining key nodes as PGM Extension (PGME).

Additionally, another concern in simulations is how to choose the number of external control sources, which is detailed illustrated in Remark 3.6.

Remark 3.6. Numerical controllability transition problem [73] means that when a small number of key nodes is barely enough to ensure numerical controllability, failure of finding a numerical solution may not be overcome by merely increasing numerical precision. A successful control requires instead increasing the number of control inputs beyond the numerical controllability transition [73]. In the following simulations, there is a trade-off between the number of control sources and control energy. Being able to identify the minimum control energy when having different number of external control sources respectively helps to make most cost-effective decisions in applications.

TABLE 3.1: Control energy in elementary topologies.

Network	M	$\mathcal{E}(B^s) (t_f = 1)$	$(t_f = 2)$	$(t_f = 3)$
a1	1	6.48E09	3.98E06	7.44E04
a2	2	1.10E03	50.21	11.85
a3	3	32.67	6.75	3.77
b1	1	1.17E09	7.86E05	1.49E04
b2	2	1.10E03	50.28	11.98
b3	3	32.62	6.78	3.81
c1	2	1.04E03	49.21	11.27
c2	3	31.90	6.72	3.72

In the following, we will examine the effectiveness of proposed algorithms through a great number of experimental tests on three elementary topologies, Erdős-Rényi, scale-free networks and real-life networks. Some interesting results are found.

3.6.1 PGME in three elementary topologies

Synthetic and real-life networks widely contain three elementary topologies: a stem, a circle and a dilation [18]. In this subsection, we take experiments on these elementary topologies.

Results of three elementary topologies with different number of external inputs are illustrated in Fig. 3.1. Each of which contains 6 nodes. Note that the first two elementary topologies can be controlled by having as few as a single driver node, while the third one needs at least 2 driver nodes. To explore the relationship of control energy versus the number of key nodes, we carry out simulations for these three topologies with M key nodes, where $M = 1, 2, 3$ for the first two cases and $M = 2, 3$ for the third case. The control energies for different cases are listed in Table 3.1. By employing PGME, the solution B^s can be obtained for each elementary topology. In Fig. 3.1, it is observed that i) the control energy decreases significantly as t_f increases; ii) key nodes tend to divide the elementary path equally. This observation is very important and may lead to a heuristic algorithm regarding how to choose the key node set in large complex networks; iii) lastly but arguably most importantly, having more external control sources (key nodes) helps drastically to lower the control energy. Yet it is well known that installing more external control sources may induce other costs.

FIGURE 3.1: Key nodes in three elementary topologies.

In addition, to further explore the distribution of importance indices, we have the following definitions.

Definition 3.2 (Occurrence rate). For each network, n times of independent experiments are carried out. In each experiment, nodes with the M largest importance index are considered as key nodes. Suppose there are n_i times that node i is selected as a key node in the n experiments. Then the occurrence rate p_i of node i as a key node is calculated as

$$p_i = \frac{n_i}{n}$$

Denote $\mathbf{p} = [p_1, p_2, \dots, p_N]^T$ as a vector including occurrence rates of all nodes in the network. To measure the similarity of importance indices and occurrence rates, now we define correlation coefficient ρ_{rp} as

$$\rho_{rp} = \frac{\mathbf{r}^T \mathbf{p}}{\|\mathbf{r}\| \|\mathbf{p}\|}$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector. Since $\mathbf{r}^T \mathbf{p} > 0$, we have $0 < \rho_{rp} \leq 1$. When ρ_{rp} approaches to 1, it means that $\mathbf{r}$ and $\mathbf{p}$ are highly related.

Four elementary topologies, listed as Network I, II, III and IV in Table 3.2, are respectively considered. We now make discussions on the results obtained for the

three topologies respectively. For an elementary stem, i.e. Network I, we take $N = 50, M = 10$ as an example. 10, 100 and 1000 independent experiments have been done respectively with $t_f = 2$. Fig. 3.3 illustrates the detailed distribution of r_i and p_i with 1000 independent experiments. As for r_i , it is interesting that there is a composite wave consisting of two damped oscillatory waves traveling through the medium of control path, with the two sides being the respective source of each wave. It can be observed that the importance indices present a **quasi-periodic distribution** with 5 nodes being a quasi-period. In detail, nodes $\{1, 6, \dots, 46\}$ correspond to 10 largest importance indices and constitute key node set. Obviously, these key nodes divide the stem equally. On one hand, within each quasi-period, the importance index of the first node is highest and is decreasing as the node index increases. On the other hand, the importance indices of key nodes at both ends, i.e. nodes 1 and 46, are higher than those in the middle, as the envelopes shown. This is reasonable as we can pinpoint key nodes in the starting and ending part easily with certainty for controlling an elementary stem with minimum control cost. However, for nodes in the middle, uncertainty on which node can be a key node is higher. Such certainty and uncertainty are clearly illustrated by the occurrence rate of each node being a key node. Actually, such phenomena have also been explored in psychological and social science. As for memory, psychologists agree that chunking, the process by which individual units of information are grouped into clusters, increases the capacity limit of working memory [74–76]. It becomes easier to retain and recall information by chunking when people try to memorize some things. For example, a phone number 93208215 may be chunked into two clusters 9320 – 8215. From the point of view of energy cost in complex networks, to remember some information, each chunk will be regarded as one quasi-period. Driving each chunk with a key node consumes minimum energy. On the other hand, people tend to recall primary information presented better than information presented later on. This is known as “**primary effect**” [77, 78]. Another interesting and important phenomenon in human memory is “**recency effect**” [79, 80], i.e. the most recently presented items or experiences will most likely be remembered best. Viewing the importance index vector of an elementary stem as the ability of memory with time sequence, it is obvious that key nodes locating in the first and last chunks are of greater importance than the middle chunks, which helps explain primary effect and recency effect.

For an elementary cycle, i.e. Network II, we set $N = 100, M = 25$ which implies

FIGURE 3.2: **Elementary dilations.** (a) An elementary dilation with $N = 8, M = 4$. (b) An elementary dilation with $N = 100, M = 25$. Red points are key nodes with minimum control cost.

$\{(v_1 \rightarrow v_2), (v_2 \rightarrow v_3), \dots (v_{99} \rightarrow v_{100}), (v_{100} \rightarrow v_1)\}$. Fig. 3.4 shows the distribution of importance index and occurrence rate when 1000 rounds of simulations are run independently. Table 3.3 presents the variance of r_i and p_i with different n , which is near zero. It is obvious that the importance of every node is almost the same and each node has the same chance to act as a key node. When $n = 1000$, the correlation coefficient is 0.9985 in Fig. 3.6, which is almost 1.

Two elementary dilations networks are constructed as in Figure 3.2. For elementary dilation (a), i.e. Network III, it can be considered as two elementary stems: one is $\{(v_1 \rightarrow v_2), (v_2 \rightarrow v_3), (v_3 \rightarrow v_4)\}$, and the other is $\{(v_1 \rightarrow v_5), (v_5 \rightarrow v_6), (v_6 \rightarrow v_7), (v_7 \rightarrow v_8)\}$, with the fact that node 1 should be one of key nodes to guarantee controllability of the network system but its control path direction is uncertain. Actually, for $N = 8, M = 4$, there will be 25 possible driver node sets directly connected with external control sources to guarantee the controllability of the network. Table 3.4 summarizes control costs consumed by all these possible sets. Clearly, nodes $\{1, 3, 5, 7\}$ are selected as key nodes because the cost is lowest. This is consistent with the simulation result that the importance indices and occurrence rates of nodes $\{1, 3, 5, 7\}$ are both the highest. Fig. 3.5 shows importance index and occurrence rate distribution of another elementary dilation, i.e. Network IV. From Fig. 3.5, we also conclude that key nodes distribute as equally as possible in an elementary dilation. In addition, an elementary dilation can be seen as two stems and importance indices of nodes in each stem also show similar pattern as Fig. 3.3.

TABLE 3.2: Four elementary networks

#	Network	Topologies	N	M
I		elementary stem	50	10
II		elementary cycle	100	25
III		elementary dilation	8	4
IV		elementary dilation	100	25

TABLE 3.3: Variance of r_i and p_i with different n in Network II

n	$\text{Var}(r_i)$	$\text{Var}(p_i)$
10	9.9683×10^{-4}	0.0132
100	1.2078×10^{-4}	0.0019
1000	1.5656×10^{-5}	2.1226×10^{-4}

FIGURE 3.3: Importance and occurrence rate distribution of nodes in an elementary stem with $N = 50$, $M = 10$ based on 1000 independent simulations. The dotted lines present envelopes.

3.6.2 PGME in Erdős-Rényi and scale-free networks

To further validate the effectiveness of proposed algorithm, we employ PGME on SF ER [81, 82] and SF networks [83], which are known as the most two pervasive complex networks [18]. SF networks are generated by Barabási-Albert (BA) model [58]. To start the experiments, we set $N = 500$ nodes and mean degree $\langle k \rangle = 4$, and give a random initial B_0 for a fixed M . The same as [73], to ensure that the $\mathbf{x}(t_f)$ approaches the origin point within the error tolerance 10^{-6} , we set the percentages of key nodes to be at least higher than 20% in all simulations to avoid the numerical controllability transition mentioned in Remark 3.6. Specifically, we set $M = 145, 175$ and 205 , respectively, as the percentages of driver nodes for ER and

FIGURE 3.4: Importance and occurrence rate distribution of nodes in an elementary cycle with $N = 100$, $M = 25$ based on 1000 independent simulations.

TABLE 3.4: Control cost with different controlled nodes in Network III

Controlled nodes	$\mathcal{E}(B^s)$	Controlled nodes	$\mathcal{E}(B^s)$
$\{1, 2, 3, 4\}$	1.1295×10^5	$\{1, 3, 4, 5\}$	1.7244×10^3
$\{1, 2, 3, 6\}$	60.5961	$\{1, 3, 5, 6\}$	57.4903
$\{1, 2, 3, 7\}$	61.8234	$\{1, 3, 5, 7\}$	16.2479
$\{1, 2, 3, 8\}$	1.7982×10^3	$\{1, 3, 5, 8\}$	60.1645
$\{1, 2, 4, 6\}$	60.8472	$\{1, 4, 5, 6\}$	103.7561
$\{1, 2, 4, 7\}$	64.4.93	$\{1, 4, 5, 7\}$	64.3614
$\{1, 2, 4, 8\}$	1.7989×10^3	$\{1, 4, 5, 8\}$	108.2778
$\{1, 2, 6, 7\}$	59.6116	$\{1, 5, 6, 7\}$	1.7957×10^3
$\{1, 2, 6, 8\}$	60.1139	$\{1, 5, 6, 8\}$	1.7982×10^3
$\{1, 2, 7, 8\}$	108.0136	$\{1, 5, 7, 8\}$	1.7986×10^3
$\{1, 2, 3, 5\}$	1.7238×10^3	$\{1, 2, 4, 5\}$	1.7238×10^3
$\{1, 2, 5, 6\}$	98.9739	$\{1, 2, 5, 7\}$	57.7246
$\{1, 2, 5, 8\}$	100.2378		

SF networks are around 20%. The comparisons between the control energy for the random initial B_0 and the B^s obtained by PGME are listed Table 3.5. It can be seen that the control energy decreases as M increases and the PGME algorithm lowers the control energy by about 3 orders of magnitude compared with a randomly given B_0 .

Note that once M is given, we can identify M most important nodes based on the definition of importance index. These M nodes can be marked as the key nodes. One important observation is that the mean degree of the key nodes is significantly below the mean degree of the network as shown in Table 3.6. Here the

FIGURE 3.5: Importance and occurrence rate distribution of key nodes in an elementary dilation with $N = 100$, $M = 25$ based on 1000 independent simulations. The dotted lines present envelopes.

FIGURE 3.6: The correlation between importance index r and occurrence rate p with different n

TABLE 3.5: Energy consumed in ER and SF networks.

$N = 500, \langle k \rangle = 4$	$M = 145$	$M = 175$	$M = 205$
ER (Initial)	$2.65E08$	$3.61E07$	$4.51E06$
ER (PGME)	$1.65E05$	$1.89E04$	$6.13E03$
SF (Initial)	$4.04E08$	$5.65E07$	$8.61E06$
SF (PGME)	$2.79E05$	$5.16E04$	$9.16E03$

TABLE 3.6: $\langle k \rangle^{\text{in}} + \langle k \rangle^{\text{out}}$ and $\langle k \rangle_{\text{key}}^{\text{in}} + \langle k \rangle_{\text{key}}^{\text{out}}$ in ER and SF networks.

Networks	N	M	$\langle k \rangle^{\text{in}} + \langle k \rangle^{\text{out}}$	$\langle k \rangle_{\text{key}}^{\text{in}} + \langle k \rangle_{\text{key}}^{\text{out}}$
ER	500	145	8.04	5.64
		175	8.04	5.74
		205	8.04	6.06
SF	500	145	8.07	4.78
		175	8.07	4.91
		205	8.07	5.08

FIGURE 3.7: Key nodes (red color) located in an ER network.

mean degrees of both ER and SF networks are around 4, which implies that the sum of the mean input and output degrees of the network $\langle k \rangle^{\text{in}} + \langle k \rangle^{\text{out}}$ is around 8. However, the sum of the mean input and output degrees of the key node set we identified, denoted as $\langle k \rangle_{\text{key}}^{\text{in}} + \langle k \rangle_{\text{key}}^{\text{out}}$, is only around 5.64 and 4.78, respectively. This suggests that the key node is unlikely to be the hub in the networks, as illustrated in Fig.3.7 (ER network with $N = 30$ and $\langle k \rangle = 4$). This finding is consistent with Slotine and Barabási's observation [18] as well as Menichetti et al.'s relationship between density of low degree nodes and controllability [84]. In [18], it is emphasized that the driver node set avoids the hub node set, and such an observation is said to be a “surprise”, which has caused quite a bit debates and discussions afterwards. In [84], Menichetti further points out that network controllability is mainly determined by the density of low in-degree and out-degree nodes. Our finding suggests that one needs to add control sources to the low-degree nodes instead of hub nodes for achieving a minimum control of networks.

Moreover, such a phenomenon could give some insights in social networks and it probably explains why nowadays governments of many countries in the world start to pay much attention to fight against some extremists who have not aroused enough attention before.

3.6.3 Comparisons between PGME and random allocation method (RAM)

Since there is no available scheme addressing the minimum-cost control of complex networks, we compare the performance of PGME with random allocation method (RAM). For RAM, firstly we apply maximum matching to obtain N_D driver nodes for ensuring the controllability of dynamic systems. Secondly, we randomly choose the rest $M - N_D$ nodes, and then we get M nodes which are regarded as the key node set in PGME. Thirdly, to reconstruct the solution of RAM, denoted as B^r , if node i is connected with j -th external control source, we set $B_{ij}^r = 1$, otherwise, $B_{ij}^r = 0$. Comparisons are made in both synthetic and real-life networks by considering the “numerical controllability transition” problem, stated in Remark 3.6. Results about an elementary stem ($N = 100$) and a real-life electronic network (circuits208) ($N = 128$) [25] are plotted in Fig. 3.8, with m_0 denoting the “numerical controllability transition” area. So the minimum number of control nodes for both PGME and random allocation method is considered as $N_D + m_0$, i.e., $M \geq N_D + m_0$. We repeat the simulations 100 times and compute the average results. From Fig. 3.8, it is observed that the control cost obtained by random allocation method can be improved by up to 8-9 orders of magnitude by using PGME in both synthetic and real-life networks. When $M = N$, the results of PGME and RAM become the same as each node becomes a control node.

Comparisons between PGME and RAM are also made on ER and SF networks, as shown in Fig. 3.9. In Figure 3.9(a), with the number of nodes $N = 100$ and fixed number of external input $M = 40$, minimum control cost is decreasing when the mean degree is increasing. This is because that when more edges are added, the longest control chain[85] are getting shorter, resulting in decrease of control cost. Clearly, the control cost obtained by PGME is much lower than RAM.

Figure 3.9(b) compares control cost obtained by PGME and RAM with different mean degree $\langle k \rangle$. We set $N = 100$ and M as the minimum number of driver nodes

FIGURE 3.8: The comparisons between PGME and random allocation method in both synthetic and real-life networks. (a) Elementary stem with $N = 100$. (b) Real-life electronic network

plus a constant, i.e. $M = N_D + \Delta M$ with ΔM fixed. For relatively low-degree networks, the minimum number of driver nodes N_D decreases remarkably, leading the increase of control cost. However, as $\langle k \rangle$ continues to be higher, the networks become denser, compensating for an insignificant drop of N_D . Then the control cost starts to decrease. Compared with RAM, PGME can drastically reduce the control cost when $\langle k \rangle$ is relatively small. When $\langle k \rangle$ is increasing, the values of control cost achieved by RAM and PGME tend to approach to each other, implying that key nodes can be approximately obtained by RAM for dense ER networks.

FIGURE 3.9: The comparisons between PGME and RAM on SF and ER networks. (a) SF networks $N = 100$, $M = 40$. (b) ER networks with $N = 100$.

3.6.4 Strength-based random allocation method (S-RAM)

Note that in RAM, each element of B^r is either one or zero depending on whether a node is connected to an external control source or not. We also consider a Strength-based random allocation method (S-RAM) in which the zero element of B^r is fixed as zero while the non-zero element of B^r becomes a variable. Therefore, B^r , a sparse matrix constructed by RAM, can be considered as the initial matrix of S-RAM. After obtaining the derivative $\frac{\partial \mathcal{E}(B^r)}{\partial B^r}$, we could also apply projected gradient method in S-RAM under the constraint that $\text{tr}([B^r]^T B^r) = M$ (just as PGM did) to determine the optimal strength of each key node connected to a particular control source. Comparisons between RAM, S-RAM and PGME in various real-life networks are shown in Table 3.7. It is observed that S-RAM could achieve a lower cost than RAM when the strength of each connection becomes selectable, but PGME indeed reduces the control cost significantly compared with both RAM and S-RAM as more freedom is allowed.

Finally it is worth mentioning that, starting from any feasible initial (A, B_0) , the network controllability can be guaranteed as the control energy determined by (A, B_k) keeps non-increasing in each iteration of PGM. It is however difficult to prove that the same conclusion holds for (A, B^s) . Observing in our extensive simulation studies that (A, B^s) always generates a feasible solution with comparable performance as that of (A, B^*) , we have good reasons to expect that (A, B^s) would almost always guarantee the controllability of the system, especially when we anyway have to select a large enough number of key nodes to avoid the numerical controllability transition problem.

3.6.5 Robustness test of PGME

In real-world applications, sometimes the information of matrix A cannot be exactly obtained. In this subsection, we try to test the robustness of the proposed algorithm by numerical simulations when there exists modelling discrepancy from the actual network.

Recall that one of the important findings is that key nodes tend to divide elementary stems/cycles/dilations equally. Suppose that the network in Fig. 3.10(a) is the actual network to be considered. Key nodes obtained by proposed algorithm are

TABLE 3.7: Energy consumed in ER and SF networks for RAM, and PGME.

Dataset	Nodes	Edges	M	RAM	S-RAM	PGME
ER	100	299	50	3.95E05	4.53E04	2.06E03
	500	1492	263	1.64E06	6.80E04	8.18E03
SF	100	341	41	1.39E07	1.53E06	2.64E04
	500	1548	244	2.56E07	9.98E06	4.26E04
Maspalomas[1]	30	87	15	2.46E05	4.37E04	3.16E02
Rhode[1]	25	58	15	4.37E04	5.48E03	0.83E02
circuit-s208[25]	122	289	77	6.47E06	1.38E04	4.35E03
circuit-s838[25]	512	819	297	2.85E09	2.45E07	4.67E04
phys-discuss[86]	231	498	185	1.31E03	5.39E02	3.18E02
phys-friend[87]	228	506	118	1.07E06	4.01E05	1.43E04
Florida[88]	128	2106	79	2.48E07	3.72E03	2.02E04
celegans[81]	297	2345	170	1.03E05	4.92E04	6.28E02

FIGURE 3.10: **The robustness test when designing B with A pre-given.**
Key nodes are marked by red color.

presented by red color. The network in Fig.3.10(b) is a model of that in Fig.3.10(a), where the edge Node 17 $\rightarrow$ Node 18 is missing. With the proposed algorithms, it can be seen that other key nodes remain the same except the ones in the local area of the missing edge, i.e, the cycle {Node 15 $\rightarrow$ Node 16 $\rightarrow$ Node 17 $\rightarrow$ Node 18 $\rightarrow$ Node 15}. Specifically, Node 15 and Node 17 are key nodes in 3.10(a) while Node 16 and Node 18 are key nodes in 3.10(b). From this example, we note that the propose algorithm has certain robustness with respect to modelling errors.

3.7 Conclusion

How to select the key node set for minimizing the energy of controlling complex networks is an important and promising work. In this chapter, we have formulated a mathematical model to address this issue. PGM and PTM are proposed to solve the problem. We suggest to employ PGM instead of PTM in real applications due to its computational efficiency. In addition, PGM extension is proposed to identify the key node set in Monte-Carlo realizations. Through extensive experiments, we obtain the following three findings when the energy attains minimum: 1) Key nodes distribute as evenly as possible in elementary stems/circles/dilations. More specifically, for an elementary stem, importance indices present quasi-periodic distribution while each node's importance is uniformly distributed in an elementary cycle. An elementary dilation can be divided into two stems and in each stem importance indices show similar pattern as an elementary stem. 2) When the importance index is high, its occurrence rate of a node being a key node is also high, and vice versa. 3) The hubs are of less likely to be selected as key nodes such that low degree nodes may be more important than hub nodes. These findings provide meaningful physical insights for the optimal control of complex networks and shall be very important to design efficient algorithms for selecting the key node set in large scale complex networks in the future.

Chapter 4

Key Node Selection in Directed Complex Networks via Convex Optimization

In last chapter, key node selection problem under minimum-cost control scheme is investigated based on matrix differentiation and PGME is proposed based on the definition of importance index. We find that key nodes are related to nodes' degree. Motivated by this finding, in this chapter, we revisit key node selection problem and propose three algorithms step by step with degree information, which are comparable with PGME. Specifically, through a newly formulated optimization model with Boolean constraints, which can be relaxed to a convex problem eventually, inexact alternating direction method of multipliers (IADMM) is proposed and convergence property is theoretically established. Based on the degree distribution, an extension method named degree-based IADMM (D-IADMM) is proposed such that key nodes are pinpointed. In addition, with the technique of local optimization, the performance of D-IADMM is greatly improved. The comparison between the proposed algorithms and PGME are represented on different networks ranging from Erdős-Rényi networks, scale-free networks to some real-life networks.

4.1 Introduction

Identifying a set of key nodes to be connected with external control sources so that a given control cost is minimized is an important problem. As pointed in [20, 89], such an intractable combinatorial problem is NP-hard. A straightforward idea is to list all possible combinations with the given size and select the one with minimum control cost. However, it is almost impossible for large scale complex networks due to high computational cost. Recently, in [39], the smallest eigenvalue of controllability Gramian is adopted as a metric to derive a lower bound on control cost for a discrete linear system. But no general algorithm is proposed to achieve such a lower bound. In [52, 61], the authors show submodularity of controllability metrics proposed in [53]. Nevertheless, to find the node set with minimum control cost, a set of nodes that provides controllability of whole network should be given first. Then extra nodes are added to the given node set by optimizing the energy metrics. In Chapter 3, we can identify such a node set by the definition of importance indices of nodes based on the solution obtained by PGM. It is found that the key nodes are related to network's degree distribution. Departing from these works, in this chapter, we consider key node selection problem for finite-horizon continuous-time linear system from the point view of convex optimization by taking advantage of nodes' degree.

Specifically, to minimize the average control cost, we first formulate the optimization model in boolean constraints where nodes and external controllers are in one-to-one connections. By relaxing the constraints to their convex hull, a convex problem is obtained. Then an efficient algorithm called inexact alternating direction method of multipliers (IADMM) is developed to solve the problem iteratively. The convergence of IADMM is theoretically analyzed. Through rounding the solution by exploiting degree information, degree-based IADMM (D-IADMM) is proposed such that the key nodes are pinpointed. To further improve the performance of D-IADMM, a greedy algorithm named local optimization is employed. The proposed algorithms are validated by Erdős-Rényi networks, scale-free networks as well as real-life networks. It is worthwhile to highlight the differences of D-IADMM (IADMM) with PGME (PGM) proposed in Chapter 3 as follows:

1. The main idea of IADMM is to decompose the original problem into several subproblems while PGM is to obtain the solution by projecting the gradient

of $\mathcal{E}(B)$ on the space orthogonal to the gradient of normal function defined in Chapter 3.

2. By taking degree information into account, the continuous solutions are rounded to binary values by D-IADMM. For PGME, key nodes are pinpointed are based on the importance indices of nodes.
3. IADMM is proposed to solve a relaxed convex optimization problem while PGM is introduced to solve a non-convex problem.

The rest of this chapter is organized as follows. In Section 4.2, we formulate the problem of selecting M key nodes in one-to-one connection with external control sources. In Section 4.3, we relax the model to a convex problem and solve it by IADMM. The convergence of IADMM is also analyzed. Section 4.4 deals with the problem of key nodes determination, where D-IADMM and LD-IADMM are proposed. In section 4.5, we validate our proposed algorithms through extensive numerical examples. Comparison with PGME, which is proposed in Chapter 3, is presented. Section 4.6 gives the conclusion.

4.2 Problem Formulation

As mentioned in Introduction, we consider how to identify M key nodes in one-to-one connection with external control sources, such that the states of a system can be driven to the origin at $t = t_f$, i.e., $\mathbf{x}(t_f) = \mathbf{0} = [0, \dots, 0]^T$, subject to the condition that a given control cost is minimized. Considering the LTI system described in (2.1), the following assumptions are made to achieve this objective.

Assumption 4.1. The number of external control sources is sufficient such that (A, B) is controllable at initial time.

Assumption 4.2. $BB^T = \text{diag}(\mathbf{b})$ where $\mathbf{b} = [b_1, b_2, \dots, b_N]^T$ is an N -dimensional binary vector. $\text{diag}(\mathbf{b})$ is an $N \times N$ square diagonal matrix with all elements of vector $\mathbf{b}$ on its main diagonal. Under this assumption, if $b_i = 1$, the node i is connected with external control source. That is to say, node i is a key node.

Similar to the problem formulation in Chapter 3, the average control energy is given by

$$\begin{aligned}
\mathcal{E}(B) &= \text{tr} \left(e^{A^T t_f} W_B^{-1} e^{A t_f} \mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] \right) \\
&= \text{tr} \left(e^{A^T t_f} W_B^{-1} e^{A t_f} \right) \\
&= \text{tr} \left(\left(\int_0^{t_f} e^{A(t-t_f)} B B^T e^{A^T(t-t_f)} dt \right)^{-1} \right) \\
&= \text{tr} \left(\left(\int_0^{t_f} e^{-A t} B B^T e^{-A^T t} dt \right)^{-1} \right)
\end{aligned} \tag{4.1}$$

where $\mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T]$ is regarded as an identity matrix by assuming that each element of initial state $\mathbf{x}_0 = [x_1(0), \dots, x_N(0)]^T$ is an identically independently distributed (i.i.d) variable with zero mean and variance 1.

Under Assumption 4.2, the minimization of the average control cost finally leads to the following matrix function optimization model

$$\begin{aligned}
\arg \min_{\mathbf{b}} \quad & \mathcal{E}(\mathbf{b}) = \text{tr} \left(\int_0^{t_f} e^{-A t} D_{\mathbf{b}} e^{-A^T t} dt \right)^{-1} \\
s.t. \quad & \mathbf{b}^T \mathbb{1} = M, \quad b_i \in \{0, 1\}, \quad i = 1, \dots, N
\end{aligned} \tag{4.2}$$

where $D_{\mathbf{b}} = \text{diag}(\mathbf{b})$ and $\mathbb{1}$ is all-ones vector.

4.3 Convex Relaxation for Key Nodes Selection

In this section, we first relax the optimization problem (4.2) into a convex model. Then an efficient and effective algorithm called IADMM is developed. By taking the information of nodes' degree, we round the solution obtained by IADMM such that D-IADMM is proposed and key nodes are identified. Moreover, we further employ a greedy method, named local optimization on D-IADMM to improve the performance.

4.3.1 Inexact alternating direction multipliers of method

Note that (4.2) is a non-convex problem. Now we relax this constraint to its convex hull $0 \leq b_i \leq 1$, which yields to the following optimization problem,

$$\begin{aligned} \arg \min_{\mathbf{b}} \quad & \mathcal{E}(\mathbf{b}) = \text{tr} \left(\int_0^{t_f} e^{-At} D_{\mathbf{b}} e^{-A^T t} dt \right)^{-1} \\ \text{s.t.} \quad & \mathbf{b}^T \mathbf{1} = M, \quad 0 \leq b_i \leq 1, \quad i = 1, \dots, N \end{aligned} \quad (4.3)$$

Before we solve this problem, we present the following lemmas and theorems.

Lemma 4.1. [90] *Let X, Y be $\mathbb{R}^{n \times n}$ Hermitian matrices. If X is positive definite, then there is a nonsingular $S \in \mathbb{R}^{n \times n}$ such that $X = SS^*$ and $Y = S\Lambda S^*$, in which Λ is real diagonal. The inertias of Y and Λ are the same, so Λ is nonnegative diagonal (positive diagonal) if Y is positive semidefinite (positive definite). The main diagonal entries of Λ are the eigenvalues of the diagonalizable matrix $X^{-1}Y$.*

Theorem 4.1. The optimization model (4.3) is strictly convex. That is, $\mathcal{E}(\mathbf{b})$ is strictly convex on $0 \leq b_i \leq 1$ ($i = 1, \dots, N$), $\mathbf{b}^T \mathbf{1} = M$.

Proof. For convenience, let $R_i(t)$ be the i -th column of e^{-At} . Then $\mathcal{E}(\mathbf{b})$ can be written as

$$\begin{aligned} \mathcal{E}(\mathbf{b}) &= \text{tr} \left(\sum_{i=1}^N \int_0^{t_f} b_i R_i(t) R_i^T(t) dt \right)^{-1} \\ &= \text{tr} \left(\sum_{i=1}^N b_i \int_0^{t_f} R_i(t) R_i^T(t) dt \right)^{-1} \end{aligned} \quad (4.4)$$

To prove the optimization problem (4.3) is strictly convex, we must show that

$$\mathcal{E}(\alpha \mathbf{a} + (1 - \alpha) \mathbf{b}) \leq \alpha \mathcal{E}(\mathbf{a}) + (1 - \alpha) \mathcal{E}(\mathbf{b})$$

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

Based on Eq. (4.4), we have

$$\begin{aligned}
& \mathcal{E}(\alpha \mathbf{a} + (1 - \alpha) \mathbf{b}) \\
&= \text{tr} \left(\sum_{i=1}^N (\alpha a_i + (1 - \alpha) b_i) \int_0^{t_f} R_i(t) R_i^T(t) dt \right)^{-1} \\
&= \text{tr} \left(\alpha \sum_{i=1}^N a_i \int_0^{t_f} R_i(t) R_i^T(t) dt + (1 - \alpha) \sum_{i=1}^N b_i \int_0^{t_f} R_i(t) R_i^T(t) dt \right)^{-1}
\end{aligned} \tag{4.5}$$

For simplicity, let $R' = \sum_{i=1}^N a_i \int_0^{t_f} R_i(t) R_i^T(t) dt$ and $R'' = \sum_{i=1}^N b_i \int_0^{t_f} R_i(t) R_i^T(t) dt$. With the Assumption 4.1, it is obvious that R' and R'' are positive definite. According to Lemma 4.1, we can write $R' = SS^T$, $R'' = S\Lambda S^T$ for some nonsingular matrix S and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ with each $\lambda_i > 0$. Let $s_1, \dots, s_n$ be positive main diagonal entries of the positive definite matrix $S^{-1}S^{-T}$. Then it can be obtained that

$$\begin{aligned}
\mathcal{E}(\alpha \mathbf{a} + (1 - \alpha) \mathbf{b}) &= \text{tr} (\alpha R' + (1 - \alpha) R'')^{-1} \\
&= \text{tr} (\alpha SS^T + (1 - \alpha) S\Lambda S^T)^{-1} \\
&= \text{tr} (S(\alpha + (1 - \alpha)\Lambda)S^T)^{-1} \\
&= \text{tr} ((\alpha + (1 - \alpha)\Lambda)^{-1} S^{-1}S^{-T}) \\
&= \sum_{i=1}^N (\alpha + (1 - \alpha)\lambda_i)^{-1} s_i \\
&\leq \sum_{i=1}^N (\alpha s_i + (1 - \alpha)\lambda_i^{-1} s_i) \\
&= \alpha \text{tr}(S^{-1}S^{-T}) + (1 - \alpha) \text{tr}(\Lambda^{-1}S^{-1}S^{-T}) \\
&= \alpha \text{tr}(SS^T)^{-1} + (1 - \alpha) \text{tr}(S\Lambda S^T)^{-1} \\
&= \alpha \text{tr}(R')^{-1} + (1 - \alpha) \text{tr}(R'')^{-1} \\
&= \alpha \mathcal{E}(\mathbf{a}) + (1 - \alpha) \mathcal{E}(\mathbf{b})
\end{aligned} \tag{4.6}$$

the equality holds iff $\lambda_i = 1$, which implies $\mathbf{a} = \mathbf{b}$.

□

Lemma 4.2. *The gradient of $\mathcal{E}(\mathbf{b})$ in (4.2) with regards to $\mathbf{b}$ yields that*

$$\begin{aligned}
 & \frac{\partial \operatorname{tr} \left(\int_0^{t_f} e^{-At} D_{\mathbf{b}} e^{-A^T t} dt \right)^{-1}}{\partial \mathbf{b}_i} \\
 &= - \left(\int_0^{t_f} e^{-A^T t} H_{\mathbf{b}}^{-2} e^{-At} dt \right)_{ii}
 \end{aligned} \tag{4.7}$$

Proof. For convenience, let $H_{\mathbf{b}} = \int_0^{t_f} e^{-At} D_{\mathbf{b}} e^{-A^T t} dt$. Based on Lemmas 2.1 and 2.2, we have

$$\begin{aligned}
 & \frac{\partial \operatorname{tr} \left(\int_0^{t_f} e^{-At} D_{\mathbf{b}} e^{-A^T t} dt \right)^{-1}}{\partial [D_{\mathbf{b}}]_{ii}} \\
 &= \delta_{mn} \cdot \frac{\partial [H_{\mathbf{b}}^{-1}]_{mn}}{\partial [H_{\mathbf{b}}]_{fr}} \cdot \frac{\partial [H_{\mathbf{b}}]_{fr}}{\partial [D_{\mathbf{b}}]_{ii}} \\
 &= -[H_{\mathbf{b}}^{-1}]_{mf} \cdot [H_{\mathbf{b}}^{-1}]_{rm} \cdot \frac{\partial \left[\int_0^{t_f} e^{-At} B B^T e^{-A^T t} dt \right]_{fr}}{\partial [D_{\mathbf{b}}]_{ii}} \\
 &= -[H_{\mathbf{b}}^{-1}]_{mf} \cdot [H_{\mathbf{b}}^{-1}]_{rm} \cdot \int_0^{t_f} [e^{-At}]_{fs} \cdot \frac{\partial [D_{\mathbf{b}}]_{ss}}{\partial [D_{\mathbf{b}}]_{ii}} \cdot [e^{-A^T t}]_{sr} dt \\
 &= -[H_{\mathbf{b}}^{-1}]_{mf} \cdot [H_{\mathbf{b}}^{-1}]_{rm} \cdot \int_0^{t_f} [e^{-At}]_{fs} \cdot \delta_{si} \cdot [e^{-A^T t}]_{sr} dt \\
 &= - \left[\int_0^{t_f} e^{-A^T t} H_{\mathbf{b}}^{-2} e^{-At} dt \right]_{ii}
 \end{aligned} \tag{4.8}$$

Thus this lemma holds. $\square$

As the optimization problem (4.3) is demonstrated to be strictly convex in Theorem 4.1 and the gradient of the objective function is given in Lemma 4.2, now it is time for us to introduce inexact alternating method of multipliers (IADMM) to solve the problem (4.3).

Define $\mathcal{S}_1 := \{\mathbf{b} \mid 0 \leq b_i \leq 1\}$ and $\mathcal{S}_2 := \{\mathbb{1}^T \mathbf{b} = M\}$. $\phi_1(\mathbf{b})$, $\phi_2(\mathbf{b})$ are indicator functions of $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively, which are given by

$$\phi_1(\mathbf{b}) = \begin{cases} 0 & \mathbf{b} \in \mathcal{S}_1 \\ \infty & \mathbf{b} \notin \mathcal{S}_1 \end{cases}$$

$$\phi_2(\mathbf{b}) = \begin{cases} 0 & \mathbf{b} \in \mathcal{S}_2 \\ \infty & \mathbf{b} \notin \mathcal{S}_2 \end{cases}$$

Then problem (4.3) is equivalent to

$$\arg \min_{\mathbf{b}} \quad \mathcal{E}(\mathbf{b}) + \phi_1(\mathbf{b}) + \phi_2(\mathbf{b}) \quad (4.9)$$

Now we introduce an additional variable $\mathbf{z}$ and rewrite (4.9) as the following optimization problem

$$\begin{aligned} \arg \min_{\mathbf{b}} \quad & f(\mathbf{b}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{b} - \mathbf{z} = 0 \end{aligned} \quad (4.10)$$

where $f(\mathbf{b}) = \mathcal{E}(\mathbf{b}) + \phi_1(\mathbf{b})$ and $g(\mathbf{z}) = \phi_2(\mathbf{z})$.

Note that the augmented Lagrangian associated with (4.10) is

$$\mathcal{L}_\rho(\mathbf{b}, \mathbf{z}, \boldsymbol{\lambda}) = f(\mathbf{b}) + g(\mathbf{z}) + \boldsymbol{\lambda}^T(\mathbf{b} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{b} - \mathbf{z}\|^2 \quad (4.11)$$

where $\boldsymbol{\lambda}$ is the Lagrangian multiplier and $\rho > 0$.

To find the optimal solution of (4.10), we develop IADMM with the help of augmented augmented Lagrangian function (4.11) through the following alternating iterations :

$$\mathbf{b}^{k+1} = \Pi_{\mathcal{S}_1} \left(\mathbf{b}^k - \mu \left(\nabla \mathcal{E}(\mathbf{b}^k) + \boldsymbol{\lambda}^k + \rho(\mathbf{b}^k - \mathbf{z}^k) \right) \right) \quad (4.12a)$$

$$\mathbf{z}^{k+1} = \arg \min_{\mathbf{z}} \left(g(\mathbf{z}) - \boldsymbol{\lambda}^{kT} \mathbf{z} + \frac{\rho}{2} \|\mathbf{b}^{k+1} - \mathbf{z}\|^2 \right) \quad (4.12b)$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{b}^{k+1} - \mathbf{z}^{k+1}) \quad (4.12c)$$

where μ is a step size and $\Pi_{\mathcal{S}_1}$ is the Euclidean projection onto the set $\mathcal{S}_1$. The update of $\boldsymbol{\lambda}$ is actually the dual ascent step. Note that if ADMM [91] were applied here, $\mathbf{b}$ should be updated through minimizing the subproblem, i.e. $\mathbf{b}^{k+1} :=$

$\arg \min_{\mathbf{b}} \left(f(\mathbf{b}) + \boldsymbol{\lambda}^{kT} \mathbf{b} + \frac{\rho}{2} \|\mathbf{b}^{k+1} - \mathbf{z}\|^2 \right)$, which is of high computational complexity. Instead of obtaining the exact minimum point of the subproblem, here we update $\mathbf{b}$ inexactly by a single gradient step in (4.12a), yet the convergence is still ensured which will be presented later. Therefore, we term the proposed algorithm as inexact alternating method of multipliers (IADMM).

For $\mathbf{z}$ -minimization step in (4.12), it is equivalent to the following optimization problem

$$\begin{aligned} \arg \min_{\mathbf{z}} \quad & -\boldsymbol{\lambda}^{kT} \mathbf{z} + \frac{\rho}{2} \|\mathbf{b}^{k+1} - \mathbf{z}\|^2 \\ \text{s.t.} \quad & \mathbb{1}^T \mathbf{z} = M \end{aligned} \quad (4.13)$$

The Lagrangian function of (4.13) is given by

$$\mathcal{L}_\kappa(\mathbf{z}) = -\boldsymbol{\lambda}^{kT} \mathbf{z} + \frac{\rho}{2} \|\mathbf{b}^{k+1} - \mathbf{z}\|^2 + \kappa(\mathbb{1}^T \mathbf{z} - M)$$

Based on the KKT (Karush-Kuhn-Tucker) condition [92], we can obtain that

$$\begin{cases} \frac{\partial \mathcal{L}_\kappa(\mathbf{z})}{\partial \mathbf{z}} = \mathbf{0} \\ \frac{\partial \mathcal{L}_\kappa(\mathbf{z})}{\partial \kappa} = 0 \end{cases} \quad (4.14)$$

By solving the equations in (4.14), the exact solution for $\mathbf{z}$ -update in each iteration is given as follows

$$\mathbf{z}^{k+1} = (I - \frac{\mathbb{1} \mathbb{1}^T}{N})(\mathbf{b}^{k+1} + \frac{\boldsymbol{\lambda}^k}{\rho}) + \frac{M}{N} \mathbb{1} \quad (4.15)$$

The whole iterative process of (4.12) will be terminated until the primal and dual residuals are sufficiently small, as specified by

$$\begin{aligned} \|\mathbf{b}^{k+1} - \mathbf{z}^{k+1}\|^2 &< \epsilon^{\text{pri}} \\ \|\rho(\mathbf{z}^{k+1} - \mathbf{z}^k)\|^2 &< \epsilon^{\text{dual}} \end{aligned} \quad (4.16)$$

where ϵ^{pri} and ϵ^{dual} are positive small constants specified by user.

Denote the solution obtained by the proposed algorithm IADMM as $\mathbf{b}^{I*}$. The above iterative process is summarize in Algorithm 4.1. The relevant convergence analysis of IADMM will be presented later .

Algorithm 4.1 Inexact alternating direction method of multipliers (IADMM)

Input: Functions $f(\mathbf{b})$ and $g(\mathbf{z})$, parameter ρ .

Output: $\mathbf{b}^{I*}$

1: **Initialization:** Initialize $\mathbf{b}$, $\mathbf{z}$, $\boldsymbol{\nu}$

2: **repeat**

3:

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

$$\mathbf{z}^{k+1} = \left(I - \frac{\mathbb{1} \mathbb{1}^T}{N} \right) (\mathbf{b}^{k+1} + \frac{\boldsymbol{\lambda}^k}{\rho}) + \frac{M}{N} \mathbb{1}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{b}^{k+1} - \mathbf{z}^{k+1})$$

4: **until** meet certain stopping criterion

5: **return** $\mathbf{b}^{I*} = \mathbf{b}$

4.3.2 Convergence analysis of IADMM

In this subsection, we mainly study the convergence properties of the proposed algorithm IADMM. Before proceeding to the main result, we introduce the following preliminaries and lemma.

Subdifferential and monotone operator[93] On Euclidean space $\mathcal{H}$, an operator is a set-valued mapping, i.e. $T : \mathcal{H} \rightarrow 2^{\mathcal{H}}$. Given a convex function f , the subdifferential at $\mathbf{x} \in \text{dom } f$ is a set-valued operator, $\partial f := \{p \in \mathcal{H} | f(\mathbf{y}) \geq f(\mathbf{x}) + p^T(\mathbf{y} - \mathbf{x}), \forall \mathbf{y} \in \mathcal{H}\}$.

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

$$\forall (\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}') \in \text{gra } T, \langle \mathbf{x} - \mathbf{y}, \mathbf{x}' - \mathbf{y}' \rangle \geq 0.$$

A monotone operator T is maximal monotone if no monotone operator T' such that $\text{gra } T \subsetneq \text{gra } T'$. The best-known example of maximal monotone operator is the subdifferential ∂f of a closed proper convex function. The resolvent of T

is $R_{\tau T} = (I + \tau T)^{-1}$. An application of the resolvent operator is the proximity operator which is defined as

$$\mathbf{Prox}_{\tau f}(v) = \arg \min_{\mathbf{x} \in \mathcal{H}} f(\mathbf{x}) + \frac{1}{2\tau} \|\mathbf{x} - \mathbf{v}\|^2.$$

Lemma 4.3. *The update of $\mathbf{b}$ in (4.12a) is equivalent to*

$$\mu \partial \phi_1(\mathbf{b}^{k+1}) = \mathbf{b}^k - \mathbf{b}^{k+1} - \mu \left(\nabla \mathcal{E}(\mathbf{b}^k) + \boldsymbol{\lambda}^k + \rho(\mathbf{b}^k - \mathbf{z}^k) \right)$$

Proof. As projection on the convex set is equivalent to computing the proximal operator of its indicator function [94], the update of $\mathbf{b}$ in (4.12a) can be written as

$$\begin{aligned} \mathbf{b}^{k+1} &= \Pi_{\mathcal{S}_1} \left(\mathbf{b}^k - \mu \left(\nabla \mathcal{E}(\mathbf{b}^k) + \boldsymbol{\lambda}^k + \rho(\mathbf{b}^k - \mathbf{z}^k) \right) \right) \\ &= \mathbf{Prox}_{\mu \phi_1(\mathbf{b})} \left(\mathbf{b}^k - \mu \left(\nabla \mathcal{E}(\mathbf{b}^k) + \boldsymbol{\lambda}^k + \rho(\mathbf{b}^k - \mathbf{z}^k) \right) \right) \\ &= \arg \min_{\mathbf{b}} \phi_1(\mathbf{b}) + \frac{1}{2\mu} \left\| \mathbf{b} - \left(\mathbf{b}^k - \mu \left(\nabla \mathcal{E}(\mathbf{b}^k) + \boldsymbol{\lambda}^k + \rho(\mathbf{b}^k - \mathbf{z}^k) \right) \right) \right\|^2 \end{aligned}$$

Therefore, with the optimal condition, we have

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

Rearranging the terms of above expression yields the desired result.

□

Note that $\nabla \mathcal{E}(\mathbf{b})$ is continuously differentiable on the compact set $\mathcal{S}_1$. Therefore, $\nabla \mathcal{E}(\mathbf{b})$ is Lipschitz continuous in the domain $\mathcal{S}_1$, that is

$$\|\nabla \mathcal{E}(\mathbf{b}) - \nabla \mathcal{E}(\mathbf{b}')\| \leq L \|\mathbf{b} - \mathbf{b}'\| \quad \forall \mathbf{b}, \mathbf{b}' \in \mathcal{S}_1 \quad (4.17)$$

Now we are ready to introduce the following theorem.

Theorem 4.2. Given $\mu < 1/(\rho + L)$, the sequence $\{(\mathbf{b}^k, \mathbf{z}^k, \boldsymbol{\lambda}^k)\}$ generated by the proposed IADMM will converge to the saddle point $(\mathbf{b}^*, \mathbf{z}^*, \boldsymbol{\lambda}^*)$ of $\mathcal{L}_\rho(\mathbf{b}, \mathbf{z}, \boldsymbol{\lambda})$.

Proof. Let us first consider the optimality condition of problem (4.10). Let $G = \frac{\mathbb{1}\mathbb{1}^T}{N}$. Then the primal and dual optimality condition satisfied by individual steps in Algorithm 4.1 can be expressed as follows

$$\begin{cases} \mathbb{1}^T \mathbf{b}^* - M = 0 & (\text{primal feasibility}) \\ \mathbf{b}^* - \mathbf{z}^* = 0 & (\text{primal feasibility}) \\ \partial f(\mathbf{b}^*) + \boldsymbol{\lambda}^* = 0 & (\text{dual feasibility}) \\ \partial g(\mathbf{b}^*) - \boldsymbol{\lambda}^* = 0 & (\text{dual feasibility}) \end{cases} \quad (4.18)$$

where $\partial f(\mathbf{b}^*) = \nabla \mathcal{E}(\mathbf{b}^*) + \partial \phi_1(\mathbf{b}^*)$.

Apparently, $G\mathbf{b}^* = \frac{M}{N}\mathbb{1}$. Then we substitute (4.12b) into (4.12c) and obtain that

$$\boldsymbol{\lambda}^{k+1} = \rho G \mathbf{b}^{k+1} + G \boldsymbol{\lambda}^k - \rho G \mathbf{b}^* \quad (4.19)$$

Multiplying G on both sides of (4.19) and using the fact $G^2 = G$, it follows that

$$\begin{cases} (I - G)\boldsymbol{\lambda}^k = 0 \\ \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k = \rho G(\mathbf{b}^{k+1} - \mathbf{b}^*) \end{cases} \quad (4.20)$$

By Lemma 4.3, we have

$$\begin{aligned} \mu \partial f(\mathbf{b}^{k+1}) &= \mu \nabla \mathcal{E}(\mathbf{b}^{k+1}) + \mu \partial \phi_1(\mathbf{b}^{k+1}) \\ &= \mathbf{b}^k - \mathbf{b}^{k+1} + \mu \left(\nabla \mathcal{E}(\mathbf{b}^{k+1}) - \nabla \mathcal{E}(\mathbf{b}^k) \right) - \mu \boldsymbol{\lambda}^k - \mu \rho (\mathbf{b}^k - \mathbf{z}^k). \end{aligned} \quad (4.21)$$

Knowing that ∂f is maximal monotone, based on $\partial f(\mathbf{b}^*) + \boldsymbol{\lambda}^* = 0$ from the optimality condition (4.18) and properties in (4.20), we have

$$\begin{aligned}
& \mu \langle \partial f(\mathbf{b}^{k+1}) - \partial f(\mathbf{b}^*), \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \left\langle \mathbf{b}^k - \mathbf{b}^{k+1} + \mu \left(\nabla \mathcal{E}(\mathbf{b}^{k+1}) - \nabla \mathcal{E}(\mathbf{b}^k) \right) + \mu(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k) - \mu\rho(\mathbf{b}^k - \mathbf{z}^k), \mathbf{b}^{k+1} - \mathbf{b}^* \right\rangle \\
&= \left\langle \mathbf{b}^k - \mathbf{b}^{k+1} + \mu \left(\nabla \mathcal{E}(\mathbf{b}^{k+1}) - \nabla \mathcal{E}(\mathbf{b}^k) \right) + \mu(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k) + \mu(\boldsymbol{\lambda}^{k-1} - \boldsymbol{\lambda}^k), \mathbf{b}^{k+1} - \mathbf{b}^* \right\rangle \\
&= \langle \mathbf{b}^k - \mathbf{b}^{k+1}, \mathbf{b}^{k+1} - \mathbf{b}^* \rangle + \mu \langle \nabla \mathcal{E}(\mathbf{b}^{k+1}) - \nabla \mathcal{E}(\mathbf{b}^k), \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&\quad + \mu \langle \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k, \mathbf{b}^{k+1} - \mathbf{b}^* \rangle + \mu \langle \boldsymbol{\lambda}^{k-1} - \boldsymbol{\lambda}^k, \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&\geq 0
\end{aligned} \tag{4.22}$$

For the first term of the above inequality, we have

$$\begin{aligned}
& \langle \mathbf{b}^k - \mathbf{b}^{k+1}, \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \frac{1}{2} \left(\|\mathbf{b}^k - \mathbf{b}^*\|^2 - \|\mathbf{b}^{k+1} - \mathbf{b}^*\|^2 - \|\mathbf{b}^{k+1} - \mathbf{b}^k\|^2 \right)
\end{aligned} \tag{4.23}$$

With the identity of Bregman distance, it can be obtained that

$$\begin{aligned}
& \langle \nabla \mathcal{E}(\mathbf{b}^{k+1}) - \nabla \mathcal{E}(\mathbf{b}^k), \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \mathcal{D}_{\mathcal{E}}(\mathbf{b}^*, \mathbf{b}^{k+1}) - \mathcal{D}_{\mathcal{E}}(\mathbf{b}^*, \mathbf{b}^k) + \mathcal{D}_{\mathcal{E}}(\mathbf{b}^{k+1}, \mathbf{b}^k)
\end{aligned} \tag{4.24}$$

where $\mathcal{D}_{\mathcal{E}}(\mathbf{x}, \mathbf{x}') = \mathcal{E}(\mathbf{x}) - \mathcal{E}(\mathbf{x}') - \nabla \mathcal{E}(\mathbf{x}')^T(\mathbf{x} - \mathbf{x}')$ is the Bregman distance between $\mathbf{x}$ and $\mathbf{x}'$ [95].

With (4.20), we have

$$\begin{aligned}
& \langle \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k, \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \langle G(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k), \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \langle \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k, G(\mathbf{b}^{k+1} - \mathbf{b}^*) \rangle \\
&= \frac{1}{\rho} \langle \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k, \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k \rangle \\
&= \frac{1}{2\rho} \left(\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 + \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 \right)
\end{aligned} \tag{4.25}$$

$$\begin{aligned}
& \langle \boldsymbol{\lambda}^{k-1} - \boldsymbol{\lambda}^k, \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \langle G(\boldsymbol{\lambda}^{k-1} - \boldsymbol{\lambda}^k), \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \rho \langle G(\mathbf{b}^* - \mathbf{b}^k), \mathbf{b}^{k+1} - \mathbf{b}^* \rangle \\
&= \frac{\rho}{2} \|\mathbf{b}^{k+1} - \mathbf{b}^k\|_G^2 - \frac{\rho}{2} \|\mathbf{b}^k - \mathbf{b}^*\|_G^2 - \frac{\rho}{2} \|\mathbf{b}^{k+1} - \mathbf{b}^*\|_G^2 \\
&= \frac{\rho}{2} \|\mathbf{b}^{k+1} - \mathbf{b}^k\|_G^2 - \frac{1}{2\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 - \frac{1}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2
\end{aligned} \tag{4.26}$$

where $\|\mathbf{b}^{k+1} - \mathbf{b}^k\|_G = \sqrt{\langle G(\mathbf{b}^{k+1} - \mathbf{b}^k), \mathbf{b}^{k+1} - \mathbf{b}^k \rangle}$ and $\|\cdot\|_G$ is a semi-norm.

Now substituting (4.23)-(4.26) into (4.22) yields that

$$\begin{aligned}
& \|\mathbf{b}^k - \mathbf{b}^*\|^2 - \|\mathbf{b}^{k+1} - \mathbf{b}^*\|^2 - \|\mathbf{b}^{k+1} - \mathbf{b}^k\|^2 + \mu\rho \|\mathbf{b}^{k+1} - \mathbf{b}^k\|_G^2 \\
&+ 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^*, \mathbf{b}^{k+1}) - 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^*, \mathbf{b}^k) + 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^{k+1}, \mathbf{b}^k) \\
&+ \frac{\mu}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 - \frac{\mu}{\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \frac{\mu}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 \geq 0
\end{aligned} \tag{4.27}$$

Let $V_k = \|\mathbf{b}^k - \mathbf{b}^*\|^2 + \frac{\mu}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 - 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^*, \mathbf{b}^k)$. As $\mu < 1/(\rho+L)$, $\|\mathbf{b}^k - \mathbf{b}^*\|^2 \geq \frac{2}{L}\mathcal{D}_\mathcal{E}(\mathbf{b}^*, \mathbf{b}^k) > \frac{2}{L+\rho}\mathcal{D}_\mathcal{E}(\mathbf{b}^*, \mathbf{b}^k) > 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^*, \mathbf{b}^k)$ such that V_k is positive.

Then we have

$$\begin{aligned}
V_k - V_{k+1} &\geq \frac{\mu}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 + \|\mathbf{b}^{k+1} - \mathbf{b}^k\|^2 - \mu\rho \|\mathbf{b}^{k+1} - \mathbf{b}^k\|_G^2 - 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^{k+1}, \mathbf{b}^k) \\
&= \frac{\mu}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 + \|\mathbf{b}^{k+1} - \mathbf{b}^k\|_{I-\mu\rho G}^2 - 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^{k+1}, \mathbf{b}^k)
\end{aligned} \tag{4.28}$$

As $I-\mu\rho G$ is semidefinite positive and $\|\mathbf{b}^{k+1} - \mathbf{b}^k\|_{I-\mu\rho G}^2 \geq (1-\mu\rho) \|\mathbf{b}^{k+1} - \mathbf{b}^k\|^2 > \mu L \|\mathbf{b}^{k+1} - \mathbf{b}^k\|^2 \geq 2\mu\mathcal{D}_\mathcal{E}(\mathbf{b}^{k+1}, \mathbf{b}^k)$, the above inequality can be rewritten as

$$V_k - V_{k+1} \geq \frac{\mu}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 + (1 - \mu\rho - \mu L) \|\mathbf{b}^{k+1} - \mathbf{b}^k\|^2 \tag{4.29}$$

This shows that V_k is non-increasing. Because $V_k \leq V_0$, it follows that $\mathbf{b}^k$ and $\boldsymbol{\lambda}^k$ are bounded. Summing (4.29) over k from 0 to infinity leads to

$$\sum_{k=0}^{\infty} \frac{\mu}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 + (1 - \mu\rho - \mu L) \|\mathbf{b}^{k+1} - \mathbf{b}^k\|^2 \leq V_0$$

which implies $\lim_{k \rightarrow \infty} \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1} = 0$ and $\lim_{k \rightarrow \infty} \mathbf{b}^{k+1} - \mathbf{b}^k = 0$. Through (4.12b), it is easy to obtain $\lim_{k \rightarrow \infty} \mathbf{z}^{k+1} - \mathbf{z}^k = G(\lim_{k \rightarrow \infty} \mathbf{b}^{k+1} - \mathbf{b}^k) + G/\rho(\lim_{k \rightarrow \infty} \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) = 0$.

Thus with (4.27) and the fact that $\{V_k\}_{k \geq 0}$ converges, we conclude that when $\mu < 1/(\rho + L)$, the sequence $\{(\mathbf{b}^k, \mathbf{z}^k, \boldsymbol{\lambda}^k)\}$ uniquely converges to the saddle point $(\mathbf{b}^*, \mathbf{z}^*, \boldsymbol{\lambda}^*)$ of the augmented Lagrangian $\mathcal{L}_\rho(\mathbf{b}, \mathbf{z}, \boldsymbol{\lambda})$. $\square$

4.4 Determination of Key Nodes

In this section, we discuss how to select key nodes and construct the solution of the original problem (4.2) based on the solution of IADMM. By taking the degree information of nodes, D-IADMM is proposed and key nodes are identified. To further improve the performance of D-IADMM, we employ a greedy algorithm and propose LD-IADMM.

4.4.1 Key node selection based on degree distribution

Note that $\mathbf{b}^{I*}$ is the solution of the problem (4.3) where $0 \leq \mathbf{b}_i^{I*} \leq 1$. This implies that node i is connected with external controller with input weight $\mathbf{b}_i^{I*}$, $i = 1, \dots, N$. Summing all these input weights leads to $\mathbf{b}^T \mathbf{1} = M$. Getting back to the original problem (4.2), we hope to find M most important nodes to construct the key node set with obtained weight information $\mathbf{b}_i^{I*}$. On the other hand, in [96], it is stated that key nodes are related to networks' degree distribution trend for directed networks. And in [18], it is pointed out that each stem of the directed network must be driven by an independent external control source to ensure the controllability of the networked system [1]. Therefore, motivated by these results, we propose another algorithm, named degree-based inexact alternating direction method of multipliers (D-IADMM) to identify key node set based on the solution of IADMM by exploiting degree information.

Denote the in-degree and out-degree of node i as k_i^{in} and k_i^{out} respectively. Similar to [1], sources are defined as nodes with no edges pointing to them, i.e. $k_i^{\text{in}} = 0$. Sinks are nodes with edges only pointing towards them. That is, $k_i^{\text{out}} = 0$ and $k_i^{\text{in}} \neq 0$. On one hand, based on the results in [1], we know that the sources are definitely connected with external inputs. On the other hand, in [97, 98], it is

concluded that the longest control chain (LLC) dominates the control cost of the network and the extra control inputs should be placed near the middle of the chain so as to minimize the length of the LCC. Therefore, the sinks are almost unlikely to be key nodes except the case where two sinks have the same parent node. Such an exception is illustrated by an example in Fig. 4.1. Note that in-degree and out-degree of Node 2 / Node 3 are 1 and 0 respectively, and they have a common parent node, i.e., Node 1. However, according to [18], there is still a possibility for Node 2 and/or Node 3 to be a key node. We call the pattern illustrated in Fig. 4.1 as Ω -pattern.

After pre-determination of sources as key nodes and pre-elimination of some sinks from key node set based on above descriptions, now we determine the remaining key nodes. Suppose that the number of pre-determined key nodes is N_s and the number of eliminated nodes from key node set is N_e . Then we calculate $b_i^{I^*}/k_i^{\text{in}}$ for the remaining $N - N_s - N_e$ nodes. Finally, nodes with $M - N_s$ largest values are chosen as key nodes [62]. Let $\mathcal{K}^*$ be the key node set and $\mathbf{b}^*$ be the solution obtained by D-IADMM. Obviously, if node i is a key node, $b_i^* = 1$ and $i \in \mathcal{K}^*$. Otherwise, $b_i^* = 0$ and $i \notin \mathcal{K}^*$. The above rounding procedure is detailedly presented in Algorithm 4.2.

FIGURE 4.1: Ω -pattern.

Combining Algorithms 4.1 and 4.2, we term the whole process of determining key nodes as degree-based inexact alternating direction method of multipliers (D-IADMM).

4.4.2 Local optimization

Note that the key node set identified by D-IADMM yields a control cost no more than a near-optimal solution. Therefore, we hope to further decrease the control

Algorithm 4.2 key node selection in directed networks based on nodes' degree distribution (D-IADMM)

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

Output: $\mathbf{b}^*$, key node set $\mathcal{K}^*$

```

1: Initialization: Initialize  $\mathcal{K} = \emptyset$ ,  $\tilde{\mathcal{K}} = \emptyset$ ,  $\mathbf{b} = \mathbf{0}$ 
2: Calculation: Calculate the in-degree  $k_i^{\text{in}}$  and out-degree  $k_i^{\text{out}}$  of node  $i$  respectively ( $i = 1, 2, \dots, N$ ).
3: for  $i$  from 1 to  $N$  do
4:   if  $k_i^{\text{in}} = 0$  then
5:      $\mathcal{K} \leftarrow \mathcal{K} \cup \{i\}$ 
6:   else if  $k_i^{\text{out}} = 0$  and  $i \notin \Omega\text{-pattern}$  then
7:      $\tilde{\mathcal{K}} \leftarrow \tilde{\mathcal{K}} \cup \{i\}$ 
8:   end if
9: end for
10: while  $|\mathcal{K}| < M$  do
11:    $i \in \arg \max_{i \in \mathcal{V} \setminus \{\mathcal{K} \cup \tilde{\mathcal{K}}\}} b_i^{I*} / k_i^{\text{in}}$ 
12:    $b_i = 1$ 
13:    $\mathcal{K} \leftarrow \mathcal{K} \cup \{i\}$ 
14: end while
15: return  $\mathbf{b}^* = \mathbf{b}$  and  $\mathcal{K}^* = \mathcal{K}$ .
    
```

cost. In this subsection, we employ a greedy algorithm, i.e. local optimization (LO)[99] based on the result of D-IADMM.

Specifically, for a key node set $\mathcal{K}^*$ obtained from Algorithm 4.2, we swap one node with $k_i^{\text{in}} \neq 0$ in $\mathcal{K}^*$ with the node in candidate key node set $\mathcal{V} \setminus \mathcal{K}^*$ which can help maximize the reduction of the control cost. This technique is similar to Fedorov's exchange algorithm [100] and Wynn's algorithm [101]. For large scale networks, as the total number of swaps may be large, the maximum number of swaps will be restricted as a linear function of network size [70]. Denote the key node set obtained by LD-IADMM as $\mathcal{K}'$ and the solution as $\mathbf{b}'$. Similar to D-IADMM, if node i is a key node, $b'_i = 1$ and $i \in \mathcal{K}'$. Otherwise, $b'_i = 0$ and $i \notin \mathcal{K}'$. The detailed procedure is given in Algorithm 4.3.

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

Algorithm 4.3 Local optimization with the solution obtained by LD-IADMM

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

Output: $\mathbf{b}'$, new key node set $\mathcal{K}'$ obtained by LD-IADMM,

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

4.5 Simulation Results

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

Note that effectiveness of proposed algorithms can be validated by the control cost associated with the selected key nodes. The lower the control cost, the better the algorithm performs. We first carry out simulations on ER and SF networks where SF networks are generated by Barabási-Albert (BA) model. Fig. 4.2 presents the minimum control cost with different M obtained by PGME, IADMM and LD-IADMM respectively. In Fig. 4.3, the number of external control sources is fixed as $M = 20$ while the mean degree of ER and SF networks varies. Based on results in Fig. 4.2 and Fig. 4.3, it is worthy of highlighting the following observations. Firstly, it is noted that the performance of D-IADMM is slightly better than PGME. This is because D-IADMM has taken the degree information into account. Secondly, it is observed that the technique of local optimization can greatly improve the performance of D-IADMM. Thirdly, since the simulations in Fig. 4.2 are carried on ER and SF networks with the same mean degree $\langle k \rangle = \langle k \rangle^{\text{in}} = \langle k \rangle^{\text{out}} = 7.5$ and the same size of network $N = 100$, it is obvious that the minimum control cost

FIGURE 4.2: Control cost versus number of key nodes for directed ER and SF networks with $N = 100$, $\langle k \rangle = 7.5$. (a) ER network (b) SF network

FIGURE 4.3: Comparison between proposed algorithms for directed networks with different mean degree. (a) Directed ER network with $N = 50$, $M = 20$. (b) Directed SF network with $N = 50$, $M = 20$

of ER network is lower than that of SF network. The results in Fig. 4.3 further demonstrate this finding. Finally, when the number of external control sources is fixed, the denser the network, the lower the control cost, as shown in Fig. 4.3.

Note that in Algorithm 4.2, we mainly take the information of in-degree of nodes into account. This is motivated by the process of maximum matching in directed networks [18]. Now we explore the in-degree distribution for key nodes obtained by LD-IADMM, based on the results of ER and SF networks with $N = 50$ and $M = 15$ presented in Fig. 4.4 which are obtained by averaging over 10 realizations. It is seen that the fraction of key nodes is significantly high among nodes with low

FIGURE 4.4: **In-degree distribution of key nodes in directed ER and SF networks with $N = 50$, $M = 15$, $\langle k \rangle = 5$.** (a) ER network (b) SF network

TABLE 4.1: The performance comparison in ER and SF networks

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

in-degree and medium in-degree. Clearly, key nodes tend to avoid nodes with high in-degree.

To further validate the effectiveness of the algorithms, larger ER and SF networks with $N = 500$ are considered. The comparison between PGME, D-IADMM, LD-IADMM and RAM is shown in Table 4.1. It is obvious that performances of D-IADMM and LD-IADMM are much better than PGME and RAM.

We also apply our proposed algorithm on real-world networks, including electronic circuit network, food web and social networks. Four algorithms, PGME, D-IADMM, LD-IADMM and RAM are compared. The obtained results are summarized in Table 4.2. It can be noted that the control cost obtain by LD-IADMM is 1-6 orders of magnitude lower than the cost obtained by the other three algorithms.

TABLE 4.2: The performance comparison between proposed algorithms in real-world networks

Data set	Networks	N	M	PGME	D-IADMM	LD-IADMM	RAM
Circuit	circuit-s208	122	70	7.48E03	6.01E03	5.28E02	9.16E05
	circuit-s420	252	100	2.62E06	1.13E06	8.72E03	2.15E08
Food web	Rhode	25	12	2.52E03	2.95E03	8.01E01	2.83E05
	Maspalomas	30	10	3.83E05	2.50E05	1.28E02	1.18E08
Social	cons-fre-rev	46	12	1.23E09	3.29E08	1.51E05	8.20E09
	phys-friend-rev	228	90	2.02E07	4.67E06	1.30E03	1.07E09

4.6 Conclusion

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

Chapter 5

Implicit Linear Quadratic Regulation for Directed Complex Networks

In Chapter 3 and Chapter 4, only input energy, i.e. $\int_0^{t_f} \|\mathbf{u}(t)\|^2 dt$ is minimized when we design the input matrix B . Similar to traditional linear quadratic regulator (LQR) problem in control theory, in this chapter, we add another quadratic term that penalizes the state deviation, i.e. $\int_0^{t_f} \|\mathbf{x}(t)\|^2 dt$, to the objective function. This provides us a flexibility to control the networks under different control objectives.

Although linear quadratic problem which minimizes a quadratic cost function for controlling linear time invariant (LTI) directed networks has been well investigated, it becomes inapplicable when certain system parameters are free to choose and thus are considered as variables to be determined for the optimization problem. In this chapter, we consider such an issue that input matrix B is selectable instead of pre-given. An implicit linear quadratic regulator (ImLQR) with B being considered as a variable matrix in the open-loop controller design is formulated. To address such an issue, orthonormal-constraint-based projected gradient method (OPGM) is proposed to determine B iteratively. Convergence of the proposed algorithm is established. Extensive simulation results show the effectiveness of the proposed algorithm. Through simulation studies, we also investigate what kind of nodes is most important for minimizing an average control cost in directed stems/circles and

small networks. The presented results in this chapter bring meaningful physical insights into the optimal control of LTI directed networks.

5.1 Introduction

Optimal control is to determine a control strategy that drives the states of a dynamic system to predefined states with minimum cost. Since linear time invariant dynamics on directed networks has been widely adopted to model various real-life systems [1, 18, 102], such a problem is generally formulated as a linear quadratic (LQ) problem subject to a linear first-order dynamic constraint in (2.1). As mentioned in Chapter 1, the LQ problem has been well investigated when A and B are pre-given and fixed. However, it is generally difficult, if not impossible, to identify the input matrix B if the objective is instead to properly control the network with selectable input matrix. Here “selectable” means that each element of the input matrix, denoted as b_{im} , is a variable to be determined instead of a given value. That is, one needs to determine whether a node is connected to an external control source and what the exact connection weight is if connected, through the optimal way of minimizing an appropriate cost function.

Thus when the input matrix B becomes selectable, there are two essential problems in controlling an LTI directed network: (i) to find the minimum number of external control sources M connected to external control inputs for ensuring the controllability of the system [18, 103]; and (ii) for a given number of control sources which ensures the system’s controllability, to further determine the input matrix B together with a solution of driving the system state to any predefined state with minimum cost. The former issue is known as the network controllability problem and the latter one is termed as minimum-cost control problem with selectable inputs.

Existing studies [18, 103, 104] have mainly focus on the former issue and made important advances in identifying minimum number of external control sources, through which we can ensure the controllability of the system as one node can alter the states of other nodes through the interactions between nodes. Such a control process plays an important role in the regulation of protein expression

[105], the coordination of moving robots [106], and the inhibition of undesirable social contagions [107].

However, for many real-life applications, an even more important problem is how to utilize M external control sources including the determination of their weights for minimizing the cost required in controlling the networks. This significant problem has many applications in various real-life networks in physics [18] [103], biology [108], social networks [81, 109], brain networks [11] and engineering [106]. For example, in the coordination of moving robots [106], how to select the optimal leader nodes which are regarded as nodes connected to the external control sources is quite essential to achieve the coordination objective most efficiently. In the gene regulatory networks [108], how external signals can be allocated to the nodes so that the levels (e.g., protein concentration) of all nodes can be adjusted in a finite time to some values most efficiently has been identified as a key technique.

Recently, researchers start to consider the second issue in the field of complex networks. Some interesting heuristic results have been available, including [37, 38, 61]. On the other hand, this problem is related to the leader selection problem in undirected networks [70, 71, 110], and the constrained input selection problems in directed networks [20, 39, 61, 111], as they are often approached as minimization of the control cost [112]. The external control sources could be regarded as sensors/actuators in [89], or communication/control towers in [113]. In [114], the problem is formulated as an optimization problem with matrix-trace-based cost function under trace boundary constraint or orthonormal boundary constraint. But in all of these works, the quadratic term of system state is not considered in the cost function while the deviation of system state is undesired in many practical applications.

Departing from works [37, 114] and analogously from [34], in this chapter, we investigate the minimum cost control problem with selectable inputs by adopting the cost function summed over both quadratic terms of system input and system state with a weighting factor α . Including such a quadratic term in the cost function makes the problem become challenging due to the difficulty of obtaining the derivative of the cost function with respect to B . To solve this problem, we first manage to obtain the derivative of the quadratic term of system state by I-Chain rule introduced in Lemma 2.2 for matrix function differentiation. Then an algorithm named orthonormal-condition-based projected gradient method (OPGM) is

proposed to based on the obtained gradient information. To guarantee the convergence property of OPGM, a key technique is to introduce the positive parameter λ_k during the iteration process of OPGM. Thus, with the converged solution, the input matrix is obtained, and the optimal controller is designed. We name the optimal controller as implicit linear quadratic regulator (ImLQR) since it is implicitly dependent on input matrix found by OPGM. Extensive experiments are conducted so that the effectiveness of ImLQR including the effects of the weighting factor α is verified. Our findings help us to better understand how the network should be controlled when B becomes selectable.

5.2 Problem Formulation

Different from existing works, the problem of an ImLQR is to determine an input matrix $B \in \mathbb{R}^{N \times M}$ and also design control input vector $\mathbf{u} \in \mathbb{R}^{M \times 1}$ such that the quadratic cost of controlling a directed network $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$, $\mathbf{x}(0) = \mathbf{x}_0$ is minimum. Since the pairs $B, \mathbf{u}(t)$ and $kB, \frac{1}{k}\mathbf{u}(t)$ represent the same inputs and outputs for a nonzero k , it is natural to fix the norm of matrix B . There are different ways of doing this and each way corresponds to a distinctive boundary condition.

In this chapter, we consider the orthonormal boundary condition, that is, B is restricted to be an orthonormal matrix. This is because in real applications, if the columns of the input matrix B are chosen as dissimilar as possible, then the redundancy of the input signals can be reduced as far as possible. We surmise that an ideal case could be the orthonormal boundary constraint, i.e., $B^T B = I_M$ where I_M is an identity matrix with a dimension M , which means all the columns of B are orthogonal to each other. For example, when considering a case that different external sources are restricted to be only connected to different node sets, the constraint $B^T B = I_M$ can be easily satisfied. In this case, the minimum cost control problem with selectable inputs is formulated as a constrained optimization problem, where the constraint is to drive the system from an initial state $\mathbf{x}_0$ to the

origin during time interval $[0, t_f]$,

$$\begin{aligned}
\min_{\mathbf{u}, B} \quad & \alpha \int_0^{t_f} \|\mathbf{u}(t)\|^2 dt + (1 - \alpha) \int_0^{t_f} \|\mathbf{x}(t)\|^2 dt \\
s.t. \quad & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\
& B^T B = I_M, \quad (A, B) \text{ is controllable} \\
& \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_{t_f} = \mathbf{0}
\end{aligned} \tag{5.1}$$

where $\mathbf{x}(t) = [x_1^T(t), \dots, x_M^T(t)]^T$ and $\mathbf{u}(t) = [u_1^T(t), \dots, u_M^T(t)]^T$, and $\mathbf{0}$ (vector) denotes the origin. Here the cost function contains two parts with a weight parameter $0 < \alpha \leq 1$. Note that both $\mathbf{u}(t)$ and B are decision variables to be determined, where $\mathbf{u}(t)$ specifies the values of controller inputs at each operating time and B determines the nodes to which the controller inputs are connected and the weights of the connections.

When B is selectable, both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ become functions of B , which are denoted as $\mathbf{x}(t, B)$ and $\mathbf{u}(t, B)$, respectively. We consider the following two optimization models for two types of control cost, i.e., the position-dependent control cost and the average control cost, respectively.

Optimization model 5.1. The position-dependent control cost refers to the cost of driving an LTI system $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$ from a pre-given initial state $\mathbf{x}_0$ to the origin over $[0, t_f]$. The optimization model is formulated as

$$\begin{aligned}
\min \quad & E(t_f, B) = \alpha \int_0^{t_f} \mathbf{u}^T(t, B)\mathbf{u}(t, B)dt + (1 - \alpha) \int_0^{t_f} \mathbf{x}^T(t, B)\mathbf{x}(t, B)dt \\
s.t. \quad & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\
& B^T B = I_M \\
& \mathbf{x}(0) = \mathbf{x}_0, \mathbf{x}(t_f) = \mathbf{x}_{t_f} = \mathbf{0},
\end{aligned} \tag{5.2}$$

where (A, B) are matrices such that the network is controllable (Theorem 5.1). $E(t_f, B)$ is position-dependent cost function, which is dependent on t_f and B .

Optimization model 5.2. When it comes to the average control cost, we sum all realizations of position-dependent control cost for all possible initial states $\mathbf{x}_0$. Then, the optimization model becomes

$$\begin{aligned}
\min \quad & \mathcal{E}(t_f, B) = \alpha \int_0^{t_f} \mathbb{E}\{\mathbf{u}^T(t, B)\mathbf{u}(t, B)\}dt + (1 - \alpha) \int_0^{t_f} \mathbb{E}\{\mathbf{x}^T(t, B)\mathbf{x}(t, B)\}dt \\
s.t. \quad & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\
& B^T B = I_M \\
& \mathbf{x}(t_f) = \mathbf{x}_{t_f} = \mathbf{0}
\end{aligned} \tag{5.3}$$

where $\mathbb{E}\{\cdot\}$ takes the expectation of the argument over all realizations of the random initial states.

For closed-loop optimal control of LTI systems with state $\mathbf{x}(t)$ involved, we may need to solve a Riccati equation when an unknown B is involved. It is known that the analytic solution of Riccati equation is very difficult, if not impossible. Thus for the above two optimization models, we only consider open-loop optimal control of LTI systems with input matrix B being a matrix variable to be determined. Extending this work to closed-loop case will be a promising yet challenging future work.

In [50, 115], for all $0 < \alpha \leq 1$ as long as $\mathbf{u}(t, B)$ is designed as

$$\mathbf{u}(t, B) = -B^T e^{A^T(t_f-t)} W_B^{-1} e^{At_f} \mathbf{x}_0 \tag{5.4}$$

where W_B is Gramian matrix. It can be guaranteed that $\mathbf{x}(t_f) = \mathbf{0}$ for any $\mathbf{x}_0$ and any B satisfying that (A, B) is controllable. This is because

$$\begin{aligned}
\mathbf{x}(t_f, B) &= e^{At_f} \mathbf{x}_0 + \int_0^{t_f} e^{A(t_f-\nu)} B \mathbf{u}(\nu) d\nu \\
&= e^{At_f} \mathbf{x}_0 - \int_0^{t_f} e^{A(t_f-\nu)} B B^T e^{A^T(t_f-\nu)} W_B^{-1} e^{At_f} \mathbf{x}_0 d\nu \\
&= e^{At_f} \mathbf{x}_0 - \left[\int_0^{t_f} e^{A\tilde{t}} B B^T e^{A^T \tilde{t}} d\tilde{t} \right] W_B^{-1} e^{At_f} \mathbf{x}_0 \\
&= e^{At_f} \mathbf{x}_0 - W_B W_B^{-1} e^{At_f} \mathbf{x}_0 \\
&= \mathbf{0}
\end{aligned} \tag{5.5}$$

Thus, as long as (A, B) is controllable, we can always design $\mathbf{u}(t)$ to drive the system from an arbitrary initial state to the origin in a fixed time interval $[0, t_f]$. So for a

given t_f , the control cost functions $E(t_f, B)$ and $\mathcal{E}(t_f, B)$ are just dependent on the selection of B . In a word, our objective is to find an optimal input matrix B subject to orthonormal boundary constraint so as to minimize $E(B)$ or $\mathcal{E}(B)$. Before presenting orthonormal-condition-based projected gradient method (OPGM) that searches for such a B iteratively, we need to consider the derivative of $E(B)$ or $\mathcal{E}(B)$ with respect to B , i.e., to obtain $\frac{\partial \int_0^{t_f} \mathbf{u}^T(t, B) \mathbf{u}(t, B) dt}{\partial B}$ and $\frac{\partial \int_0^{t_f} \mathbf{x}^T(t, B) \mathbf{x}(t, B) dt}{\partial B}$, respectively.

From (5.4)-(5.5), it is seen that the system input $\mathbf{u}(t, B)$ depends on B , the state transition matrix e^{At} and the inverse of Gramian matrix W_B^{-1} which is again a function of B , and that the system state $\mathbf{x}(t, B)$ is a convolution of the state transition matrix and the system input to be designed. Therefore, obtaining the derivatives of the quadratic terms involving system input and system state with respect to B is a very complicated process. By I-Chain rule in Lemma 2.2 for matrix function differentiation, we manage to solve the problem, which enables us to propose the orthonormal-condition-based projected gradient method (OPGM) to be presented in the next section.

Lemma 5.1. *The derivative of $\int_0^{t_f} \mathbf{u}^T(t, B) \mathbf{u}(t, B) dt$ with respect to B is given by*

$$\frac{\partial \int_0^{t_f} \mathbf{u}^T(t, B) \mathbf{u}(t, B) dt}{\partial B} = -F_u(B) \cdot B \quad (5.6)$$

where $F_u(B) = 2 \int_0^{t_f} e^{A^T t} W_B^{-T} e^{At} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t} W_B^{-T} dt$.

Proof. Please see Appendix B. □

An important contribution of this work is how to obtain $\frac{\partial \int_0^{t_f} \mathbf{x}^T(t, B) \mathbf{x}(t, B) dt}{\partial B}$, which generally is more sophisticated than obtaining $\frac{\partial \int_0^{t_f} \mathbf{u}^T(t, B) \mathbf{u}(t, B) dt}{\partial B}$. The result is given by the following Lemma.

Lemma 5.2. *The derivative of $\int_0^{t_f} \mathbf{x}^T(t, B) \mathbf{x}(t, B) dt$ with respect to B is given by*

$$\frac{\partial \int_0^{t_f} \mathbf{x}^T(t, B) \mathbf{x}(t, B) dt}{\partial B} = -F_x(B) \cdot B \quad (5.7)$$

where

$$\begin{aligned}
F_x(B) &= 2 \int_0^{t_f} \int_0^t e^{A^T(t-\tau)} \mathbf{x}(t, B) \mathbf{x}_0^T e^{A^T t_f} W_B^{-T} e^{A(t_f-\tau)} d\tau dt \\
&+ 2 \int_0^{t_f} \int_0^t e^{A^T(t_f-\tau)} W_B^{-1} e^{A t_f} \mathbf{x}_0 \mathbf{x}^T(t, B) e^{A(t-\tau)} d\tau dt \\
&- 2 \int_0^{t_f} e^{A^T \sigma} W_B^{-T} \int_0^{t_f} \int_0^t e^{A(t_f-\tau)} B B^T e^{A^T(t-\tau)} \mathbf{x}(t, B) \mathbf{x}_0^T e^{A^T t_f} d\tau dt W_B^{-T} e^{A \sigma} d\sigma \\
&- 2 \int_0^{t_f} e^{A^T \sigma} W_B^{-1} \int_0^{t_f} \int_0^t e^{A t_f} \mathbf{x}_0 \mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)} d\tau dt W_B^{-1} e^{A \sigma} d\sigma
\end{aligned} \tag{5.8}$$

with

$$\begin{aligned}
\mathbf{x}_0 \mathbf{x}^T(t, B) &= \mathbf{x}_0 \mathbf{x}_0^T \left[e^{A^T t} - e^{A^T t_f} W_B^{-T} \int_0^t e^{A(t_f-\nu)} B B^T e^{A^T(t-\nu)} d\nu \right] \\
\mathbf{x}(t, B) \mathbf{x}_0^T &= \left[e^{A t} - \int_0^t e^{A(t-\nu)} B B^T e^{A^T(t_f-\nu)} W_B^{-1} e^{A t_f} d\nu \right] \mathbf{x}_0 \mathbf{x}_0^T
\end{aligned} \tag{5.9}$$

□

Proof. Please see Appendix B. □

Lemma 5.3. The derivative of $E(B)$ with respect to B is given by

$$\frac{\partial E(B)}{\partial B} = -F(B) \cdot B$$

where $F(B) = [\alpha F_u(B) + (1-\alpha) F_x(B)]$ with $F_u(B)$ and $F_x(B)$ obtained in Lemma 5.1 and Lemma 5.2, respectively.

Proof. This can be easily obtained from Lemmas 5.1-5.2. □

Remark 5.1. The derivatives in (5.6)-(5.9) are obtained when the position-dependent control cost is considered for a pre-given $\mathbf{x}_0$. When applying the expectation function $\mathbb{E}\{\cdot\}$ to investigate the average control cost, it is assumed that element of initial state $\mathbf{x}_0 = [x_1(0), \dots, x_N(0)]^T$ is an identical independent distributed (i.i.d) variable with zero mean and variance 1. Under this assumption, $x_i(0)$ and $x_j(0)$ are

two independent random variables with zero mean and unit variance. Therefore, we have $\mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] = I_N$, the same as Eq. (3.1) in Chapter 3.

Thus, for the average control cost, the derivative of $\mathcal{E}(B)$ with respect to B can also be obtained by replacing $\mathbf{x}_0 \mathbf{x}_0^T$ in Lemmas 5.1, 5.2 with $\mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T]$, which is an identity matrix with dimension N .

Define a Stiefel manifold [116] $\mathbb{O}^{N \times M} := \{B \in \mathbb{R}^{N \times M} : B^T B = I_M\}$, which is considered as an embedded submanifold of the Euclidean space $\mathbb{R}^{N \times M}$, and define a space spanned by B as $\text{Span}\{B\} := \{X : X = BY, B \in \mathbb{O}^{N \times M}, Y \in \mathbb{R}^{M \times M}\}$. We have the following Lemma.

Lemma 5.4. *The operator given by*

$$\mathcal{T}_B = (I_N - BB^T) \quad (5.10)$$

is a projection operator which projects $Z \in \mathbb{R}^{N \times M}$ onto a space perpendicular to the space $\text{Span}\{B\}$, i.e., $\mathcal{T}_B \mathbb{O}^{N \times M} := \{X \in \mathbb{R}^{N \times M} : X = (I_N - BB^T)Z, \forall B \in \mathbb{O}^{N \times M}, Z \in \mathbb{R}^{N \times M}\}$ at any $Z \in \mathbb{R}^{N \times M}$ is perpendicular to $\text{Span}\{B\}$.

Proof. Note that

$$\begin{aligned} \mathcal{T}_B^T \mathcal{T}_B &= (I_N - BB^T) \cdot (I_N - BB^T) \\ &= I_N - 2BB^T + BB^T BB^T \\ &= I_N - BB^T \\ &= \mathcal{T}_B \end{aligned} \quad (5.11)$$

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

$$\begin{aligned} \text{tr}(X^T \mathcal{T}_B Z) &= \text{tr}(Y^T B^T (I_N - BB^T) Z) \\ &= \text{tr}(Y^T (B^T - B^T BB^T) Z) \\ &= 0 \end{aligned} \quad (5.12)$$

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

Step 2) At iteration k , calculate the gradients $\nabla E(B_k)$ based on Lemmas 5.1-5.2.

Then obtain $\hat{B}_{k+1}$ by the following gradient descent iteration

$$\begin{aligned}
 \hat{B}_{k+1} &= B_k + \Delta \hat{B}_k \\
 &= B_k + \eta \cdot \Delta B_k \\
 &= B_k - \eta \cdot (I_N - B_k B_k^T) \cdot \widetilde{\nabla E(B_k)} \\
 &= B_k - \eta \cdot \mathcal{T}_{B_k} \widetilde{\nabla E(B_k)}
 \end{aligned} \tag{5.14}$$

where $\Delta \hat{B}_k = \eta \cdot \Delta B_k$, $\Delta B_k = -\mathcal{T}_{B_k} \cdot \widetilde{\nabla E(B_k)}$,

Step 3) Normalize $\hat{B}_{k+1}$ by

$$B_{k+1} = \sqrt{\frac{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})}} \cdot \hat{B}_{k+1} \tag{5.15}$$

Step 4) Update k to $k + 1$ until certain criterion is satisfied, i.e. max iteration number.

Remark 5.2. The idea of proposing update rule in (5.15) is explained as follows. We define a norm function as

$$\begin{aligned}
 \mathcal{N}(B) &= \text{tr} \left((B^T B - I_M)^T (B^T B - I_M) \right) \\
 &= \text{tr}(B^T B B^T B) - 2 \text{tr}(B^T B) + M
 \end{aligned} \tag{5.16}$$

Note that $\forall B \in \mathbb{R}^{N \times M}$, $B^T B = I_M$ is equivalent to $\mathcal{N}(B) = 0$. Let $B_{k+1} = \rho_k \cdot \hat{B}_{k+1}$. It is seen that there is no exact solution ρ_k satisfying that $\mathcal{N}(B_{k+1}) = 0$ since it is generally impossible to get a matrix with orthonormal columns by just multiplying a matrix with non-orthonormal columns with a scalar. In the iteration, we minimize $\mathcal{N}(B_{k+1})$ by solving $\frac{\partial \mathcal{N}(\rho_k \cdot \hat{B}_{k+1})}{\partial \rho_k} = 0$, which gives

$$4\rho_k^3 \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1}) - 4\rho_k \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1}) = 0 \tag{5.17}$$

Then we can obtain $\rho_k = \sqrt{\frac{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})}}$.

Assumption 5.1. The initial $B_0 \in \mathbb{R}^{N \times M}$ satisfies that $\mathcal{N}(B_0) < 1$ where B_0 is a full rank matrix such that that (A, B_0) is controllable.

Remark 5.3. Suppose that N_D is the minimum number of external inputs to guarantee the structural controllability of directed networks, which can be determined by maximum matching in [18]. Since each element of the initial input matrix B_0 can be a random value, it is easy to meet Assumption 5.1, which requires that the rank of matrix B_0 is not less than N_D , i.e., $\text{rank}(B_0) = M \geq N_D$. Also $\mathcal{N}(B_0) < 1$ implies that B_0 is a full rank matrix, more detailed explanation is illustrated in Theorem 5.1.

Remark 5.4. As stated in the Introduction, one key technique is the use of λ_k in the iteration process of OPGM for assuring its convergence. Introducing the parameter λ_k is similar to the idea of Levenberg-Marquardt stabilization, also known as the damped least-squares (DLS)[117, 118] 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 $k \rightarrow \infty$, with B^* having orthonormal columns, provided that the step size η is sufficiently small. Once a solution of input matrix B is obtained by OPGM, the input can be easily designed by (5.4).

Lemma 5.5. For the norm function $\mathcal{N}(B)$ where the $B \in \mathbb{R}^{N \times M}$ is of full column rank, $\nabla \mathcal{N}(B_k) = 0$ implies that $\mathcal{N}(B_k) = 0$, i.e., $B_k^T B_k = I_M$.

Proof. Note that

$$\nabla \mathcal{N}(B_k) = \frac{\partial \mathcal{N}(B_k)}{\partial B_k} = 4(B_k B_k^T - I_N)B_k. \quad (5.18)$$

Assume that $\mathcal{N}(B_k) \neq 0$ when $\nabla \mathcal{N}(B_k) = 0$. As $\nabla \mathcal{N}(B_k) = 0$ at $B = B_k$ implies that $\text{tr}([\nabla \mathcal{N}(B_k)]^T [\nabla \mathcal{N}(B_k)]) = 0$, we have

$$\begin{aligned} & \text{tr} \left([(B_k B_k^T - I_N)B_k]^T [(B_k B_k^T - I_N)B_k] \right) \\ &= \text{tr}(B_k^T B_k B_k^T B_k B_k^T B_k) - 2 \text{tr}(B_k^T B_k B_k^T B_k) + \text{tr}(B_k^T B_k) \\ &= 0 \end{aligned}$$

As $B_k^T B_k$ is symmetrical and nonnegative definite matrix, there exists an orthonormal matrix $P \in \mathbb{R}^{M \times M}$ ($P^T P = P P^T = I$) such that

$$B_k^T B_k = P^T \Lambda P \quad (5.19)$$

where Λ contains all the eigenvalues of $B_k^T B_k$. We then have

$$\begin{aligned} & \text{tr} \left([(B_k B_k^T - I_N) B_k]^T [(B_k B_k^T - I_N) B_k] \right) \\ &= \text{tr} (P^T \Lambda (\Lambda^2 - 2\Lambda + I) P) \\ &= \sum_i \Lambda_i (\Lambda_i - 1)^2 \end{aligned} \quad (5.20)$$

which implies that $\Lambda_i = 1$ for all i as B_k is of full column rank. This implies that all the columns of B is orthonormal and we have $\mathcal{N}(B_k) = 0$ when $\nabla \mathcal{N}(B_k) = 0$. $\square$

5.4 Convergence Proof

An immediate problem is how to establish the convergence of OPGM. This is a tricky problem as $I - B_k B_k^T$ is not a strict projection operator unless when orthonormal boundary condition $B_k^T B_k = I_M$ can be guaranteed. However, as discussed in Remark 5.3, it is generally impossible to get a matrix with orthonormal columns by just multiplying a matrix with non-orthonormal columns by a scalar. Here we prove the convergence of OPGM with the help of the following lemmas.

Lemma 5.6. *Given a finite $\bar{x} \geq 0$, for every $\varepsilon > 0$, there exists $\eta > 0$, such that $|(1 + \eta \cdot \bar{x})^{-\frac{1}{2}} - (1 - \frac{1}{2}\eta \cdot \bar{x})| < \varepsilon$.*

Proof. The result is obtained by just applying Taylor expansion and thus the details are omitted. $\square$

Lemma 5.7. $\forall B \in \mathbb{R}^{N \times M}$ and $B \neq \mathbf{0}$, we have $\frac{\text{tr}^2(B^T B)}{\text{tr}(B^T B B^T B)} \leq M$.

Proof. Let $Q = B^T B \in \mathbb{R}^{M \times M}$, and $\nu_i (i = 1, 2, \dots, M)$ be the eigenvalue of Q . Then we have $Q^T = (B^T B)^T = B^T B = Q$, $\text{tr}(B^T B) = \text{tr}(Q)$, $\text{tr}(B^T B B^T B) =$

$\text{tr}(Q^T Q) = \text{tr}(Q^2)$ and $\nu_i^2 (i = 1, 2, \dots, M)$ is an eigenvalue of Q^2 . Considering $L = \sum_{i=1}^M (\nu_i - \bar{\nu})^2 \geq 0$, where $\bar{\nu} = \frac{\text{tr}(Q)}{M}$, we have

$$\begin{aligned}
 L &= \sum_{i=1}^M (\nu_i - \bar{\nu})^2 \\
 &= \sum_{i=1}^M \nu_i^2 - 2 \cdot \sum_{i=1}^M \nu_i \cdot \bar{\nu} + M \bar{\nu}^2 \\
 &= \text{tr}(Q^2) - 2 \cdot \text{tr}(Q) \bar{\nu} + M \bar{\nu}^2 \\
 &= \text{tr}(Q^2) - 2M \bar{\nu} \cdot \bar{\nu} + M \bar{\nu}^2 \\
 &= \text{tr}(Q^2) - M \cdot \frac{\text{tr}^2(Q)}{M^2} \geq 0
 \end{aligned} \tag{5.21}$$

Thus, $\frac{\text{tr}^2(Q)}{\text{tr}(Q^2)} \leq M \Rightarrow \frac{\text{tr}^2(B^T B)}{\text{tr}(B^T B B^T B)} \leq M$. $\square$

Lemma 5.8. For symmetric matrices $X, Y \in \mathbb{R}^{N \times N}$ and random matrices $U \in \mathbb{R}^{M \times N}$, $Z \in \mathbb{R}^{N \times M}$, $\text{tr}(UXYZ) = \text{tr}(UYXZ)$, as long as ZU is symmetric [59].

Proof. From $\text{tr}(Q^T) = \text{tr}(Q)$ and $\text{tr}(QGJ) = \text{tr}(GJQ) = \text{tr}(JQG)$, $\forall Q, G, J$, we have $\text{tr}(UXYZ) = \text{tr}(XYZU) = \text{tr}((XYZU)^T) = \text{tr}((ZU)^T Y^T X^T) = \text{tr}(ZUYX) = \text{tr}(UYXZ)$. $\square$

Theorem 5.1. Suppose that λ_k is selected as a positive parameter such that $\widetilde{F(B_k)}$ in (5.13) is positive definite. Then for a randomly chosen $B_0 \in \mathbb{R}^{N \times M}$ satisfying Assumption 5.1, the proposed OPGM in (5.13)-(5.15) with a sufficiently small η ensures that i) $E(B_k)$ is non-increasing, and ii) all (A, B_k) are controllable, and iii) $E(B_k)$ converges to $E(B^*)$ as $k \rightarrow \infty$, where B^* is a matrix having orthonormal columns.

Proof. Please see Appendix B. $\square$

Remark 5.5. Note that for the non-convex optimization problem (5.2) or (5.3), the proposed OPGM method, like any optimization method with a reasonably low complexity, can only guarantee converging to a local minimum. However, an interesting observation based on our extensive implementations of OPGM 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 OPGM is steadily close to the global optimum requests further careful studies.

5.5 Simulation Examples

In this section, we start with illustrating the effectiveness of ImLQR by implementing OPGM. We present some numerical results and analysis in minimizing the position-dependent control cost on a directed stem in subsection 5.5.1. Additionally, we show some interesting results for selecting key nodes by considering the average control cost in a directed stem, a directed circle, and a small random network as well. Unless otherwise specified, we implement OPGM with a randomly given B_0 matrix.

5.5.1 Effect of α in ImLQR

Denoting that $E_{\mathbf{u}}(t_f, B) = \int_0^{t_f} \mathbf{u}^T(t, B)\mathbf{u}(t, B)dt$ and $E_{\mathbf{x}}(t_f, B) = \int_0^{t_f} \mathbf{x}^T(t, B)\mathbf{x}(t, B)dt$, we have $E(t_f, B) = \alpha E_{\mathbf{u}}(t_f, B) + (1 - \alpha)E_{\mathbf{x}}(t_f, B)$. Note that the cost function $E(t_f, B)$ contains both quadratic terms of system input $\mathbf{u}(t_f, B)$ and system state $\mathbf{x}(t_f, B)$, which are summed with a weighting factor α . Therefore, the weighting factor α changes the weight $E_{\mathbf{u}}(t_f, B)$ involving $\mathbf{u}(t)$ and $E_{\mathbf{x}}(t_f, B)$ involving $\mathbf{x}(t)$ simultaneously. It is interesting to study how the weighting factor α affects $E_{\mathbf{u}}(t_f, B)$ and $E_{\mathbf{x}}(t_f, B)$ separately.

In the following experiments, we denote the node set of the network as $\{\text{Node } 1, \dots, \text{Node } i, \dots, \text{Node } N\}$. Particularly, we consider the network with M control inputs as a directed stem, i.e., there exists a link from Node $i - 1$ to Node i except for the first node. This implies that all nodes are accessible nodes except for the first one. By letting $a_{ij} = 1$ if there is a link from node i and j , the adjacency matrix A is given by:

$$A = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \quad (5.22)$$

TABLE 5.1: The control cost when $M = 1, N = 3$

α	0.001	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$E_x(t_f, B^*)$	0.442	0.482	0.561	0.736	0.820	0.955	1.069	1.045	1.213	1.300	1.313	1.376
$E_u(t_f, B^*)$	9.599	7.833	6.825	5.727	5.422	5.236	5.039	5.033	4.901	4.852	4.847	4.767

We first take $\alpha = 0.01$, $\alpha = 0.5$ and $\alpha = 1$ as examples when $N = 4, M = 2$. The initials $\mathbf{x}_0$ and B_0 are chosen to be the same for these three values of α . Fig. 5.2 show the variations of control cost and inputs with iterations for $\alpha = 0.01$, $\alpha = 0.5$ and $\alpha = 1$, respectively. Clearly, with iterations, the total control cost $E(t_f, B)$ converges to its minimum, which is consistent with Theorem 5.1. we find that both $E_{\mathbf{x}}(t_f, B)$ and $E_{\mathbf{u}}(t_f, B)$ with the converged input matrix B^* are significantly less than their initial values. In Fig. 5.2(a), it is observed that the cost $E_{\mathbf{u}}(t_f, B)$ is still decreasing although its weight is sufficient small in the cost function. Similar behavior can be observed in Fig. 5.2(c) where $E_{\mathbf{x}}(t_f, B)$ also decreases with iterations although its weight equals 0, i.e., $1 - \alpha = 0$.

Fig. 5.3 represents the comparison between the results obtained by OPGM with their initials. It can be seen that, in Fig. 5.3(a), the final inputs $\mathbf{u}(t, B^*)$ obtained by OPGM are much smoother compared with initial inputs $\mathbf{u}(t, B_0)$, indicating that the input energy cost of $E_{\mathbf{u}}(t_f, B^*)$ is much lower than its initial value $E_{\mathbf{u}}(t_f, B_0)$. Same conclusion can be drawn for the system states, from Fig. 5.3(b).

Fig. 5.4 shows the effect of $\alpha \in [0.1, 1]$ on designed input $\mathbf{u}(t, B^*)$ and system state $\mathbf{x}(t, B^*)$. Note that when α is decreasing, the weight of input energy in the objective function is also decreasing such that fluctuation of input is generated. A heavier penalty imposed on the transient state, results in a much smoother state trajectory, demonstrated in Fig. 5.4(b). These two figures directly demonstrate the effectiveness of our proposed algorithm. To further investigate how the two parts of the cost function vary with the weight factor α , we take $N = 3, M = 1$ and $N = 4, M = 2$ as examples and run the algorithm for different values of α for 10 times independently with randomly initialized B_0 . The results are shown in Fig. 5.5. The average costs of 10 experiments for different α are summarized in Table 5.1 and Table 5.2. From these tables and figures, it is observed that with α increasing, the minimum energy cost $E_{\mathbf{x}}(t_f, B^*)$ obtained by OPGM is increasing while the minimum energy $E_{\mathbf{u}}(t_f, B^*)$ is decreasing. The lines in Fig. 5.5 directly show such a trend.

TABLE 5.2: The control cost when $M = 2, N = 4$

α	0.001	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$E_x(t_f, B^*)$	0.509	0.620	0.730	0.893	1.032	1.098	1.136	1.218	1.238	1.242	1.260	1.273
$E_u(t_f, B^*)$	14.988	10.504	9.379	8.499	8.054	7.958	7.906	7.875	7.863	7.856	7.854	7.841

FIGURE 5.2: **The illustration for convergence of OPGM with different α .** (a) Control cost with iterations when $\alpha = 0.01$. (b) Control cost with iterations when $\alpha = 0.5$. (c) Control cost with iterations when $\alpha = 1$.

FIGURE 5.3: **System inputs and states when $\alpha = 0.5$.** (a) The comparison between system inputs $\mathbf{u}(t, B^*)$ with their initials $\mathbf{u}(t, B_0)$. (b) The comparison between system states $\mathbf{x}(t, B^*)$ with their initials $\mathbf{x}(t, B_0)$.

5.5.2 ImLQR in Erdős-Rényi and scale-free networks

In this subsection, to further validate the effectiveness of ImLQR, we start perform simulations in Erdős - Rényi (ER) and scale-free (SF) networks, where SF networks are generated by Barabási-Albert (BA) model. Note that the generated ER and SF networks have the same size $N = 75$ and the same mean degree $\langle k \rangle = 10$. The numbers of external inputs are also the same $M = 30$. We summarize the results obtained OPGM in Table 5.3. It is observed that as α increases, $E_x(t_f, B^*)$

FIGURE 5.4: **The effect of α on designed inputs and network states.** (a) The comparison of system inputs $\mathbf{u}(t, B^*)$ with different α . (b) The comparison of system states $\mathbf{x}(t, B^*)$ with different α .

FIGURE 5.5: **Control costs for different values of α .** (a) $M = 1, N = 3$ (b) $M = 2, N = 4$

increases while the input energy $E_u(t_f, B^*)$ decreases. Clearly, $E_u(t_f, B^*)$ is 1 to 2 orders lower than its initial value $E_u(t_f, B_0)$. From Table 5.3, it can be drawn the conclusion that the minimum control cost of SF networks is higher than ER networks of the same mean degree, size and external inputs. This suggests that SF networks are generally more difficult to control than ER networks.

TABLE 5.3: ImLQR in ER and SF networks

Network	N	M	$\langle k \rangle$	α	$E_u(t_f, B^*)$	$E_x(t_f, B^*)$	$E_u(t_f, B_0)$	$E_x(t_f, B_0)$
ER	75	30	10	0.1	1.66E03	2.16E02	1.75E04	5.55E03
	75	30	10	0.5	7.95E02	2.26E02	1.61E04	2.40E03
	75	30	10	1	6.74E02	6.18E02	1.29E04	2.20E03
SF	75	30	10	0.1	2.26E03	2.64E02	4.09E04	2.82E04
	75	30	10	0.5	1.10E03	3.82E02	2.67E04	6.80E03
	75	30	10	1	7.80E02	2.19E03	2.13E04	4.87E03

5.5.3 Key nodes selection in directed networks

Note that B^* requires many nodes to be connected to each of the M external control sources, which may not yield a practical solution for large-size networks. A practical and interesting problem therefore is to find the solution under the constraint that key nodes in one-to-one connection to external control sources. We feel that the characteristics of these key nodes may be of importance for the future design of ImLQR in large networks.

Note that an element of link weight B_{ij}^* is a real number, and its absolute value $|B_{ij}^*|$ actually evaluates the importance of node i for the j th external control source. The same as Chapter 3, the importance index of a node defined in Definition 3.1 will be used here. By closely examining importance indexes, determining what kinds of nodes are key nodes in directed stem/circle becomes possible.

For example, consider a directed stem with $N = 20$ and $M = 5$ and a directed circle with $N = 30$ and $M = 6$. In both cases, OPGM starts with a randomly chosen B_0 that ensures the controllability of the network. At beginning, the nodes corresponding to the M largest importance indexes r_i using B_0 are randomly distributed in both the stem and circle as shown in the first subgraph of Figures 5.6 and 5.7, respectively. After employing OPGM, we select M key nodes based on the converged B^* . For a directed stem in Fig. 5.6, the nodes denoted as $\{1, 5, 9, 13, 17\}$ are selected as key nodes. It is observed that these M ($M = 5$) nodes divide the elementary stem equally. For a directed circle, the same phenomena can be also observed, but the selected key node set is not unique. Any set that divides the circle equally could be possible. For example, both $\{1, 6, 11, 16, 21, 26\}$ and $\{5, 10, 15, 20, 25, 30\}$ are possible to be selected as key node sets. Thus, we conclude that key nodes tend to divide an elementary stem/circle equally for a fixed

FIGURE 5.6: **Key nodes in a directed stem.** The key nodes found by OPGM equally divide the stem to achieve the minimum control cost. The highlighted node connected to an external source (“star”) is a key node.

FIGURE 5.7: **Key nodes in a directed circle.** The highlighted node connected to an external source (“star”) is a key node. Similar to the results observed in a stem, the key nodes found by OPGM also equally divide the circle. The only difference is that the key node set that equally divide the circle is not unique. For example, both sets $\{1, 6, 11, 16, 21, 26\}$ and $\{5, 10, 15, 20, 25, 30\}$ can be located as key node set when implementing OPGM.

M when implementing OPGM. This observation may be of importance as it may lead to heuristic algorithm design for the minimum cost control of large networks.

We would like to point out that the key node set selected by OPGM is not unique due to the fact that the formulated optimization problem in (5.2) or (5.3) is not convex. In Fig. 5.8, we illustrate the selected key nodes in a small random network ($N = 30, M = 8$) in one experiment, and such a key node set may be different in different rounds of implementations. However, our proposed OPGM definitely generates one set of key nodes with satisfactory performance. The detailed information is illustrated as follows. After obtaining the M key nodes using OPGM, the input matrix, denoted as $B^{keynodes}$ can be reconstructed by only connecting the key

TABLE 5.4: Average control cost consumed in various real-life networks.

Dataset	Network	N	M	$\mathcal{E}(B_0)$	$\mathcal{E}(B^{keynodes})$
Circuit	circuit-s208[25]	122	77	4.22E06	4.35E03
	circuit-s420[25]	252	133	7.19E08	1.98E04
	circuit-s838[25]	512	297	3.05E07	4.67E04
Social	phy-discuss[86]	231	185	4.51E04	5.59E02
	phys-friend[87]	228	118	3.16E06	1.43E04
Food web	Maspalomas[1]	30	15	5.70E04	3.16E02
	Rhode[1]	25	15	5.40E04	5.48E03
	CrystalC[88]	24	17	4.30E06	1.00E02
	Florida[88]	128	79	5.33E05	1.02E03

FIGURE 5.8: **Illustration of the selected key nodes in a small random network.** (selected key nodes are indicated by highlighted nodes).

nodes with the external control sources: if node i is the j -th node in the key node set, we set $B_{ji}^{keynodes} = 1$, otherwise, $B_{ji}^{keynodes} = 0$. Note that the controllability of $(A, B^{keynodes})$ can be normally satisfied for a relative large M . As seen in Table 5.4, compared with a random B_0 which is usually a dense matrix, the average control cost can be generally improved by up to 2-4 orders of magnitude in various real-life networks.

5.5.4 Discussions

Note that the input $\mathbf{u}(t, B)$ is an approximate solution to the optimization problem (5.2) or (5.3). However, through extensive simulations, evident influences of weighting factor α on the both system inputs and states are shown. On the other hand, this work is designed by open-loop control strategy, as mentioned in Introduction, it is much more difficult in designing the closed-loop optimal control

scheme in complex networks due to the involved Raccati equation. In our future work, we will develop such a closed-loop scheme to make the system more robust to uncertainties and exotic disturbances.

5.6 Conclusion

In this chapter, we propose an implicit linear quadratic regulator (ImLQR) to address an outstanding issue that LQR becomes unavailable when the input information of LTI systems is incomplete. The main contributions of this chapter are summarized as follows. Firstly, we formulate an optimization model for the optimal control of LTI systems where the input matrix is selectable instead of pre-given. The optimization model is to minimize a matrix based cost function under orthonormal boundary constraint, and the cost function is a weighted combination of both quadratic terms of system input and state. Secondly, with the obtained derivative of a sophisticated matrix-trace based cost function with respect to the input matrix, an orthonormal-condition-based projected gradient method (OPGM) is developed, and its convergence is proven. Based on the solution of the input matrix B^* , the input can be designed. Through simulation studies, we also investigate what kind of nodes is most important for minimizing an average control cost in directed stems/circles and small networks. We see that the key nodes tend to divide the elementary stem and circle equally. This observation may be of importance as it may lead to heuristic algorithm design for large networks. Our results help us to gain better understanding of optimal control of LTI directed networks with selectable input matrices.

Chapter 6

Controlling Directed Networks with Evolving Topologies

In Chapter 3 and Chapter 4, the problems of designing input matrix and selecting key nodes are addressed. As shown in (2.3) and (2.4), it is obvious that the energy cost is also related to the topology A . In this chapter, we begin to investigate the role of topology A in directed networks from optimal control perspective.

Exploring how network topologies affect the cost of controlling directed networks is an important issue in both theory and application. However, its solution still remains open due to the difficulty in analyzing the characteristics of directed networks. In this chapter, a matrix function optimization model is proposed to study how the network topology evolves when the objective is to achieve an optimal control of the networks. To obtain the direction of network topology evolution, a normalized and projected gradient-decent method (NPGM) based on the obtained gradient of the network topology evolution is developed to solve the proposed optimization model. It is proven that NPGM linearly converges to a local minimum point. We further derive an optimality condition to determine whether a converged solution is global minimum or not, and such a condition is also verified through numerous experimental tests on directed networks. We find that a network adaptively changes its topology in such a way that many sub-networks are gradually evolved towards a pre-established control target. Our finding enables us to model and explain how real-world complex networks adaptively self-organize themselves to many similar subnetworks during a relatively long evolution process.

6.1 Introduction

From Eq. (2.4), it is observed that the topology A of a network also affects the control cost. Thus A can be considered as a matrix variable to minimize the cost. We will propose an algorithm method to iteratively find the optimal topology denoted as A^* . Together with this algorithm, a network evolves itself to its optimal topology A^* from its arbitrary initial topology A . In this sense, the considered network is said to have evolving topology owing to the inclusion of the proposed algorithm.

Therefore, how to model the evolution of directed networks for minimizing/maximizing certain criteria becomes an interesting yet still outstanding issue. In this chapter, to address such an issue, we propose a matrix function optimization model with a specified cost to analyze how network topology matrix evolves from the optimal control point of view. Then, based on the model, we aim to achieve the following two objectives: 1) to iteratively determine a network with optimal topology (denoted as A^*) that minimizes the specified cost and 2) to investigate certain characteristics of the resultant optimal network. One of the main difficulties is to establish the gradient/direction of the topology evolution since a matrix exponential function is involved. This requires obtaining the derivative of a defined cost function involving the criteria to be minimized/maximized with respect to the network topology matrix A . Fortunately, by exploiting I-Chain rule [60] and Cayley-Hamilton [119, 120] in this work, the gradient information of the cost function with respect to the network topology matrix is obtained with a closed form of the derivative.

To achieve an optimal control of evolvable directed networks, i.e the first objective, a normalized gradient descent method (NPGM) based on the obtained gradient of the network topology evolution is developed to determine the topology A_k at the k th step, by solving the proposed optimization model. The convergence property and convergence rate of NPGM have also been established. As convergence to a local minimum point of the cost function is only ensured, we further derive an optimality condition to determine whether such converged solution is global minimum or not. Our main idea has also been validated by comprehensive simulation studies on directed networks. Then we explore certain characteristics of the resultant

optimal network, in order to achieve the second objective. Through theoretical analysis and experimental tests, we uncover that, for achieving optimal control of an evolvable directed network consisting of N nodes by exploiting M external control sources, the effective rank [121, 122] of the network topology matrix A_k gradually approaches $N - M$, as it converges to A^* . This implies that all the eigenvalues of $A^{*T}A^*$ are significantly greater than zero, and they are strongly depend on the difference between N and M . When a new external control source is added, the effective rank of matrix A shall slowly evolve and reduce by one accordingly. When N is a multiple of M , we study the eigenvalues of (A, B_i) , in which B_i is the column of B corresponding to the i -th external control source. Interestingly, it is observed that (A, B_i) for all i has almost the same set of eigenvalues. This suggests that the network topology gradually forms $\frac{N}{M}$ subnetworks adaptively, i.e., it can be regarded that the system is decoupled into $\frac{N}{M}$ identical subsystems in some way, with each external control source driving a single subnetwork. Thus, the optimal cost of controlling the whole network is approximately equal to the sum of costs consumed in all such subnetworks. While for the case that N is not a multiple of M , the network can never be divided into M identical subsystems. But experiments show that the lower/upper bound of consumed cost can be easily obtained by utilizing the results obtained when N is a multiple of M . From energy point of view, these findings provide an explanation why various real-world complex networks self-organize to many self-similar subnetworks [123–125] during a relative long evolution history. We conjecture that the evolution of real-world complex networks may just behave like the iteration process of the gradient descent of NPGM. Our observation shall provide wide physical meanings and deepen our understanding of many real life phenomena from the control cost point of view.

The remaining part of this chapter is organized as follows. In Section 6.2, we formulate a matrix function optimization model for the optimal control of directed networks with evolvable topologies. In Section 6.3, how to obtain the gradient of a matrix function with respect to the network topology matrix is discussed by using I-Chain rule, and an algorithm to solve the proposed optimization model is presented. Convergence property of the proposed algorithm is established in Section 6.4, and an optimality condition is analyzed in Section 6.5. In Section 6.6, simulation results showing the validation of our method are presented. In Section 6.7, the proposed algorithm is extended on a fixed topological structure network. Finally, this chapter is concluded in Section 6.8.

6.2 Problem Formulation

Consider the finite-horizon continuous-time system (2.1). The element A_{ij} of the $N \times N$ network topology matrix A describes the connection/interaction strength between node i and node j , and the element B_{im} of the $N \times M$ input matrix B is nonzero if external control source m is connected to node i and zero otherwise. As mentioned in the Introduction section, recently the minimum cost control problem considers the case that A is pre-given while B is a matrix variable [62]. In this work, we instead consider the case that the network topology matrix A becomes variable for a pre-given input matrix B , and explore what kind of role it plays in achieving minimum cost control of complex networks. So the objective is to design A and $\mathbf{u}(t)$, for a given M external control sources, such that the states of system (2.1) can be driven to the origin at $t = t_f$, i.e., $\mathbf{x}(t_f) = \mathbf{0} = [0, \dots, 0]^T$, subject to the condition that the cost $\int_0^{t_f} \|\mathbf{u}(t)\|^2 dt$ can be minimized.

Assumption 6.1. The number of external control sources is sufficient such that (A, B) is controllable at initial time.

Assumption 6.2. The Frobenius norm of the topology matrix is fixed, i.e. $\|A\|_F = \omega$ for some arbitrary parameter ω .

Remark 6.1. Similar to the optimization model in Chapter 3, Assumption 6.1 implies that the number of external control sources should be greater than a lower bound at initial time, i.e., $M \geq N_D$, where N_D is the minimum number of nodes to be connected to external control sources determined by the maximum matching algorithm in [18]. The same as Chapter 3, to study how the network topology affects the control cost, it is reasonable to fix the norm of A in Assumption 6.2. This is to constrain the connection strength of the whole network to a certain level. Note that all matrix norms are equivalent for solving the optimization problem. Without losing generality, we choose the Frobenius norm $\|\cdot\|_F^2$ and set $\omega = N$ in presenting the optimization model later. $\|A\|_F^2 = N$ implies that the matrix norm constraint can be rewritten as a trace equation.

The optimization problem is then formulated as

$$\begin{aligned}
\min \quad & E(t_f, A) = \int_0^{t_f} \mathbf{u}^T(t, A) \mathbf{u}(t, A) dt \\
s.t. \quad & \text{tr}(A^T A) = N \\
& \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \mathbf{x}(t_f) = \mathbf{0},
\end{aligned} \tag{6.1}$$

where $\mathbf{u}(t, A) = [u_1^T(t, A), \dots, u_M^T(t, A)]^T \in \mathbb{R}^{M \times 1}$ means that the input signal to be designed at time t is also dependent on network topology matrix A , and $E(t_f, A)$ is a position-dependent control cost function of t_f and A . In this case, the state vector $\mathbf{x}$ at time t is also dependent on A since

$$\mathbf{x}(t, A) = e^{At} \mathbf{x}_0 + \int_0^t e^{A(t-\tau)} B \mathbf{u}(\tau) d\tau \tag{6.2}$$

Under Assumptions 6.1 and from [50, 115], the control cost $\int_0^{t_f} \|\mathbf{u}(t)\|^2 dt$ can be minimized when the input vector is given by

$$\mathbf{u}(t, A) = -B^T e^{A^T(t_f-t)} W_A^{-1} e^{At_f} \mathbf{x}_0 \tag{6.3}$$

where $W_A = \int_0^{t_f} e^{At} B B^T e^{A^T t} dt$ is the Gramian matrix depending on A when t_f and B are pre-given. At the same time, at time t_f , the constraint $\mathbf{x}(t_f) = \mathbf{0}$ can be satisfied.

By substituting (6.3) into the cost function in (6.1), we have

$$\begin{aligned}
& \int_0^{t_f} \mathbf{u}^T(t, A) \mathbf{u}(t, A) dt \\
&= \int_0^{t_f} \mathbf{x}_0^T e^{A^T t_f} W_A^{-T} e^{A(t_f-t)} B B^T e^{A^T(t_f-t)} W_A^{-1} e^{At_f} \mathbf{x}_0 dt \\
&= \mathbf{x}_0^T e^{A^T t_f} W_A^{-T} \cdot W_A \cdot W_A^{-1} e^{At_f} \mathbf{x}_0 \\
&= \mathbf{x}_0^T e^{A^T t_f} W_A^{-1} e^{At_f} \mathbf{x}_0 \\
&= \text{tr} \left(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} \right)
\end{aligned} \tag{6.4}$$

Thus, for a pre-given t_f , the matrix function optimization model is given by

$$\begin{aligned} \arg \min_A E(A) &= \text{tr} \left(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} \right) \\ \text{s.t.} \quad &\text{tr}(A^T A) = N, \quad \mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \quad (6.5)$$

where A is a matrix variable to be determined, and $E(A) = \text{tr}(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f})$ denotes the position-dependent control cost with $\mathbf{x}_0 = [x_1(0), \dots, x_N(0)]^T$ being the initial state vector.

The model in (6.5) actually refers to the position-dependent control cost as it depends on the initial state $\mathbf{x}_0$. We also consider the average control cost, which is the sum all realizations of the position-dependent control cost for all possible initial states $\mathbf{x}_0$. Similar to previous chapter, assume that each element of initial state $\mathbf{x}_0 = [x_1(0), \dots, x_N(0)]^T$ is an identical independent distributed (i.i.d) variable with zero mean and variance 1. This assumption leads to $\mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] = I_N$.

In this sense, the minimization of the average control cost leads to the following matrix function optimization model

$$\begin{aligned} \arg \min_A \mathcal{E}(A) &= \text{tr} \left(W_A^{-1} e^{At_f} \mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] e^{A^T t_f} \right) \\ \text{s.t.} \quad &\text{tr}(A^T A) = N \end{aligned} \quad (6.6)$$

where $\mathcal{E}(A) = \text{tr}(W_A^{-1} e^{At_f} \mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] e^{A^T t_f})$ denotes the average control cost. When the initial state $x_i(0)$ ($i = 1, \dots, N$) is an identical independent distributed (i.i.d) variable with zero mean and variance 1, we have $\mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] = I$.

6.3 Normalized and Projected Gradient-decent Method

In the subsequent sections, it will be shown how the optimization problem formulated in equations (6.5) and (6.6) can be solved by searching a converged A^* iteratively. One immediate question is to get the information of how the network topology matrix A affects the objective functions in (6.5) and (6.6). Therefore, we first need to obtain the derivative of the control cost function with respect to the

network topology matrix, i.e., $\frac{\partial \text{tr}(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f})}{\partial A}$. Then, the algorithm named “normalized and projected gradient-decent method” is presented.

6.3.1 The derivative of the objective function with respect to network topology

It is known that the matrix exponential function e^{At} can be expressed as $e^{At} = I + At + \frac{1}{2}(At)^2 + \frac{1}{3!}(At)^3 + \dots$ which is an infinite series. To obtain the a closed form of e^{At} , we introduce the following Lemma.

Lemma 6.1. [126] Assume that matrix A has k eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$ where λ_1 is a m -th order repeated eigenvalue and $\lambda_2, \lambda_3, \dots, \lambda_k$ ($k = N - m + 1$) are the remaining $N - m$ eigenvalues which are assumed distinct. By Cayley-Hamilton theorem [119, 120], it is obtained that

$$\begin{aligned} e^{At} &= \alpha_0(t)I + \alpha_1(t)A + \alpha_2(t)A^2 + \dots + \alpha_{N-1}(t)A^{N-1} \\ e^{A^T t} &= \alpha_0(t)I + \alpha_1(t)A^T + \alpha_2(t)(A^2)^T + \dots + \alpha_{N-1}(t)(A^{N-1})^T \end{aligned} \quad (6.7)$$

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

$$\begin{pmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{m-1}(t) \\ \alpha_m(t) \\ \vdots \\ \alpha_{N-1}(t) \end{pmatrix} = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{N-1} \\ 0 & 1 & 2\lambda_1 & \dots & (N-1)\lambda_1^{N-2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & \dots & \dots & (N-1)(N-2)\dots(N-m+1)\lambda_1^{N-m} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^{N-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \lambda_k & \lambda_k^2 & \dots & \lambda_k^{N-1} \end{pmatrix}^{-1} \begin{pmatrix} e^{\lambda_1 t} \\ t e^{\lambda_1 t} \\ \vdots \\ t^{m-1} e^{\lambda_1 t} \\ e^{\lambda_2 t} \\ \vdots \\ e^{\lambda_k t} \end{pmatrix}$$

Lemma 6.2. The derivative of $[W_A^{-1}]_{mu}$ with respect to A_{ij} is given by

$$\begin{aligned} & \frac{\partial [W_A^{-1}]_{mu}}{\partial A_{ij}} \\ &= - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} A^{k_0}]_{mi} [A^{n_0-1-k_0} B B^T e^{A^T t} W_A^{-1}]_{ju} \right) \right) dt \\ & \quad - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} e^{At} B B^T (A^{n_0-1-k_0})^T]_{mj} [(A^{k_0})^T W_A^{-1}]_{iu} \right) \right) dt \end{aligned} \quad (6.8)$$

Proof. By using the I-Chain rule in Lemma 2.2, we have

$$\begin{aligned}
& \frac{\partial[W_A^{-1}]_{mu}}{\partial A_{ij}} \\
&= \frac{\partial[W_A^{-1}]_{mu}}{\partial[W_A]_{fr}} \cdot \frac{\partial[W_A]_{fr}}{\partial A_{ij}} \\
&= -[W_A^{-1}]_{mf} \cdot [W_A^{-1}]_{ru} \cdot \frac{\partial \left[\int_0^{t_f} e^{At} B B^T e^{A^T t} dt \right]_{fr}}{\partial A_{ij}} \\
&= -[W_A^{-1}]_{mf} \cdot [W_A^{-1}]_{ru} \cdot \int_0^{t_f} \frac{\partial[e^{At}]_{fs}}{\partial A_{ij}} \cdot [B B^T]_{sd} \cdot [e^{A^T t}]_{dr} dt \\
&\quad - [W_A^{-1}]_{mf} \cdot [W_A^{-1}]_{ru} \cdot \int_0^{t_f} [e^{At}]_{fs} \cdot [B B^T]_{sd} \cdot \frac{\partial[e^{A^T t}]_{dr}}{\partial A_{ij}} dt
\end{aligned} \tag{6.9}$$

Since

$$\frac{\partial[A^{n_0}]_{kl}}{\partial A_{ij}} = \sum_{k_0=0}^{n_0-1} ([A^{k_0}]_{ki} \cdot [A^{n_0-1-k_0}]_{jl}) \tag{6.10}$$

From Lemma 6.1, we obtain

$$\frac{\partial[e^{At}]_{kl}}{\partial A_{ij}} = \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} ([A^{k_0}]_{ki} \cdot [A^{n_0-1-k_0}]_{jl}) \right) \tag{6.11}$$

and

$$\frac{\partial[e^{A^T t}]_{kl}}{\partial A_{ij}} = \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} ([A^{k_0}]_{li} \cdot [A^{n_0-1-k_0}]_{jk}) \right) \tag{6.12}$$

Finally, by substituting (6.11) and (6.12) into (6.9), we obtain

$$\begin{aligned}
& \frac{\partial [W_A^{-1}]_{mu}}{\partial A_{ij}} \\
&= -[W_A^{-1}]_{mf} \cdot [W_A^{-1}]_{ru} \cdot \int_0^{t_f} \sum_{n_0=1}^{N-1} \frac{\partial [e^{At}]_{fs}}{\partial A_{ij}} [BB^T]_{sd} [e^{A^T t}]_{dr} dt \\
&\quad - [W_A^{-1}]_{mf} \cdot [W_A^{-1}]_{ru} \cdot \int_0^{t_f} [e^{At}]_{fs} [BB^T]_{sd} \frac{\partial [e^{A^T t}]_{dr}}{\partial A_{ij}} dt \\
&= -[W_A^{-1}]_{mf} \cdot [W_A^{-1}]_{ru} \cdot \int_0^{t_f} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([A^{k_0}]_{fi} [A^{n_0-1-k_0}]_{js} \right) \right) [BB^T]_{sd} [e^{A^T t}]_{dr} dt \\
&\quad - [W_A^{-1}]_{mf} \cdot [W_A^{-1}]_{ru} \cdot \int_0^{t_f} [e^{At}]_{fs} [BB^T]_{sd} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([A^{k_0}]_{ri} [A^{n_0-1-k_0}]_{jd} \right) \right) dt \\
&= - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} A^{k_0}]_{mi} [A^{n_0-1-k_0} BB^T e^{A^T t} W_A^{-1}]_{ju} \right) \right) dt \\
&\quad - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} e^{At} BB^T (A^{n_0-1-k_0})^T]_{mj} [(A^{k_0})^T W_A^{-1}]_{iu} \right) \right) dt
\end{aligned} \tag{6.13}$$

□

Lemma 6.3. *The derivative of the position-dependent control cost function in (6.5) with respect to the network topology matrix A is given by*

$$\begin{aligned}
& \frac{\partial \text{tr}(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f})}{\partial A} \\
&= -2 \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left((A^T)^{k_0} W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} W_A^{-1} e^{At} BB^T (A^T)^{n_0-1-k_0} \right) \right) dt \\
&\quad + 2 \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \left(\sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T (A^T)^{n_0-1-k_0} \right)
\end{aligned} \tag{6.14}$$

The derivative of the average control cost function in (6.6) with respect to network topology matrix A can be obtained by replacing $\mathbf{x}_0 \mathbf{x}_0^T$ in (6.14) with an identity matrix I_N .

Proof. Please see Appendix C.

□

6.3.2 Algorithm

Now we are in the position to present an iterative algorithm to solve the optimization problems (6.5) and (6.6). Define a norm function $\mathcal{N}(A) = \text{tr}(A^T A)$, and denote $\Gamma(A)$ as a normalization operator which normalizes A onto a sphere surface $\mathcal{N}(A) = N$. The main idea of normalized and projected gradient-decent method (NPGM) is as follows. Firstly, determine the derivatives of the cost function $E(A)$ (or $\mathcal{E}(A)$) and the norm function $\mathcal{N}(A)$ with respect to matrix A , respectively. Secondly, project the gradient of $E(A)$ (or $\mathcal{E}(A)$) onto the space perpendicular to the gradient of $\mathcal{N}(A)$ (projection). Lastly, drag the projected vector onto the surface $\mathcal{N}(A) = N$ (normalization). This process is repeated until the sequence converges. We will analyze the NPGM in Section 6.4, through which, some ideas proposed in the algorithm will be made clear in ensuring its convergence.

Without of losing generality, we consider the average control cost case. Denote $\{\bullet\}$ as a vector form of an argument matrix $\bullet$ and introduce an projection operator given by

$$\mathcal{P}_{\frac{\partial \mathcal{N}(A_k)}{\partial A_k}} = \left(I - \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\} \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\}^\dagger \right)$$

The normalized and projected gradient-decent method (NPGM) is presented as follows.

step 1) Initialize A randomly as matrix A_0 with a dimension $N \times N$.

step 2) (projection). At iteration k , calculate the gradients $\frac{\partial \mathcal{E}(A_k)}{\partial A_k}$ and its vector version $\left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\}$ at $A = A_k$. Applying the projection operator on $\left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\}$, we obtain

$$\begin{aligned} \left\{ \widetilde{\frac{\partial \mathcal{E}(A_k)}{\partial A_k}} \right\} &= \mathcal{P}_{\frac{\partial \mathcal{N}(A_k)}{\partial A_k}} \left(\left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\} \right) \\ &= \left(I - \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\} \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\} \end{aligned} \quad (6.15)$$

where $\left\{ \widetilde{\frac{\partial \mathcal{E}(A_k)}{\partial A_k}} \right\}$ is a N^2 dimensional vector.

step 3) (normalization). Denote $\mathcal{U}(\cdot)$ as a column-wise operator which reshapes the N^2 dimensional vector $\left\{ \widetilde{\frac{\partial \mathcal{E}(A_k)}{\partial A_k}} \right\}$ back to an $N \times N$ matrix $\widetilde{\frac{\partial \mathcal{E}(A_k)}{\partial A_k}}$, and

do the following gradient descent iteration:

$$\begin{aligned} A_{k+1} &= \Gamma \left[A_k - \eta \cdot \mathcal{U} \left(\left\{ \widetilde{\frac{\partial \mathcal{E}(A_k)}{\partial A_k}} \right\} \right) \right] \\ &= \varrho_k \left[A_k - \eta \cdot \mathcal{U} \left(\left\{ \widetilde{\frac{\partial \mathcal{E}(A_k)}{\partial A_k}} \right\} \right) \right] \end{aligned} \quad (6.16)$$

where η is a chosen step length and ϱ_k is a scalar to realize the normalization operator $\Gamma(\cdot)$ such that $\text{tr}(A_{k+1}^T A_{k+1}) = \text{tr}(A_k^T A_k) = N$.

step 4) Calculate the angle θ_{k+1} between $\left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\}$ and $\left\{ \frac{\partial \mathcal{N}(A_k)}{\partial A_k} \right\}$ at $A = A_{k+1}$.

step 5) Update k to $k + 1$. If $\cos(\theta_k) + 1 > \xi$ for a sufficiently small positive ξ , go to Step 2; otherwise, stop.

In the next section, we will prove that $\mathcal{E}(A_k)$ (for $k = 1, 2, \dots$) is a non-increasing sequence which converges to a nonzero local minimum point $\mathcal{E}(A^*)$ as $\cos(\theta_k)$ converges to -1 , or equivalently, as θ_k converges to π . The solution of the matrix A obtained by the algorithm is denoted as A^* .

6.4 Convergence Analysis

In this section, we prove the convergence of NPGM and analyze its convergence rate. Like any optimization method with a reasonably low complexity, NPGM can only guarantee converging to a local minimum. We will further establish an optimality condition in the next section to measure how close the local minimum solution obtained by NPGM is to the global minimum solution.

Theorem 6.1. Under Assumptions 6.1-6.2, the proposed iteration process of NPGM ensures that i) the average control cost $\mathcal{E}(A_k)$ is non-increasing as k increases and ii) converges to a nonzero local minimum point $\mathcal{E}(A^*)$ linearly, and iii) all (A_k, B) are controllable.

Proof. To drag $\{A_{k+1}\}$ onto the surface $\text{tr}(A^T A) = N$, we introduce a scalar ϱ_k to realize the normalization operator $\Gamma(A)$ in (6.16) by letting

$$\begin{aligned} & \{A_{k+1}\} \\ &= \Gamma \left[\{A_k\} - \eta \left(I - \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\} \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\} \right] \\ &= \varrho_k \left[\{A_k\} - \eta \left(I - \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\} \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\} \right] \end{aligned} \quad (6.17)$$

in NPGM such that $\text{tr}(A_{k+1}^T A_{k+1}) = \text{tr}(A_k^T A_k) = N$, i.e., $\{A_{k+1}\}^T \{A_{k+1}\} = \{A_k\}^T \{A_k\} = N$. As

$$\begin{aligned} \{A_{k+1}\} &= \{A_k\} + (\varrho_k - 1)\{A_k\} \\ &\quad - \eta \varrho_k \left(I - \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\} \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\} \end{aligned} \quad (6.18)$$

then by denoting

$$P_A = \left(I - \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\} \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\}^\dagger \right) \quad (6.19)$$

we have

$$\{A_{k+1}\} = \{A_k\} + (\varrho_k - 1)\{A_k\} - \eta \varrho_k \cdot P_A \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\} \quad (6.20)$$

and

$$\begin{aligned} & \text{tr}(A_{k+1}^T A_{k+1}) \\ &= \varrho_k^2 \left(\text{tr}(A_k^T A_k) + \eta^2 \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\}^T P_A^T P_A \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\} \right) \\ &= \{A_k\}^T \{A_k\} \end{aligned} \quad (6.21)$$

By letting $\mathbf{e} = P_A \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\}$, we obtain that

$$\begin{aligned} \varrho_k &= \sqrt{\frac{\text{tr}(A_k^T A_k)}{\left(\text{tr}(A_k^T A_k) + \eta^2 \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\}^T P_A^T P_A \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\} \right)}} \\ &= 1 + \frac{1}{2} \eta^2 \frac{\mathbf{e}^T \mathbf{e}}{N} + o(\eta^2) \end{aligned} \quad (6.22)$$

where $o(\eta^2)$ denotes a higher order infinitesimal term compared with η^2 , i.e., $\lim_{\eta \rightarrow 0} \frac{o(\eta^2)}{\eta^2} = 0$. So $\varrho_k - 1$ is infinitesimal of higher order infinitesimal term of η , which is denoted as

$$\varrho_k - 1 \sim o(\eta) \quad (6.23)$$

Finally, based on Taylor expansion[127], it is obtained that

$$\begin{aligned} & \mathcal{E}(A_{k+1}) \\ &= \mathcal{E}(A_k) + (\varrho_k - 1) \{A_k\}^T \left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\} - \eta \varrho_k \left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\}^T P_A \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \\ &= \mathcal{E}(A_k) - \eta \varrho_k \left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\}^T P_A^2 \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \\ &\leq 0 \end{aligned} \quad (6.24)$$

The iterations end when $\left\{ -\frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\}$ concurs with the direction of $\left\{ \frac{\partial \mathcal{N}(A_k)}{\partial A_k} \right\}$, i.e.,

$$\left(I - \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\} \left\{ \frac{\partial \mathcal{N}(A)}{\partial A} \right\}^+ \right) \left\{ \frac{\partial \mathcal{E}(A)}{\partial A} \right\} = 0 \quad (6.25)$$

Thus, $\cos(\theta_k)$ converges to -1 and a converged A^* is obtained. In addition, let $\varrho'_k = \varrho_k \left\{ \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \right\}^T P_A^2 \frac{\partial \mathcal{E}(A_k)}{\partial A_k} \geq 0$ we have

$$\mathcal{E}(A_k) = \mathcal{E}(A_{k+1}) + \eta \varrho'_k \quad (6.26)$$

Subtracting $\mathcal{E}(A^*)$ on both sides, we have

$$\mathcal{E}(A_k) - \mathcal{E}(A^*) = \mathcal{E}(A_{k+1}) - \mathcal{E}(A^*) + \eta \varrho'_k \quad (6.27)$$

which gives

$$\frac{\mathcal{E}(A_{k+1}) - \mathcal{E}(A^*)}{\mathcal{E}(A_k) - \mathcal{E}(A^*)} = 1 - \frac{\eta \varrho'_k}{\mathcal{E}(A_k) - \mathcal{E}(A^*)} \quad (6.28)$$

As $k \rightarrow +\infty$, we have $\lim_{k \rightarrow +\infty} \frac{\mathcal{E}(A_{k+1}) - \mathcal{E}(A^*)}{\mathcal{E}(A_k) - \mathcal{E}(A^*)} = 1$, indicating that algorithm converges linearly.

On the other side, since the value of the cost function $\mathcal{E}(A_k)$ keeps strictly non-increasing as the iteration goes on, the network controllability is naturally guaranteed at each iteration step k . Because once (A_k, B) becomes uncontrollable, the Gramian matrix W_{A_k} becomes non-invertible. This leads to an infinite $\mathcal{E}(A_k)$, which is in contradiction to the non-increasing of $\mathcal{E}(A_k)$ during the iteration process. $\square$

Note that the above Theorem 6.1 still holds when position-dependent control cost $E(A)$ is considered.

6.5 Optimality Condition

As mentioned, Theorem 6.1 only ensures convergence to a local minimum point of the cost function. Indeed, NPGM, like any optimization method with a reasonably low complexity, can only find a local minimum as the optimization problem in (6.5) or (6.6) is not convex. Although the global minimum of (6.5) or (6.6) is difficult to obtain, we present an optimality condition in this section to determine whether a solution obtained by NPGM is global minimum or not.

Let $P \in \mathbb{R}^{N \times N}$ be an orthogonal matrix such that $P^T P = I_N$. By defining an orthogonal transformation $\tilde{\mathbf{x}} = P\mathbf{x}$, we have

$$P^{-1}\dot{\tilde{\mathbf{x}}} = AP^{-1}\tilde{\mathbf{x}} + B\mathbf{u} \quad (6.29)$$

which gives

$$\dot{\tilde{\mathbf{x}}} = PAP^{-1}\tilde{\mathbf{x}} + PB\mathbf{u} \quad (6.30)$$

Denote that $\tilde{A} = PAP^{-1}$ and $\tilde{B} = PB$. The following lemma shows that orthogonal transformation does not change the optimization problem given in (6.5).

Lemma 6.4. *The optimization problem*

$$\begin{aligned} \min \quad & E(\tilde{A}) = \text{tr} \left(\left(\int_0^{t_f} e^{\tilde{A}t} \tilde{B} \tilde{B}^T e^{(\tilde{A})^T t} dt \right)^{-1} e^{\tilde{A}t_f} \tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T e^{\tilde{A}^T t_f} \right) \\ \text{s.t.} \quad & \text{tr}(\tilde{A}^T \tilde{A}) = N \end{aligned} \quad (6.31)$$

is equivalent to (6.5).

Proof. Noting that $e^{P\tilde{A}P^T t} = Pe^{\tilde{A}t}P^T$ and $\tilde{\mathbf{x}}_0 = P\mathbf{x}_0$, we have

$$\begin{aligned}
& E(\tilde{A}) \\
&= \text{tr} \left(\left(\int_0^{t_f} Pe^{At}P^T P B B^T P^T Pe^{(A)^T t} P^T dt \right)^{-1} Pe^{(A)t_f} P^T P \mathbf{x}_0 \mathbf{x}_0^T P^T Pe^{(A)^T t_f} P^T \right) \\
&= \text{tr} \left(\left(\int_0^{t_f} Pe^{At} B B^T e^{A^T t} P^T dt \right)^{-1} Pe^{(A)t_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} P^T \right) \\
&= \text{tr} \left(P \left(\int_0^{t_f} e^{At} B B^T e^{A^T t} dt \right)^{-1} P^T Pe^{(A)t_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} P^T \right) \\
&= \text{tr} \left(\left(\int_0^{t_f} e^{At} B B^T e^{A^T t} dt \right)^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} P^T P \right) \\
&= \text{tr} \left(\left(\int_0^{t_f} e^{At} B B^T e^{A^T t} dt \right)^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} \right) \\
&= E(A)
\end{aligned} \tag{6.32}$$

On the other hand,

$$\begin{aligned}
\text{tr}(\tilde{A}^T \tilde{A}) &= \text{tr}(P A^T P^T P A P^T) \\
&= \text{tr}(P A^T A P^T) \\
&= \text{tr}(A^T A) \\
&= N
\end{aligned} \tag{6.33}$$

Thus, this lemma holds. $\square$

Lemma 6.5. For a matrix $W \in \mathbb{R}^{N \times M}$ ($N \geq M$), assume that $\lambda_1 \geq 0, \dots, \lambda_M \geq 0$ are the M eigenvalues $W^T W$. Then, the following optimization problem

$$\begin{aligned}
\min \quad & \text{tr}((W^T W)^{-1}) \\
\text{s.t.} \quad & \text{tr}(W^T W) = C
\end{aligned} \tag{6.34}$$

attains its global minimum value M^2/C if and only if $\lambda_1 = \dots = \lambda_M$.

Proof. The constraint $\text{tr}(W^T W) = C$ implies that

$$\lambda_1 + \dots + \lambda_M = C \quad (6.35)$$

We also have that $\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_M}$ are M eigenvalues of $(W^T W)^{-1}$. Then,

$$\text{tr}((W^T W)^{-1}) = \left(\frac{1}{\lambda_1} + \dots + \frac{1}{\lambda_M}\right) \quad (6.36)$$

By Cauchy-Schwarz inequality,

$$\begin{aligned} & \left(\frac{1}{\lambda_1} + \dots + \frac{1}{\lambda_M}\right)(\lambda_1 + \dots + \lambda_M) \\ & \geq \left(\frac{1}{\lambda_1}\lambda_1 + \dots + \frac{1}{\lambda_M}\lambda_M\right)^2 = M^2 \end{aligned} \quad (6.37)$$

Thus, we obtain

$$\text{tr}((W^T W)^{-1}) \geq M^2/C \quad (6.38)$$

and the equality holds if and only if all the eigenvalues are all equal each other. $\square$

For a matrix $W_i \in \mathbb{R}^{N \times M}$ where $1 \leq i \leq \mathcal{M}$ and $M < N$, assume that $\Lambda_i \in \mathbb{R}^{M \times M}$ is an M dimensional diagonal matrix corresponding to the M eigenvalues of $W_i^T W_i$. We have the following theorem.

Theorem 6.2. If $W_i W_i^T$ and $W_j W_j^T$ ($i \neq j$) are all perpendicular to each other, and $W_1 W_1^T + W_2 W_2^T + \dots + W_{\mathcal{M}} W_{\mathcal{M}}^T$ is a full rank matrix, then the following optimization problem

$$\begin{aligned} \min \quad & \text{tr}((W_1 W_1^T + W_2 W_2^T + \dots + W_{\mathcal{M}} W_{\mathcal{M}}^T)^{-1}) \\ \text{s.t.} \quad & \text{tr}(W_1^T W_1) + \dots + \text{tr}(W_{\mathcal{M}}^T W_{\mathcal{M}}) \leq C \end{aligned} \quad (6.39)$$

attains its minimum if and only if $\Lambda_1 = \Lambda_2 = \dots = \Lambda_{\mathcal{M}}$.

Proof. Noting that $W_i W_i^T$ is a normal matrix, we have

$$W_i W_i^T = U_i \Lambda_i U_i^T \quad (6.40)$$

where $U_i \in \mathbb{R}^{N \times M}$, satisfying that $U_i^T U_i = I_M$. As $W_i W_i^T$ and $W_j W_j^T$ for $1 \leq i, j \leq \mathcal{M}$, are all perpendicular to each other, and $W_1 W_1^T + \dots + W_i W_i^T + \dots + W_{\mathcal{M}} W_{\mathcal{M}}^T$ is of full rank,

$$\begin{aligned} & \text{tr} (W_1 W_1^T + \dots + W_i W_i^T + \dots + W_{\mathcal{M}} W_{\mathcal{M}}^T)^{-1} \\ &= \text{tr} (P (\Lambda_1 \oplus \dots \oplus \Lambda_i \oplus \dots \oplus \Lambda_{\mathcal{M}}) P^T)^{-1} \\ &= \text{tr} (P \Lambda P^T)^{-1} \end{aligned} \quad (6.41)$$

with $P = [U_1 \dots U_i \dots U_{\mathcal{M}}]$ satisfying that $P^T P = I_N$. Here $\oplus$ denotes the direct sum of orthogonal subspaces:

$$\begin{aligned} \Lambda &= \Lambda_1 \oplus \dots \oplus \Lambda_i \oplus \dots \oplus \Lambda_{\mathcal{M}} \\ &= \begin{bmatrix} \Lambda_1 & & & & \\ & \ddots & & & \\ & & \Lambda_i & & \\ & & & \ddots & \\ & & & & \Lambda_{\mathcal{M}} \end{bmatrix} \in \mathbb{R}^{N \times N} \end{aligned} \quad (6.42)$$

So we have

$$\begin{aligned} & \text{tr} (P (\Lambda_1 \oplus \dots \oplus \Lambda_i \oplus \dots \oplus \Lambda_{\mathcal{M}}) P^T)^{-1} \\ &= \text{tr} (P^T (\Lambda_1 \oplus \dots \oplus \Lambda_i \oplus \dots \oplus \Lambda_{\mathcal{M}})^{-1} P) \\ &= \text{tr} ((\Lambda_1 \oplus \dots \oplus \Lambda_i \oplus \dots \oplus \Lambda_{\mathcal{M}})^{-1} P P^T) \\ &= \text{tr} (\Lambda_1 \oplus \dots \oplus \Lambda_i \oplus \dots \oplus \Lambda_{\mathcal{M}})^{-1} \\ &= \text{tr} (\Lambda_1^{-1} + \dots + \Lambda_i^{-1} + \dots + \Lambda_{\mathcal{M}}^{-1}) \end{aligned} \quad (6.43)$$

From Lemma 6.5 and the Cauchy-Schwarz inequality [128], it can be obtained that

$$\begin{aligned} & \text{tr} (\Lambda_1^{-1} + \dots + \Lambda_i^{-1} + \dots + \Lambda_{\mathcal{M}}^{-1}) \geq \\ & \frac{\left(\sqrt{\text{tr}(\Lambda_1^{-1})} \sqrt{\text{tr}(\Lambda_1)} + \dots + \sqrt{\text{tr}(\Lambda_i^{-1})} \sqrt{\text{tr}(\Lambda_i)} + \dots + \sqrt{\text{tr}(\Lambda_{\mathcal{M}}^{-1})} \sqrt{\text{tr}(\Lambda_{\mathcal{M}})} \right)^2}{\text{tr}(\Lambda_1 + \dots + \Lambda_{\mathcal{M}})} \end{aligned} \quad (6.44)$$

and the equality is satisfied if and only if

$$\frac{\text{tr}(\Lambda_1^{-1})}{\text{tr}(\Lambda_1)} = \dots = \frac{\text{tr}(\Lambda_i^{-1})}{\text{tr}(\Lambda_i)} = \dots = \frac{\text{tr}(\Lambda_{\mathcal{M}}^{-1})}{\text{tr}(\Lambda_{\mathcal{M}})} \quad (6.45)$$

Namely

$$\begin{aligned} & \text{tr}((W_1 W_1^T + \dots + W_i W_i^T + \dots + W_{\mathcal{M}} W_{\mathcal{M}}^T)^{-1}) \\ & \geq \frac{\left(\sum_i \sqrt{\text{tr}(\Lambda_i^{-1})} \sqrt{\text{tr}(\Lambda_i)} \right)^2}{\text{tr}(\Lambda_1 + \dots + \Lambda_i + \dots + \Lambda_{\mathcal{M}})} \end{aligned} \quad (6.46)$$

if and only if

$$\frac{\text{tr}(\Lambda_i^{-1})}{\text{tr}(\Lambda_j)} = \frac{\text{tr}(\Lambda_j^{-1})}{\text{tr}(\Lambda_i)} \quad (6.47)$$

for all $1 \leq i, j \leq \mathcal{M}$, which implies that $\Lambda_1 = \dots = \Lambda_i = \dots = \Lambda_{\mathcal{M}}$. $\square$

Recall that we have $M \geq N_D$ from Assumption 6.1. When M is much less than the network size N , Theorem 6.3 below is obtained. $\square$

Theorem 6.3. The optimization model in (6.6) is approximately equivalent to (6.2) under Assumption 6.1 when the number of inputs M is significantly less than the network size N , and the control cost attains its minimum if and only if the whole systems can be regarded as consisting of M identical subsystems.

Proof. As $(e^{At_f})^{-1} = e^{-At_f}$, we have

$$\begin{aligned} \mathcal{E}(A) &= \text{tr} \left(\left(\int_0^{t_f} e^{At} B B^T e^{A^T t} dt \right)^{-1} (e^{-A^T t_f} e^{-At_f})^{-1} \right) \\ &= \text{tr} \left(e^{-A^T t_f} e^{-At_f} \int_0^{t_f} e^{At} B B^T e^{A^T t} dt \right)^{-1} \\ &= \text{tr} \left(\int_0^{t_f} e^{-A^T t_f} e^{-At_f} e^{At} B B^T e^{A^T t} dt \right)^{-1} \end{aligned} \quad (6.48)$$

As $(e^{-A^T t_f} e^{-A t_f})^{-1}$ can be considered as a constant in the above integral, we have

$$\begin{aligned}
\mathcal{E}(A) &= \text{tr} \left(\int_0^{t_f} e^{-A^T t_f} e^{-A t_f} e^{A t} B B^T e^{A^T t} dt \right)^{-1} \\
&= \text{tr} \left(\int_0^{t_f} e^{A(t-t_f)} B B^T e^{A^T(t-t_f)} dt \right)^{-1} \\
&= \text{tr} \left(\int_{-t_f}^0 e^{A t} B B^T e^{A^T t} dt \right)^{-1} \\
&= \text{tr} \left(\int_0^{t_f} e^{-A t} B B^T e^{-A^T t} dt \right)^{-1}
\end{aligned} \tag{6.49}$$

It is known that $\int_0^{t_f} e^{-A t} B B^T e^{-A^T t} dt$ is symmetrical and non-negative. Therefore there exists a matrix $\mathcal{W}$ such that

$$\mathcal{W}^T \mathcal{W} = \int_0^{t_f} e^{-A t} B B^T e^{-A^T t} dt \tag{6.50}$$

Also, Assumption 6.2 implies that $\text{tr}(\mathcal{W}^T \mathcal{W})$ is bounded, which is denoted as $\text{tr}(\mathcal{W}^T \mathcal{W}) \leq \mathcal{C}$. This results in the optimization model (6.34).

On the other hand, let $B = [B_1 \dots B_i \dots B_M]$ where B_i is i th column of the input matrix B corresponding to the i th external control source, and we define a subsystem (A, B_i) which denotes the effect of the i th external control source on the whole network. Then, we have

$$\begin{aligned}
\mathcal{E}(A) &= \text{tr} \left(\int_0^{t_f} e^{-A t} B B^T e^{-A^T t} dt \right)^{-1} \\
&= \text{tr} \left(\int_0^{t_f} e^{-A t} B_1 B_1^T e^{-A^T t} dt + \dots + \int_0^{t_f} e^{-A t} B_i B_i^T e^{-A^T t} dt + \dots \right. \\
&\quad \left. + \int_0^{t_f} e^{-A t} B_M B_M^T e^{-A^T t} dt \right)^{-1}
\end{aligned} \tag{6.51}$$

Denoting that

$$\mathcal{W}_i \mathcal{W}_i^T = \int_0^{t_f} e^{-A t} B_i B_i^T e^{-A^T t} dt \tag{6.52}$$

the cost function of (6.6) can be written as

$$\begin{aligned}
&\text{tr} (\mathcal{W} \mathcal{W}^T)^{-1} \\
&= \text{tr} (\mathcal{W}_1 \mathcal{W}_1^T + \dots + \mathcal{W}_i \mathcal{W}_i^T + \dots + \mathcal{W}_M \mathcal{W}_M^T)^{-1}
\end{aligned} \tag{6.53}$$

where $\text{tr}(\mathcal{W}_1\mathcal{W}_1^T) + \dots + \text{tr}(\mathcal{W}_M\mathcal{W}_M^T) \leq \mathcal{C}$.

When M is much less than N , the dynamics of the networks caused by B_i hardly affect the dynamics of the networks caused by B_j . This means that subsystems (A, B_i) and (A, B_j) ($i \neq j$) become almost independent of each other, i.e., $\mathcal{W}_i\mathcal{W}_i^T$ and $\mathcal{W}_j\mathcal{W}_j^T$ can be considered as perpendicular to each other approximately, which gives

$$\int_0^{t_f} e^{-At} B_i B_i^T e^{-A^T t} dt = U_i \Lambda_i U_i^T + \epsilon_i \quad (6.54)$$

where $\epsilon_i \in \mathbb{R}^{N \times N}$ denotes the approximation error, $U_i \in \mathbb{R}^{N \times M}$, $\Lambda_i \in \mathbb{R}^{M \times M}$ and $U_i^T U_i = I_M$ for $i = 1, \dots, M$ and $U_i \Lambda_i U_i^T$ are perpendicular to $U_j \Lambda_j U_j^T$ as long as $i \neq j$, and ϵ_i can be considered as sufficiently small.

Thus, the optimization model in (6.6) can be regarded as approximately equivalent to the model considered in Theorem 6.2. We have

$$\begin{aligned} \mathcal{E}(A^*) &= \text{tr} \left(\sum_i \Lambda_i + \sum_i \epsilon_i \right)^{-1} \\ &\approx \text{tr} \left(\sum_i \Lambda_i \right)^{-1} \\ &= \sum_i \text{tr}(\Lambda_i)^{-1} \end{aligned} \quad (6.55)$$

where $\mathcal{W}_i\mathcal{W}_i^T$ is a normal matrix and Λ_i is a diagonal matrix containing the largest $\frac{N}{M}$ eigenvalues of $\mathcal{W}_i\mathcal{W}_i^T$, denoted as $\lambda_1^i \geq \lambda_2^i \geq \dots \geq \lambda_{\frac{N}{M}}^i$.

It is known that $\lambda_1^i \gg \lambda_2^i \gg \dots \gg \lambda_{\frac{N}{M}}^i$. Therefore $\text{tr}(\Lambda_i^{-1})$ is mainly dependent on $\frac{1}{\lambda_{\frac{N}{M}}^i}$. We obtain the optimality condition that

$$\mathcal{E}(A^*) = M \left(\frac{1}{\lambda_1^i} + \frac{1}{\lambda_2^i} + \dots + \frac{1}{\lambda_{\frac{N}{M}}^i} \right) \approx M \frac{1}{\lambda_{\frac{N}{M}}^i} \quad (6.56)$$

and the first equality follows when $\Lambda_1 = \dots = \Lambda_i = \dots = \Lambda_M$. $\square$

Based on the optimality condition in (6.56), the global minimum point of the objective function is achieved, i.e., the cost approaches the lower bound, if and only if the whole systems can be regarded as consisting of M identical subsystems. This is

of great significance, since for the non-convex optimization problem (6.1), its global minimum is difficult to obtain. Typically for a non-convex and highly-complex optimization problem, Monte Carlo simulations with different initial conditions may help finding a suboptimal solution. In the next section, examples will be illustrated regarding how the optimality condition can be applied to determine whether a solution obtained by NPGM is global minimum or not.

6.6 Simulation Results

Network topology plays an important role in affecting the cost of controlling a network. In this section, we carry out experiments to show the effectiveness of NPGM whose convergence has been established in Theorem 6.1, and validate the optimality condition established in Theorem 6.3. We present some interesting observations regarding how the topologies of the networks evolves for achieving optimal control.

6.6.1 Convergence illustration of NPGM

We take two examples in which the initial network topology A_0 is randomly generated with a) $N = 15$, $M = 3$ and b) $N = 14$, $M = 3$, respectively. In the first example, N is a multiple of M while the second example is not. Note that for a randomly generated A_0 , A_0 is a full rank matrix almost surely. This implies that the controllability of the network can be naturally guaranteed as long as $M \geq 1$. Without losing generality, we just consider the case that the three external control sources are one to one connected to the first three nodes. In this case, the input matrix B is given by

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix}$$

FIGURE 6.1: **Control cost when $M = 3$ and $N = 15$** FIGURE 6.2: **Control cost when $M = 3$ and $N = 14$**

In Figures 6.1 and 6.2, it is seen that the control cost reduces monotonously as the network topology gradually evolves in the iteration process which ends when $\cos(\theta_k)$ approaches -1 . The convergence of NPGM is validated.

6.6.2 Effective rank of the network topology matrix A^*

Denote the converged network topology matrix found by NPGM as A^* . We investigate what kind of characteristics of an optimal topology A^* has when the minimum control cost is achieved in this subsection. We first study the effective rank of A^* through studying the eigenvalues of $A^{*T}A^*$. The reason of doing this is that A^* is not necessary to be symmetric while $A^{*T}A^*$ is always symmetric and

TABLE 6.1: The effective rank of networks

M	N	$\text{erank}(A^*)$	$N - M$
3	6	3	3
3	14	11	11
3	15	12	12
5	10	5	5
5	15	10	10
5	16	11	11

has non-negative real eigenvalues. Suppose that the eigenvalue set of $A^T A$ is given by $\bar{\lambda} = \{\lambda_1, \dots, \lambda_N\}$ such that $\lambda_1 \geq \dots \geq \lambda_{N_0} \geq \lambda_{N_0+1} \geq \dots \geq \lambda_N \geq 0$. The effective rank of A^* , denoted as “ $\text{erank}(A^*)$ ”, is defined as

$$\text{erank}(A^*) = \min\{N_0 | \lambda_{N_0+1} \leq \epsilon, \frac{\lambda_{N_0}}{\lambda_{N_0+1}} \gg \epsilon, \lambda_{N_0} \in \bar{\lambda}\} \quad (6.57)$$

where ϵ is a pre-given small positive constant. Here $\lambda_{N_0+1} \leq \epsilon$ and $\frac{\lambda_{N_0}}{\lambda_{N_0+1}} \gg \epsilon$ imply that there is a large jump from the eigenvalue λ_{N_0} near zero to λ_{N_0-1} . Actually the effective rank of a matrix is the number of its eigenvalues that are significantly greater than zero. In the experiments, it is observed that the effective rank of A^* always approaches $N - M$ when the minimum cost control of an evolutionary network is obtained, no matter what initial eigenvalues of the randomly initialized $A_0^T A_0$ are. This means that $\text{erank}(A^*)$ is an inherent property of the network topology matrix and independent of its initial condition. Take $N = 6$ and $M = 3$ as an example. When initializing the network topology matrix randomly in one experiment, the eigenvalues of randomly initialized $A_0^T A_0$ are 4.905, 0.959, 0.524, 0.365, 0.130 and 0.017. By applying NPGM, we obtain a converged topology A^* , and observe that the eigenvalues of $A^{*T} A^*$ are 1.999, 1.986, 1.982, 0.008, 0.08 and 0.005. It is seen that the first three values are significantly larger than zero and the last three values are very close to zero. Then, the effective rank of A^* is regarded as 3, which is just $N - M$. We then repeat such a process for a number of times and have the same observation. For other networks, experimental results also show the effective rank of A^* similarly. Table 6.1 summarizes values of $\text{erank}(A^*)$ for some networks with different values of N and M , which verifies that $\text{erank}(A^*) = N - M$.

6.6.3 Control cost when N is a multiple of M

In this subsection, we further consider that N is a multiple of M . To explore the properties of a converged topology A^* , suppose that the N eigenvalues of subsystem (A^*, B_i) are given by $\{\lambda_1^{i*}, \lambda_2^{i*}, \dots, \lambda_N^{i*}\}$. As mentioned, we select the $\frac{N}{M}$ largest eigenvalues, denoted as $\Lambda_i^* = \text{diag}\{\lambda_1^{i*}, \lambda_2^{i*}, \dots, \lambda_{\frac{N}{M}}^{i*}\}$ such that $\lambda_1^{i*} \geq \lambda_2^{i*} \geq \dots \geq \lambda_{\frac{N}{M}}^{i*}$, to characterize the properties of (A^*, B_i) for $i = 1, \dots, M$. We say that subnetworks (A^*, B_i) and (A^*, B_j) are identical iff the $\frac{N}{M}$ largest eigenvalues of Λ_i^* and Λ_j^* are all the same. Throughout the experiments, we find that all subnetworks (A^*, B_i) for all i present almost the same set of eigenvalues. Based on such an observation, we conclude that, when the whole network can be decoupled into M identical subsystems, we say an optimal A^* is obtained. Specifically, the system (A^*, B) can be regarded as the composition of M identical independent subsystems (A^*, B_i) , just like each i -th external control source controlling $\frac{N}{M}$ nodes.

Now we examine the relationship between the cost $\mathcal{E}(A^*)$ and the eigenvalue $\lambda_{\frac{N}{M}}^{*i}$ that is the $\frac{N}{M}$ -th largest eigenvalue for each subsystem (A^*, B_i) . Extensive simulations are conducted to study it and some results are summarized in Table 6.2 and Fig. 6.3. It is observed that the control cost can be approximated by

$$\mathcal{E}(A^*) \approx M \cdot \frac{1}{\lambda_{\frac{N}{M}}^{*i}} \quad \text{for all } \frac{N}{M} \geq 1 \quad (6.58)$$

We conclude that the cost of controlling N nodes with M external control sources, denoted as $\mathcal{E}(A^*)$, can be estimated by the product of M and $\frac{1}{\lambda_{\frac{N}{M}}^{*i}}$.

It is worthy to point out that the observation in (6.58) is consistent with the conclusion in Theorem 6.3: the cost approaches the lower bound if and only if the whole systems can be regarded as consisting of M identical subsystems. This also motivates us to estimate the optimal cost of large size networks through decoupling them into small size networks when the network topology has evolved into a converged A^* . To do the experiment, we consider 1) controlling $N = 10$ nodes with $M = 2$ external control sources, and 2) controlling $N = 500$ nodes with $M = 100$ external control sources. The converged network topology matrices for 1) and 2) obtained by NPGM are denoted as A_1^* and A_2^* , respectively. So the system (A_1^*, B) in 1) can be decoupled into 2 identical subsystems (A_1^*, B_i) ($i = 1, 2$) with each

TABLE 6.2: The relationship between $\frac{1}{\tilde{\lambda}_{\frac{N}{M}}^i}$ and minimum cost when N is a multiple of M

M	N	$\tilde{\lambda}_{\frac{N}{M}}^i$	$\frac{M}{\tilde{\lambda}_{\frac{N}{M}}^i}$	NPGM
3	9	0.00309	993.562	970.873
3	12	2.315×10^{-5}	1.296×10^5	1.317×10^5
3	15	9.471×10^{-8}	3.167×10^7	3.181×10^7
5	10	0.159	31.466	34.298
5	15	0.00305	1.639×10^3	1.684×10^3
5	20	2.333×10^{-5}	2.143×10^5	2.165×10^5
8	24	0.00308	2.597×10^3	2.657×10^3
10	30	0.00262	3.817×10^3	4.036×10^3

FIGURE 6.3: **Control cost when $\frac{N}{M}$ is an integer.** By taking $M = 3$, $M = 4$ and $M = 5$ as examples, the minimum cost is computed by $\frac{M}{\tilde{\lambda}_{\frac{N}{M}}^i}$ and NPGM respectively.

i -th external control source controlling 5 nodes. By denoting $\mathcal{E}_5 = \frac{\mathcal{E}(A_1^*)}{2}$, we find that the control cost of 2) can be estimated as $\mathcal{E}(A_2^*) \approx 100\mathcal{E}_5$.

For more general cases, we consider Case i) controlling $2 \cdot \frac{N}{M}$ nodes with $M = 2$ external control sources ($M = 2$ is the simplest case of multiple external control sources), and Case ii) controlling N nodes with M external control sources where N is an integer multiple of M . The optimal network topology for Cases i) and ii) obtained by NPGM are denoted as A_1^* and A^* , respectively. Denote $\mathcal{E}_{\frac{N}{M}}$ as the half of the optimal control cost $\mathcal{E}(A_1^*)$ corresponding to A_1^* obtained in Case i), i.e., $\mathcal{E}_{\frac{N}{M}} = \frac{\mathcal{E}(A_1^*)}{2}$, which is about the control cost of each subsystem.

From (6.58), we denote $\mathcal{E}_{\frac{N}{M}} = \frac{\mathcal{E}(A_1^*)}{2} \approx \frac{1}{\lambda_{\frac{N}{M}}^{*i}}$. As $\frac{N}{M}$ is the same in both Case i)

FIGURE 6.4: Control cost computed by NPGM with $\frac{N}{M}$ fixed.

and Case ii), $\frac{1}{\lambda^{*i} \frac{N}{M}}$ are also almost the same. Therefore, the control cost $\mathcal{E}(A^*)$ is strongly linear dependent on $\mathcal{E}_{\frac{N}{M}}$ with M being the slope:

$$\mathcal{E}(A^*) \approx M \cdot \mathcal{E}_{\frac{N}{M}}, \quad \frac{N}{M} \geq 2 \quad (6.59)$$

Fig. 6.4 validates that (6.59) holds approximately, where the linear relationship between $\mathcal{E}(A^*)$ and $\mathcal{E}_{\frac{N}{M}}$ are tested under different value sets of $\frac{N}{M}$. Besides, by collecting on the eigenvalues of all subnetworks, it is obvious to determine whether a given topology A^* is optimal or not. If all eigenvalues of all subnetworks are almost the same, it can be regarded that an optimal topology of A^* has been achieved. Otherwise, no optimal topology of A^* is achieved. This is consistent with the conclusion of Theorems 6.2 and 6.3. Basically (6.59) implies that, the cost of a large network can be estimated by the sum of a group of small networks of the same size when the network evolves to an optimal network topology.

6.6.4 Control cost when N is not a multiple of M

Now we consider how to estimate the control cost $\mathcal{E}(A^*)$ of networks when N is not a multiple of M . Denote $\lceil \cdot \rceil$ as the ceiling function, and $\lfloor \cdot \rfloor$ as the floor function. We discover that the lower and upper bound of $\mathcal{E}(A^*)$ can be estimated by calculating the costs of controlling $\lfloor \frac{N}{M} \rfloor \cdot M$ nodes and $\lceil \frac{N}{M} \rceil \cdot M$ nodes with M external controller sources, respectively. For example, consider $N = 100$ and $M = 21$. As $\lfloor \frac{N}{M} \rfloor = 4$ and $\lceil \frac{N}{M} \rceil = 5$, we have $\lfloor \frac{N}{M} \rfloor \cdot M = 84$ and $\lceil \frac{N}{M} \rceil \cdot M = 105$. Then, the lower and upper bounds of $\mathcal{E}(A^*)$ can be calculated by respectively

FIGURE 6.5: **Control cost when $\frac{N}{M}$ is not an integer with $M = 3$.** The bottom of the rectangle is the lower bound of the minimum energy cost while the top is its upper bound.

studying the case of controlling $N = 84$ nodes with $M = 21$, and $N = 105$ nodes with $M = 21$. Fig. 6.5 shows the range of control cost, in which the scatter dots represent the values of $\mathcal{E}(A^*)$ calculated by NPGM when $M = 3$, and the bar graph denotes the lower and upper bounds of $\mathcal{E}(A^*)$ in each case. It is validated that the values of $\mathcal{E}(A^*)$ are within the range of bar graphs.

Based on the experiments and analysis on Sections 6.3 and 6.4, we make the following conclusions. When N is an integer multiple of M , the optimal cost of controlling whole network can be decoupled into controlling M identical subnetworks, with its consumed energy being the sum of the cost of all subnetworks. While for the later case when N is not an integer multiple of M , the M subsystems can never be identical, but the consumed energy bounds can also be easily obtained by using the results in the former case.

6.6.5 Application to social networks

Inspired by the work [39, 129], we take the cooperation problem in social networks as an illustrative example. We aim at understanding how the individuals influence each other with the help of optimal control theory. Now consider a group of N people whose initial opinions are different. Among them, M individuals are directly affected by external signals like leaders, advertisements and so on. With a limited time period t_f , the final network-wide agreement is obtained. This process can be

FIGURE 6.6: The “optimal” topology for cooperation in social networks.

modeled by the continuous-time consensus system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \mathbf{x}(t_f) = \mathbf{0} \quad (6.60)$$

where $\mathbf{x}_0$ represents initial opinions of different individuals and $\mathbf{x}(t_f)$ is the feasible agreement. For brevity and clarity, here we take $N = 6, M = 3$ as an example. The input matrix B is given by $B = [I_3, 0]^T$ where I_3 is a 3×3 identity matrix. By employing our proposed algorithm NPGM, we obtain a topology A^* , through which the control cost is minimum. To make the connections more simple, we apply a small threshold value $\varrho = 0.01$ on each element of A^* . That is, if $A_{ij}^* < \varrho$, we set $A_{ij}^* = 0$. Finally, we get a topology reflecting the opinion dissemination, shown in Fig. 6.6. The weight here illustrates the influence strength by the source. From Fig. 6.6, it is seen that the network adaptively changes its topology in such a way that three almost identical sub-networks are gradually evolved. Specifically, we observe that each individual, directly influenced by external signals, will have an impact on two other persons respectively to help achieve the cooperative state. And a person is mainly persuaded by two other friends. This is reasonable in real life as people are impressionable by others, particularly by one or two such friends.

6.6.6 Robustness test of NPGM

Actually, when designing A with inaccurate information of input matrix, the obtained topology A is not far from the optimal one for the actual network. Here we use the example of social networks in subsection 6.6.5 to test the robustness of proposed algorithm. Compared with Fig. 6.6, the external input in Fig. 6.7 is connected with Node 4 instead of Node 1. For the obtained topology by NPGM, it is seen

FIGURE 6.7: The robustness test of NPGM.

that only the edges connected with Node 1 and Node 4 are affected while other edges remain the same.

6.7 Extension NPGM on Networks with Fixed Structure

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, $\mathcal{A}_{ij} = 0$. Without losing generality, suppose that $(\mathcal{A}, B)$ is controllable. Then it is easy to see that Assumptions 6.1-6.2 can be guaranteed by setting $A = \mathcal{A}$ at initial time. However, to guarantee that the network structure matrix $\mathcal{A}$ remains fixed in the iterative process of NPGM, the gradient $\frac{\partial \mathcal{E}(A)}{\partial A}$ needs to be replaced by $\frac{\partial \mathcal{E}(A)}{\partial A} \circ \mathcal{A}$, where the ‘ $\circ$ ’ denotes the Hadamard (elementwise) product of two matrices. Note that Theorems 6.1-6.3 still hold when implementing NPGM in this way as there is no requirement on specific form of the gradient. In the following, some interesting observations regarding how the edge weight of networks changes for achieving optimal control are presented.

6.7.1 Clustering the network into M control-flow subnetworks

To explore the characteristics of optimal A^* with a preserving topological structure $\mathcal{A}$, we take experiments on three typical network structure, i.e., (a_1) , (b_1) and (c_1) shown in Fig. 6.8 into consideration. Taking (a_1) as an example, the network structure matrix is built as

$$\mathcal{A} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

Obviously, Assumptions 6.1-6.2 are satisfied. Then we implement NPGM, with the replacement of the gradient $\frac{\partial \mathcal{E}(A)}{\partial A}$ by $\frac{\partial \mathcal{E}(A)}{\partial A} \circ \mathcal{A}$ at each iteration, in order to find out how the connection strength evolves when the control cost attains its minimum. The effective rank of A^* is 4, which verifies that $erank(A^*) = N - M$.

With control sources (stars) connecting different nodes, the corresponding optimal networks obtained by NPGM are presented in the second and third columns of Fig. 6.8 where the dash line means the connection strength approaches zero. From our experiment, it is noted that the link weights evolve depending on the distances (range) of the control sources. By ignoring the extremely small connection strengths, it can be observed that when the control cost attains minimum, M subnetworks are self-formed. In each subnetwork, it just contains a stem or a stem-cycle (stem connected with cycles). As such a subnetwork reflects the control path with a single input, we call such a subnetwork as *control-flow* network. Fig. 6.8 (a_2) , (a_3) , (b_2) and (b_3) show that two stems are evolved naturally. For (c_2) , (c_3) , two subnetworks are a stem and a stem-cycle, marked by purple color and red color respectively.

In addition, Fig. 6.9 clearly presents how the interaction strengths of nodes change with iterations for networks (a_2) , (b_2) shown in Fig. 6.8.

FIGURE 6.8: Change of interaction strengths when the control sources are pre-allocated.

FIGURE 6.9: Interaction strength changes with iterations for network. A_{ij} is weight of the connection from node j to node i . (a) Elementary stem with $N = 6$, $M = 2$. Nodes $\{1, 3\}$ are connected to external controller sources. (b) Elementary cycle with $N = 6$, $M = 2$. Nodes $\{1, 3\}$ are connected to external controller sources.

TABLE 6.3: Minimum control cost of elementary networks

	Topologies	N	M	t_f	key nodes	$\mathcal{E}(A^*)$
(a)	stem	6	2	1	Nodes $\{1, 2\}$	6.9269×10^6
(b)	stem	6	2	1	Nodes $\{1, 3\}$	1.9722×10^4
(c)	stem	6	2	1	Nodes $\{1, 4\}$	900.7143
(d)	stem	6	2	1	Nodes $\{1, 5\}$	1.9722×10^4
(e)	stem	6	2	1	Nodes $\{1, 6\}$	6.9269×10^6
(f)	cycle	6	2	1	Nodes $\{1, 2\}$	6.9270×10^6
(g)	cycle	6	2	1	Nodes $\{1, 3\}$	1.9723×10^4
(h)	cycle	6	2	1	Nodes $\{1, 4\}$	900.7140

6.7.2 Control cost with different key nodes

As the number of external control sources should be larger than a lower bound, there will be different ways to select a subset of nodes connected with the control sources. In this subsection, we take elementary stems and cycles as examples and obtain the minimum control cost with NPGM. Technically, for $N = 6$, $M = 2$, there are totally 5 ways for stems and 3 ways for cycles to select key nodes connected to external controllers. The detailed networks and their minimum costs obtained by NPGM are listed in Table 6.3.

It is obvious that the control cost of network (c) is smallest among the 6 different control sources placements for elementary stems. For elementary cycles, the converged cost of network (h) is smallest. Note that the controller sources in both (c) and (h) are evenly allocated. Therefore, it coincides with the result in Chapter 3 that control cost attains its minimum when key nodes distribute evenly. Moreover, as the evolved subnetworks are similar for networks (a), (e) and (f), their control costs are very close. Same conclusions can be drawn for networks (b), (d), (g), and network (c), (h). These findings coincide with Theorem 6.3.

Based on above results, our optimization model can be applied to explain the evolution of many real-life systems, such as biological system [130], ecosystem [131] and metabolic networks [132]. It is worthy of mentioning that such networks adaptively change their interaction strengths like NPGM. Similar to above simulation examples, they eventually evolve into networks to make their weight matrices optimal.

6.8 Conclusions and Discussions

In this chapter, we address an important issue regarding how the network topologies affect the cost of controlling directed networks. We have proposed a matrix function optimization model to address how the network topology evolves for achieving more efficient control of the networks. To our best knowledge, this problem has never been addressed by existing literatures. One of the major difficulties lies in how to obtain the evolution gradient of the network topology. To this end, an *I-Chain rule* to take the derivative of an defined objective function with respect to the network topology is utilized. An algorithm named “normalized and projected gradient-decent method” (NPGM) is then proposed to solve the optimization problem. The convergence property and convergence rate have also been established. The condition that the global minimum point of the objective function is also explored and established. As having also been validated by extensive simulations results on directed networks, such a condition implies that some sub-networks of a larger network gradually form for achieving a pre-established minimum cost control target. This is also consistent with the well known fact that networks of large real-world systems reveal a number of common properties including small-world phenomena, scale-free degree distributions and modularity [46, 48, 133]. Recently, network self-similarity as a natural outcome of the evolution of real-world systems has also attracted much attention within the physics literature [123–125]. We believe that our findings provide an explanation why various real-world complex networks adaptively self-organize to many similar subnetworks during a relative long evolution history: the impetus behind such evolution may be due to the purpose of getting better control of those complex networks. It will help us to gain better understanding of optimal control of directed networks with evolutionary network topologies.

Chapter 7

Conclusions and Future Works

7.1 Conclusion

In this thesis, we have investigated the minimum-cost control problem for directed complex networks, which arises from a myriad of applications such as gene regulatory network, brain networks, traffic networks, smart grid and so on. The main results obtained are summarized as follows:

- In Chapter 3, we propose two algorithms named projected gradient method (PGM) and projected trust-region method (PTM), to address the problem of designing input matrix for a linear time-invariant system with a minimum control cost. It is shown that the PGM is of lower complexity compared with PTM. By applying PGM in different types of networks, the effectiveness of proposed algorithm is validated. In addition, projected gradient method extension (PGME) is also introduced in Monte-Carlo realizations to determine the key node set based on the definition of importance indices. It is pointed out that the importance index of a node is strongly related to its occurrence rate to be a key node. In addition, it is found that key nodes tend to divide the stem, circle and dilation equally. Moreover, the hubs with high degree are less likely to be key nodes .
- In Chapter 4, we further investigate key node selection problem and improve the corresponding results. With newly proposed algorithms, key nodes are

located via convex optimization and nodes' degree distribution. The optimization problem is formulated in Boolean constraints since key nodes are in one-to-one connection with external control resources. By relaxing these constraints into a convex hull, we apply inexact alternating direction method of multipliers (IADMM). In addition, the degree distribution is exploited to convert the result of IADMM to binary values. Thus, an extension algorithm, named Degree-based IADMM (D-IADMM) is proposed and key nodes are identified. With local optimizations, the performance of D-IADMM is greatly improved in extensive simulations from ER networks, SF networks to some real-life networks.

- In Chapter 5, we investigate implicit linear quadratic regulation problem where the input matrix B is considered as a variable. Different from the problems in Chapter 3 and Chapter 4, ImLQR provides us a flexibility to deal with the variable cost of the input energy and state-trajectory. To address this problem, we introduce orthonormal-condition-based projected gradient method (OPGM) to obtain the desired input matrix iteratively. The convergence of OPGM is guaranteed theoretically. Its validity and effectiveness are demonstrated through extensive simulations.
- The evolutions of topologies from optimal control perspective are investigated in Chapter 6. The algorithm named normalized and projected gradient-decent method (NPGM) is proposed with established convergence property and convergence rate. The optimality condition is also derived to validate whether the obtained solution by NPGM is optimal or not. It is shown that several similar subsystems are self-formed with the proposed algorithm when the optimal solution is obtained. Particularly, we extend NPGM to explore how the connection strengths between nodes affect the control cost on a fixed structural network. It is shown that, when the optimal weight matrix is obtained, several control-flow networks are formed. In addition, key nodes that are just equally distributed in a network is of minimum control cost.

7.2 Recommendations for Future Works

The study of complex networks is a challenging topic due to its interdisciplinarity and complexity, although enormous research works are emerging in recent years.

Apart from the results achieved in this thesis and literature, there are still some interesting problems worthy of further study.

7.2.1 Sparsity-promoting topology design for directed complex networks

In Chapter 6, the topology obtained by NPGM may compose of many connection links, resulting in a high cost for real implementations. For example, it may cause expensive communication in smart grid and sensor networks due to dense connections [134–136]. On the other hand, many real-life networks are of sparsity nature. Motivated by this, a sparsity-promoting topology can be designed based on optimal control theory. Combining the case that connection strength is nonnegative, the optimization model can be reformulated as

$$\begin{aligned} \arg \min_A \quad & \text{tr} \left(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f} \right) \\ \text{s.t.} \quad & |A|_{\ell_1} \leq \lambda, \quad A \geq 0, \quad \mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \quad (7.1)$$

where $|A|_{\ell_1} = \sum_{ij} |a_{ij}|$ is ℓ_1 norm, and $A \geq 0$ indicates A is a nonnegative matrix. It is expected that projected gradient method can be developed to solve above problem.

7.2.2 Closed-loop optimal control of complex networks

In this thesis, we study controlling of directed complex networks with minimum control cost under open-loop scheme. As it is known, feedback control can make the system more robust to uncertainties and exotic disturbances [137]. Therefore, it is worthwhile to investigate closed-loop optimal control of complex networks.

A feedback control is injected to the system (2.1), that is,

$$\mathbf{u}(t) = -K\mathbf{x}$$

and the objective function is

$$\mathcal{J} = \int_0^{t_f} \alpha \|\mathbf{u}(t)\|^2 + \beta \|\mathbf{x}(t)\|^2 dt$$

where the term $\int_0^{t_f} \|\mathbf{x}(t)\|^2 dt$ penalizes the deviation of transient state. Similar to the problems in this thesis, we can develop the optimal control strategy and input matrix B with known topology A , or topology A with given input matrix B , respectively.

7.2.3 Alternative control of complex networks

In Chapters 3-5, we address the problem of controlling of directed complex networks with known topology A . In Chapter 6, the role of the topology A is explored with input matrix B pre-given. What if neither A nor B is given? It is interesting to explore such a case from the optimal control perspective. To address this issue, an idea of alternative optimization is employed. Specifically, with initial A_0 and B_0 , we will optimize A by gradient method with B fixed and update B with A fixed alternatively. Now this work is under way.

7.2.4 Distributed optimal control of large-scale complex networks

Distributed optimization has been an active research area in a variety of applications in virtue of its ability to deal with large-scale networks [91, 138]. To control large-scale complex networks, distributed methods make it feasible with low computational cost. Consider a network composed by l subnetworks. Each subnetwork is described by N_i -dimensional linear dynamics

$$\dot{\mathbf{x}}_i(t) = A_i \mathbf{x}_i(t) + B_i \mathbf{u}_i(t), \quad \mathbf{x}_i(0) = \mathbf{x}_0, \quad (7.2)$$

The objective is to choose a subset of nodes of the whole network such that the input energy $\int_0^{t_f} \|\mathbf{u}_1(t), \mathbf{u}_2(t), \dots, \mathbf{u}_l(t)\|^2 dt$ is minimized.

Appendix A

Proofs for Chapter 3

A.1 Proof of Lemma 3.1

The proof is obvious since $\frac{\partial \mathcal{N}(B)}{\partial B} = 2B$.

A.2 Proof of Lemma 3.2

From Lemma 2.1 , we have

$$\begin{aligned}
& \frac{\partial \mathcal{E}(B)}{\partial B_{ij}} \\
&= \frac{\partial \text{tr}[W_B^{-1} e^{At_f} e^{A^T t_f}]}{\partial B_{ij}} \\
&= \frac{\partial \text{tr}[W_B^{-1} e^{At_f} e^{A^T t_f}]}{\partial [W_B]_{kl}} \cdot \frac{\partial [W_B]_{kl}}{\partial B_{ij}} \\
&= -[W_B^{-T} e^{At_f} e^{A^T t_f} W_B^{-T}]_{kl} \cdot \frac{\partial [\int_0^{t_f} e^{At} B B^T e^{A^T t} dt]_{kl}}{\partial B_{ij}} \\
&= -[W_B^{-T} e^{At_f} e^{A^T t_f} W_B^{-T}]_{kl} \cdot \frac{\partial [\int_0^{t_f} [e^{At}]_{km} \cdot [B]_{mn} \cdot [B^T]_{nz} \cdot [e^{A^T t}]_{zl} dt]}{\partial B_{ij}} \\
&= -[W_B^{-T} e^{At_f} e^{A^T t_f} W_B^{-T}]_{kl} \cdot \left[\int_0^{t_f} [e^{At}]_{km} \cdot [\delta_{mi} \cdot \delta_{nj} \cdot [B^T]_{nz} - [B]_{mn} \cdot \delta_{nj} \cdot \delta_{zi}] \cdot [e^{A^T t}]_{zl} dt \right] \\
&= -[W_B^{-T} e^{At_f} e^{A^T t_f} W_B^{-T}]_{kl} \cdot \left[\int_0^{t_f} [e^{At}]_{ki} \cdot [B^T]_{jz} \cdot [e^{A^T t}]_{zl} dt - [W_B^{-T} e^{At_f} e^{A^T t_f} W_B^{-T}]_{kl} \cdot \int_0^{t_f} [e^{At}]_{km} \cdot [B]_{mj} \cdot [e^{A^T t}]_{il} dt \right] \\
&= -2 \cdot \int_0^{t_f} [e^{A^T t}]_{ik} \cdot [W_B^{-T} e^{At_f} e^{A^T t_f} W_B^{-T}]_{kl} \cdot [e^{At}]_{lz} \cdot B_{zj} dt \\
&= -2 \left[\int_0^{t_f} e^{A^T t} W_B^{-T} e^{At_f} e^{A^T t_f} W_B^{-T} e^{At} dt B \right]_{ij}
\end{aligned} \tag{A.1}$$

where i, j, k, l, m, n, z are indexes of elements in the matrix. Thus the lemma is proved. $\square$

A.3 Proof of Lemma 3.3

Note that

$$\begin{aligned}
\frac{\partial[W_B^{-1}]_{mn}}{\partial B_{kl}^T} &= \frac{\partial[W_B^{-1}]_{mn}}{\partial[W_B]_{pq}} \cdot \frac{\partial[W_B]_{pq}}{\partial B_{lk}} \\
&= -[W_B^{-1}]_{mp} \cdot [W_B^{-1}]_{qn} \cdot \int_0^{t_f} \frac{[e^{At}]_{ps} \cdot [B]_{su} \cdot [B^T]_{uv} \cdot [e^{A^T t}]_{vq}}{B_{lk}} dt \\
&= -[W_B^{-1}]_{mp} \cdot [W_B^{-1}]_{qn} \cdot \int_0^{t_f} \left[[e^{At}]_{ps} \cdot \delta_{sl} \cdot \delta_{uk} \cdot [B^T]_{uv} \cdot [e^{A^T t}]_{vq} + [e^{At}]_{ps} \cdot [B]_{su} \cdot \delta_{uk} \cdot \delta_{vl} \cdot [e^{A^T t}]_{vq} \right] dt \\
&= -[W_B^{-1}]_{mp} \cdot [W_B^{-1}]_{qn} \cdot \int_0^{t_f} \left[[e^{At}]_{pl} \cdot [B^T]_{kv} \cdot [e^{A^T t}]_{vq} + [e^{At}]_{ps} \cdot [B]_{sk} \cdot [e^{A^T t}]_{lq} \right] dt \\
&= - \int_0^{t_f} \left[[W_B^{-1} e^{At}]_{ml} \cdot [B^T e^{A^T t} W_B^{-1}]_{kn} + [W_B^{-1} e^{At} B]_{mk} \cdot [e^{A^T t} W_B^{-1}]_{ln} \right] dt
\end{aligned} \tag{A.2}$$

Then, we have

$$\begin{aligned}
&\frac{\partial \left[\frac{\partial \mathcal{E}(B)}{\partial B} \right]_{ij}}{\partial B_{kl}^T} \\
&= -2 \int_0^{t_f} \frac{\partial \left[[e^{A^T t}]_{im} \cdot [W_B^{-1}]_{mn} \cdot [e^{At_f} e^{A^T t_f}]_{nz} \cdot [W_B^{-1}]_{zr} \cdot [e^{At}]_{rh} \cdot [B]_{hj} \right]}{\partial B_{lk}} dt \\
&= -2 \int_0^{t_f} [e^{A^T t}]_{im} \cdot \frac{\partial[W_B^{-1}]_{mn}}{\partial B_{lk}} \cdot [e^{At_f} e^{A^T t_f} W_B^{-1} e^{At} B]_{nj} dt \\
&\quad - 2 \int_0^{t_f} [e^{A^T t} W_B^{-1} e^{At_f} e^{A^T t_f}]_{iz} \cdot \frac{\partial[W_B^{-1}]_{zr}}{\partial B_{lk}} \cdot [e^{At} B]_{rj} dt \\
&\quad - 2 \int_0^{t_f} [e^{A^T t} W_B^{-1} e^{At_f} e^{A^T t_f} W_B^{-1} e^{At}]_{ih} \cdot \frac{\partial[B]_{hj}}{\partial B_{lk}} dt \\
&= 2 \int_0^{t_f} [e^{A^T t}]_{im} \cdot \left[\int_0^{t_f} [[W_B^{-1} e^{At}]_{ml} \cdot [B^T e^{A^T t} W_B^{-1}]_{kn} + [W_B^{-1} e^{At} B]_{mk} \cdot [e^{A^T t} W_B^{-1}]_{ln}] dt \right] \cdot [e^{At_f} e^{A^T t_f} W_B^{-1} e^{At} B]_{nj} dt \\
&\quad + 2 \int_0^{t_f} [e^{A^T t} W_B^{-1} e^{At_f} e^{A^T t_f}]_{iz} \cdot \left[\int_0^{t_f} [[W_B^{-1} e^{At}]_{zl} \cdot [B^T e^{A^T t} W_B^{-1}]_{kr} + [W_B^{-1} e^{At} B]_{zk} \cdot [e^{A^T t} W_B^{-1}]_{lr}] dt \right] \cdot [e^{At} B]_{rj} dt \\
&\quad - 2 \int_0^{t_f} [e^{A^T t} W_B^{-1} e^{At_f} e^{A^T t_f} W_B^{-1} e^{At}]_{il} \cdot \delta_{jk} dt
\end{aligned} \tag{A.3}$$

Based on Lemma 3.2 and formula (2.16), the Hessian matrix of $E(\{B\})$ can be obtained.

A.4 Proof of Lemma 3.4

From Lemma 3.1, it is known that the gradient of norm function $\mathcal{N}(B)$ is parallel with B . For an arbitrary vector v , as

$$\begin{aligned} \{B\}^T \mathcal{P}_{\frac{\partial \mathcal{N}(B)}{\partial B}}(v) &= \{B\}^T (I - \{B\}\{B\}^\dagger) v \\ &= \{B\}^T \left(I - \{B\} (\{B\}^T \{B\})^{-1} \{B\}^T \right) v \\ &= (\{B\}^T - \{B\}^T) v \\ &= 0 \end{aligned}$$

So this lemma holds.

A.5 Proof of Theorem 3.1

Let matrix $C = e^{A^T t} W_B^{-1} e^{At_f}$. As (A, B) is controllable, matrices $e^{A^T t}$ and W_B^{-1} are invertible. Hence we have

$$\int_0^{t_f} e^{A^T t} W_B^{-1} e^{At_f} e^{A^T t_f} W_B^{-1} e^{At} dt = \int_0^{t_f} C^T C dt > 0$$

where $P > 0$ means that the matrix is positive definite. On the other hand, since (A, B) is controllable, $B \neq 0$. Thus $\frac{\partial \mathcal{E}(B)}{\partial B} = -2 \int_0^{t_f} C^T C \cdot B dt$ can never be a zero matrix as $\int_0^{t_f} C^T C dt$ is positive definite.

A.6 Proof of Theorem 3.2

Combining the steps in PGM, we have

$$\{B_{k+1}\} = \Gamma \left[\{B_k\} - \eta \cdot \left(I - \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\} \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} \right] \quad (\text{A.4})$$

To drag $\{B_{k+1}\}$ onto the surface $\text{tr}(B^T B) = M$, we introduce a scalar ϱ_k in the iteration:

$$\begin{aligned}\{B_{k+1}\} &= \varrho_k \left[\{B_k\} - \eta \cdot \left(I - \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\} \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} \right] \\ &= \varrho_k \{B_k\} - \eta \varrho_k \cdot \left(I - \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\} \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} \\ &= \{B_k\} + (\varrho_k - 1) \{B_k\} - \eta \varrho_k \cdot \left(I - \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\} \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\}^\dagger \right) \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\}\end{aligned}$$

such that $\{B_{k+1}\}^T \{B_{k+1}\} = \{B_k\}^T \{B_k\} = \text{tr}(B_{k+1}^T B_{k+1}) = \text{tr}(B_k^T B_k) = M$. Denoting

$$P_B = \left(I - \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\} \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\}^\dagger \right) \quad (\text{A.5})$$

we have

$$\{B_{k+1}\} = \{B_k\} + (\varrho_k - 1) \{B_k\} - \eta \varrho_k \cdot P_B \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} \quad (\text{A.6})$$

and

$$\text{tr}(B_{k+1}^T B_{k+1}) = \varrho_k^2 \left(\text{tr}(B_k^T B_k) + \eta^2 \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\}^T P_B^T P_B \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} \right) = \{B_k\}^T \{B_k\} \quad (\text{A.7})$$

By letting $\mathbf{e} = P_B \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\}$, then obtain

$$\varrho_k = \sqrt{\frac{\text{tr}(B_k^T B_k)}{\left(\text{tr}(B_k^T B_k) + \eta^2 \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\}^T P_B^T P_B \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} \right)}} = \sqrt{\frac{M}{M + \eta^2 \mathbf{e}^T \mathbf{e}}} = 1 + \frac{1}{2} \eta^2 \frac{\mathbf{e}^T \mathbf{e}}{M} + o(\eta^2) \quad (\text{A.8})$$

So $\varrho_k - 1$ is infinitesimal of higher order of η , which is denoted as

$$\varrho_k - 1 \sim o(\eta) \quad (\text{A.9})$$

Finally, it is obtained that

$$\begin{aligned}
\mathcal{E}(B_{k+1}) &= \mathcal{E}(B_k) + \left((\varrho_k - 1)\{B_k\} - \eta\varrho_k \cdot P_B \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} \right)^T \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \\
&= \mathcal{E}(B_k) + o(\eta)\{B_k\}^T \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\} - \eta\varrho_k \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}^T P_B \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\} \\
&= \mathcal{E}(B_k) - \eta\varrho_k \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}^T P_B \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\} \\
&= \mathcal{E}(B_k) - \eta\varrho_k \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}^T P_B^2 \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}
\end{aligned} \tag{A.10}$$

$$\leq 0$$

The iterations end when $\left\{ -\frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}$ concurs with the direction of $\left\{ \frac{\partial \mathcal{N}(B_k)}{\partial B_k} \right\}$, i.e.,

$$\left(I - \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\} \left\{ \frac{\partial \mathcal{N}(B)}{\partial B} \right\}^+ \right) \left\{ \frac{\partial \mathcal{E}(B)}{\partial B} \right\} = 0 \tag{A.11}$$

Thus, $\cos(\theta_k)$ converges to -1 . Thus, this theorem holds. $\square$

A.7 Proof of Theorem 3.3

Let $\varrho'_k = \varrho_k \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}^T P_B^2 \left\{ \frac{\partial \mathcal{E}(B_k)}{\partial B_k} \right\}$. Then we have

$$\mathcal{E}(B_k) = \mathcal{E}(B_{k+1}) + \eta\varrho'_k \tag{A.12}$$

Minus $\mathcal{E}(B^*)$ on both sides, we have

$$\mathcal{E}(B_k) - \mathcal{E}(B^*) = \mathcal{E}(B_{k+1}) - \mathcal{E}(B^*) + \eta\varrho'_k \tag{A.13}$$

which gives

$$\frac{\mathcal{E}(B_{k+1}) - \mathcal{E}(B^*)}{\mathcal{E}(B_k) - \mathcal{E}(B^*)} = 1 - \frac{\eta\varrho'_k}{\mathcal{E}(B_k) - \mathcal{E}(B^*)} \tag{A.14}$$

As $k \rightarrow +\infty$, we have $\lim_{k \rightarrow +\infty} \frac{\mathcal{E}(B_{k+1}) - \mathcal{E}(A^*)}{\mathcal{E}(A_k) - \mathcal{E}(A^*)} = 1$, indicating that algorithm converges linearly.

Appendix B

Proofs for Chapter 5

B.1 Proof of Lemma 5.1

Please refer to the proof of Lemma 3.2.

B.2 Proof of Lemma 5.2

By applying the chain rule for deriving the differentiation of a matrix-trace-based function, we have

$$\begin{aligned} & \frac{\partial \int_0^{t_f} \mathbf{x}^T(t, B) \mathbf{x}(t, B) dt}{\partial B_{ij}} \\ &= 2 \int_0^{t_f} [\mathbf{x}^T(t, B)]_{1k} \cdot \frac{\partial [\mathbf{x}(t, B)]_{k1}}{\partial B_{ij}} dt \\ &= 2 \int_0^{t_f} [\mathbf{x}^T(t, B)]_{1k} \cdot \frac{\partial [\mathbf{x}(t, B)]_{k1}}{\partial B_{ij}} dt \\ &= 2 \int_0^{t_f} [\mathbf{x}^T(t, B)]_{1k} \cdot \frac{\partial [e^{At} x_0 + \int_0^t e^{A(t-\tau)} B \mathbf{u}(\tau) d\tau]_{k1}}{\partial B_{ij}} dt \\ &= -2 \int_0^{t_f} [\mathbf{x}^T(t, B)]_{1k} \cdot \frac{\partial [\int_0^t e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)} W_B^{-1} e^{At_f} \mathbf{x}_0 d\tau]_{k1}}{\partial B_{ij}} dt \end{aligned} \tag{B.1}$$

Then, we have

$$\begin{aligned}
& \frac{\partial \int_0^{t_f} \mathbf{x}^T(t, B) \mathbf{x}(t, B) dt}{\partial B_{ij}} \\
&= -2 \int_0^{t_f} \int_0^t \partial \left([\mathbf{x}^T(t, B) e^{A(t-\tau)}]_{1a} \cdot [B]_{ab} \cdot [B^T]_{bc} \cdot [e^{A^T(t_f-\tau)} W_B^{-1} e^{At_f} \mathbf{x}_0]_{c1} \right) / \partial B_{ij} d\tau dt \\
&= -2 \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B)]_{1k} \cdot [e^{A(t-\tau)}]_{ka} \cdot \frac{\partial B_{ab}}{\partial B_{ij}} \cdot [B^T e^{A^T(t_f-\tau)} W_B^{-1} e^{At_f} \mathbf{x}_0]_{b1} d\tau dt \\
&\quad - 2 \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B]_{1b} \cdot \frac{\partial B_{bc}^T}{\partial B_{ij}} \cdot [e^{A^T(t_f-\tau)} W_B^{-1} e^{At_f} \mathbf{x}_0]_{c1} d\tau dt \\
&\quad - 2 \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)}]_{1f} \cdot [e^{At_f} \mathbf{x}_0]_{g1} d\tau dt \cdot \frac{\partial [W_B^{-1}]_{fg}}{\partial B_{ij}} \\
&= -2 \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B)]_{1k} \cdot [e^{A(t-\tau)}]_{ka} \cdot \delta_{ai} \cdot \delta_{bj} \cdot [B^T e^{A^T(t_f-\tau)} W_B^{-1} e^{At_f} \mathbf{x}_0]_{b1} d\tau dt \\
&\quad - 2 \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B]_{1b} \cdot \delta_{ic} \cdot \delta_{bj} \cdot [e^{A^T(t_f-\tau)} W_B^{-1} e^{At_f} \mathbf{x}_0]_{c1} d\tau dt \\
&\quad - 2 \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)}]_{1f} \cdot [e^{At_f} \mathbf{x}_0]_{g1} d\tau dt \cdot \frac{\partial [W_B^{-1}]_{fg}}{\partial [W_B]_{mn}} \cdot \frac{\partial [W_B]_{mn}}{\partial B_{ij}} \\
&\hspace{15em} (B.2)
\end{aligned}$$

For the last part of (B.2), by applying the commutative principle for multiplication, we have

$$\begin{aligned}
& - \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)}]_{1f} \cdot [e^{At_f} \mathbf{x}_0]_{g1} d\tau dt \cdot \frac{\partial [W_B^{-1}]_{fg}}{\partial [W_B]_{mn}} \cdot \frac{\partial [W_B]_{mn}}{\partial B_{ij}} \\
&= \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)}]_{1f} \cdot [e^{At_f} \mathbf{x}_0]_{g1} d\tau dt \cdot [W_B^{-1}]_{fm} \cdot [W_B^{-1}]_{ng} \cdot \int_0^{t_f} \frac{\partial \left([e^{A\sigma}]_{mo} \cdot [B]_{op} \cdot [B^T]_{pq} \cdot [e^{A^T\sigma}]_{qn} \right)}{\partial B_{ij}} d\sigma \\
&= \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)}]_{1f} \cdot [e^{At_f} \mathbf{x}_0]_{g1} d\tau dt \cdot [W_B^{-1}]_{fm} \cdot [W_B^{-1}]_{ng} \cdot \int_0^{t_f} [e^{A\sigma}]_{mo} \cdot \delta_{oi} \cdot \delta_{pj} \cdot [B^T]_{pq} \cdot [e^{A^T\sigma}]_{qn} d\sigma \\
&\quad + \int_0^{t_f} \int_0^t [\mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)}]_{1f} \cdot [e^{At_f} \mathbf{x}_0]_{g1} d\tau dt \cdot [W_B^{-1}]_{fm} \cdot [W_B^{-1}]_{ng} \cdot \int_0^{t_f} [e^{A\sigma}]_{mo} \cdot [B]_{op} \cdot \delta_{pj} \cdot \delta_{qi} \cdot [e^{A^T\sigma}]_{qn} d\sigma \\
&= \left[\int_0^{t_f} e^{A^T\sigma} W_B^{-T} \int_0^{t_f} \int_0^t e^{A(t_f-\tau)} B B^T e^{A^T(t-\tau)} \mathbf{x}(t, B) \mathbf{x}_0^T e^{At_f} d\tau dt W_B^{-T} e^{A\sigma} B d\sigma \right]_{ij} \\
&\quad + \left[\int_0^{t_f} e^{A^T\sigma} W_B^{-1} \int_0^{t_f} \int_0^t e^{At_f} \mathbf{x}_0 \mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)} d\tau dt W_B^{-1} e^{A\sigma} B d\sigma \right]_{ij} \\
&\hspace{15em} (B.3)
\end{aligned}$$

By substituting (B.3) into (B.2), we finally obtain

$$\begin{aligned}
& \frac{\partial \int_0^{t_f} \mathbf{x}^T(t, B) \mathbf{x}(t, B) dt}{\partial B_{ij}} \\
&= -2 \left[\int_0^{t_f} \int_0^t e^{A^T(t-\tau)} \mathbf{x}(t, B) \mathbf{x}_0^T e^{A^T t_f} W_B^{-T} e^{A(t_f-\tau)} B d\tau dt \right]_{ij} \\
&\quad - 2 \left[\int_0^{t_f} \int_0^t e^{A^T(t_f-\tau)} W_B^{-1} e^{A t_f} \mathbf{x}_0 \mathbf{x}^T(t, B) e^{A(t-\tau)} B d\tau dt \right]_{ij} \\
&\quad + 2 \left[\int_0^{t_f} e^{A^T \sigma} W_B^{-T} \int_0^{t_f} \int_0^t e^{A(t_f-\tau)} B B^T e^{A^T(t-\tau)} \mathbf{x}(t, B) \mathbf{x}_0^T e^{A^T t_f} d\tau dt W_B^{-T} e^{A \sigma} B d\sigma \right]_{ij} \\
&\quad + 2 \left[\int_0^{t_f} e^{A^T \sigma} W_B^{-1} \int_0^{t_f} \int_0^t e^{A t_f} \mathbf{x}_0 \mathbf{x}^T(t, B) e^{A(t-\tau)} B B^T e^{A^T(t_f-\tau)} d\tau dt W_B^{-1} e^{A \sigma} B d\sigma \right]_{ij}
\end{aligned} \tag{B.4}$$

where

$$\begin{aligned}
\mathbf{x}_0 \mathbf{x}^T(t, B) &= \mathbf{x}_0 \mathbf{x}_0^T \left[e^{A^T t} - e^{A^T t_f} W_B^{-T} \int_0^t e^{A(t_f-\nu)} B B^T e^{A^T(t-\nu)} d\nu \right] \\
\mathbf{x}(t, B) \mathbf{x}_0^T &= \left[e^{A t} - \int_0^t e^{A(t-\nu)} B B^T e^{A^T(t_f-\nu)} W_B^{-1} e^{A t_f} d\nu \right] \mathbf{x}_0 \mathbf{x}_0^T
\end{aligned} \tag{B.5}$$

Thus, this lemma holds. $\square$

B.3 Proof of Theorem 5.1

We use $\mathcal{T}_{B_k}$, $\nabla \mathcal{N}(B_k)$ and $\nabla E(B)$ to represent $(I_N - B_k B_k^T)$, $\frac{\partial \mathcal{N}(B_k)}{\partial B_k}$ and $\frac{\partial E(B_k)}{\partial B_k}$ respectively. Our method is to minimize $E(B)$ and $\mathcal{N}(B)$ simultaneously. To prove the convergence, we have to prove two parts as follows:

$$E(B_{k+1}) - E(B_k) \leq 0, \text{ and } E(B_k) \geq 0, \tag{B.6}$$

$$\mathcal{N}(B_{k+1}) - \mathcal{N}(B_k) \leq 0, \text{ and } \mathcal{N}(B_k) \geq 0 \tag{B.7}$$

From Lemma 5.7 and (5.15), we have

$$0 < \text{tr}(B_{k+1}^T B_{k+1}) = \frac{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})} \cdot \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1}) \tag{B.8}$$

for $k = 0, 1, 2, \dots$, and

$$\begin{aligned}
 \text{tr}(B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1}) &= \frac{\text{tr}^2(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}^2(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})} \cdot \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1}) \\
 &= \frac{\text{tr}^2(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})} \\
 &= \text{tr}(B_{k+1}^T B_{k+1}) > 0
 \end{aligned} \tag{B.9}$$

For $\mathcal{N}(B_{k+1})$, from (B.8) and (B.9), we have

$$\begin{aligned}
 \mathcal{N}(B_{k+1}) &= \text{tr}(B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1}) - 2 \text{tr}(B_{k+1}^T B_{k+1}) + M \\
 &= M - \text{tr}(B_{k+1}^T B_{k+1}) \geq 0.
 \end{aligned} \tag{B.10}$$

Now we consider $\mathcal{N}(\hat{B}_{k+1}) - \mathcal{N}(B_k)$. As $\mathcal{N}(B_k) = 4(B_k B_k^T - I_N)B_k$ and $B_k B_k^T$ is symmetric, from Lemma 5.8, and by using Taylor expansion to expand $\mathcal{N}(B)$ [127], we have

$$\begin{aligned}
 \mathcal{N}(\hat{B}_{k+1}) &= \mathcal{N}(B_k + \eta \Delta B_k) \\
 &= \mathcal{N}(B_k) + \eta \text{tr}(\nabla \mathcal{N}(B_k)^T \cdot \Delta B_k) \\
 &= \mathcal{N}(B_k) - 4\eta \cdot \text{tr}\left(B_k^T (I_N - B_k B_k^T)(I_N - B_k B_k^T) \widetilde{F(B_k)} B_k\right)
 \end{aligned} \tag{B.11}$$

and (B.11) can be written as

$$\mathcal{N}(\hat{B}_{k+1}) - \mathcal{N}(B_k) = -4\eta \cdot \text{tr}\left(B_k^T (I_N - B_k B_k^T)(I_N - B_k B_k^T) \widetilde{F(B_k)} B_k\right) \leq 0 \tag{B.12}$$

due to the positive definiteness of $\widetilde{F(B_k)}$. Then, we have $\mathcal{N}(\hat{B}_{k+1}) - \mathcal{N}(B_k) \leq 0$ due to the positive definite of $\widetilde{F(B_k)}$.

As to $\mathcal{N}(B_{k+1}) - \mathcal{N}(\hat{B}_{k+1})$, we have

$$\begin{aligned}
\mathcal{N}(B_{k+1}) &= \mathcal{N}\left(\sqrt{\frac{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})}} \cdot \hat{B}_{k+1}\right) \\
&= \frac{\text{tr}^2(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}^2(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})} \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1}) \\
&\quad - 2 \frac{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})} \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1}) + M \\
&= M - \frac{\text{tr}^2(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})}.
\end{aligned} \tag{B.13}$$

Thus,

$$\begin{aligned}
&\mathcal{N}(B_{k+1}) - \mathcal{N}(\hat{B}_{k+1}) \\
&= M - \frac{\text{tr}^2(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})} - \left(\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1}) - 2 \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1}) + M\right) \\
&= - \frac{(\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1}) - \text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1}))^2}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})} \leq 0
\end{aligned} \tag{B.14}$$

From (B.12) and (B.14), we have

$$\mathcal{N}(B_{k+1}) - \mathcal{N}(B_k) = \mathcal{N}(B_{k+1}) - \mathcal{N}(\hat{B}_{k+1}) + \mathcal{N}(\hat{B}_{k+1}) - \mathcal{N}(B_k) \leq 0. \tag{B.15}$$

From (B.10) and (B.15), the convergence of $\mathcal{N}(B)$ has been proven.

Now we prove that $\mathcal{N}(B)$ converges to zero when we obtain a full column rank matrix B^* . From Lemma 5.5, we have $\mathcal{N}(B^*) = \sum_i (\Lambda_i^* - 1)^2$ where Λ_i^* is an eigenvalue of $B^{*T} B^*$. Then, it is obtained that $\Lambda_i^* > 0$. Otherwise, $\mathcal{N}(B^*) \geq 1$ which is contradictory with Assumption 5.1 since $\mathcal{N}(B_k)$ is non-increasing. Thus, from Assumption 5.1 and Lemma 5.5, all eigenvalues of $B^{*T} B^*$ cannot be zero and $\forall \epsilon > 0, \exists K \in \mathbb{N}, \forall k > K, |\mathcal{N}(B_k)| < \epsilon$, which can also be written as $\mathcal{N}(B_k) = o(\eta)$ as $k \rightarrow \infty$ and $\eta \rightarrow 0$ where $o(\eta)$ is a remainder of the Taylors expansion representing a polynomial with its order not smaller than η . Thus, we obtain that

$$\lim_{k \rightarrow \infty, \eta \rightarrow 0} B_k^T B_k = I_M \tag{B.16}$$

To prove $E(B_{k+1}) - E(B_k) \leq 0$, we first use B_k to express B_{k+1} directly. As $\eta \cdot \Delta B_k = \eta \cdot \mathcal{T}_{B_k} \cdot \nabla E(B_k)$, we have

$$\begin{aligned}
B_{k+1} &= \sqrt{\frac{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1})}{\text{tr}(\hat{B}_{k+1}^T \hat{B}_{k+1} \hat{B}_{k+1}^T \hat{B}_{k+1})}} \cdot \hat{B}_{k+1} \\
&= \sqrt{\frac{\text{tr}((B_k + \eta \Delta B_k)^T (B_k - \eta \Delta B_k))}{\text{tr}(((B_k + \eta \Delta B_k)^T (B_k + \eta \Delta B_k))^2)}} (B_k + \eta \cdot \Delta B_k) \\
&= (\text{tr}(B_k^T B_k + \eta \Delta B_k^T B_k + \eta B_k^T \Delta B_k + \eta^2 \Delta B_k^T \Delta B_k))^{\frac{1}{2}} \\
&\quad \cdot \left(\text{tr}(B_k^T B_k B_k^T B_k) + 4\eta \text{tr}(\Delta B_k^T B_k B_k^T B_k) + 6\eta^2 \text{tr}(\Delta B_k^T \Delta B_k B_k^T B_k) \right. \\
&\quad \left. + 4\eta^3 \text{tr}(\Delta B_k^T \Delta B_k \Delta B_k^T B_k) + \eta^4 \text{tr}(\Delta B_k^T \Delta B_k \Delta B_k^T \Delta B_k) \right)^{-\frac{1}{2}} \cdot (B_k + \eta \cdot \Delta B_k)
\end{aligned} \tag{B.17}$$

Then, by Lemma 5.6, the above formula (B.17) can be written as

$$\begin{aligned}
B_{k+1} &= \text{tr}^{\frac{1}{2}}(B_k^T B_k) \left(1 + \eta \frac{\text{tr}(\Delta B_k^T B_k)}{\text{tr}(B_k^T B_k)} + o(\eta) \right) \\
&\quad \cdot \text{tr}^{-\frac{1}{2}}(B_k^T B_k B_k^T B_k) \left(1 - 2\eta \frac{\text{tr}(\Delta B_k^T B_k B_k^T B_k)}{\text{tr}(B_k^T B_k B_k^T B_k)} + o(\eta) \right) (B_k + \eta \cdot \Delta B_k) \\
&= \sqrt{\frac{\text{tr}(B_k^T B_k)}{\text{tr}(B_k^T B_k B_k^T B_k)}} \left(1 + \eta \frac{\text{tr}(\Delta B_k^T B_k)}{\text{tr}(B_k^T B_k)} - 2\eta \frac{\text{tr}(\Delta B_k^T B_k B_k^T B_k)}{\text{tr}(B_k^T B_k B_k^T B_k)} + o(\eta) \right) (B_k + \eta \cdot \Delta B_k)
\end{aligned} \tag{B.18}$$

By using (B.9), we have $\sqrt{\frac{\text{tr}(B_k^T B_k)}{\text{tr}(B_k^T B_k B_k^T B_k)}} = 1$. Then, we apply Taylor expansion to expand $E(B)$ [127] when $\eta \rightarrow 0$ by substituting $\Delta B_k = -\mathcal{T}_{B_k} \cdot \nabla E(B_k)$ and $\nabla E(B_k) = -\widehat{F(B_k)} \cdot B_k$, we obtain

$$\begin{aligned}
&E(B_{k+2}) \\
&= E\left(\left(1 + \eta \frac{\text{tr}(\Delta B_{k+1}^T B_{k+1})}{\text{tr}(B_{k+1}^T B_{k+1})} - 2\eta \frac{\text{tr}(\Delta B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1})}{\text{tr}(B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1})} + o(\eta) \right) \cdot (B_{k+1} + \eta \cdot \Delta B_{k+1}) \right) \\
&= E(B_{k+1}) + \eta \left(\frac{\text{tr}(\Delta B_{k+1}^T B_{k+1})}{\text{tr}(B_{k+1}^T B_{k+1})} - 2 \frac{\text{tr}(\Delta B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1})}{\text{tr}(B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1})} \right) \cdot \text{tr}(B_{k+1}^T [F(B_{k+1}) + \lambda_k I]^T B_{k+1}) \\
&\quad - \eta \text{tr}(B_{k+1}^T F(B_{k+1})^T \mathcal{T}_{B_{k+1}} [F(B_{k+1}) + \lambda_k I] B_{k+1}) + o(\eta)
\end{aligned} \tag{B.19}$$

From (B.16), we have $\exists K \in \mathbb{N}$, $\forall k > K$, $\mathcal{T}_{B_{k+1}} B_{k+1} = B_{k+1} - B_{k+1} B_{k+1}^T B_{k+1} = o(\eta)$. Thus,

$$\begin{aligned} & \text{tr}(B_{k+1}^T F(B_{k+1})^T \mathcal{T}_{B_{k+1}} [F(B_{k+1}) + \lambda_k I]^T B_{k+1}) \\ &= \text{tr}(B_{k+1}^T F(B_{k+1})^T \mathcal{T}_{B_{k+1}} [F(B_{k+1})]^T B_{k+1}) + o(\eta) \end{aligned} \quad (\text{B.20})$$

Note that $\mathcal{T}_{B_{k+1}}^2 = (I_N - B_{k+1} B_{k+1}^T)^2 = \mathcal{T}_{B_{k+1}} + o(\eta)$. Then,

$$\eta \text{tr}(B_{k+1}^T F(B_{k+1})^T \mathcal{T}_{B_{k+1}} F(B_{k+1}) B_{k+1}) = \eta \text{tr} \left(B_{k+1}^T F(B_{k+1})^T \mathcal{T}_{B_{k+1}}^2 F(B_{k+1}) B_{k+1} \right) + o(\eta). \quad (\text{B.21})$$

Likewise, we have $\text{tr}(\Delta B_{k+1}^T B_{k+1}) = o(\eta)$ and $\text{tr}(\Delta B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1}) = o(\eta)$ when $\eta \rightarrow 0$. Thus, we have

$$\eta \left(\frac{\text{tr}(\Delta B_{k+1}^T B_{k+1})}{\text{tr}(B_{k+1}^T B_{k+1})} - 2 \frac{\text{tr}(\Delta B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1})}{\text{tr}(B_{k+1}^T B_{k+1} B_{k+1}^T B_{k+1})} \right) \cdot \text{tr}(B_{k+1}^T [\widetilde{F(B_{k+1})}]^T B_{k+1}) = o(\eta). \quad (\text{B.22})$$

With (B.21) and (B.22), (B.19) can be written as $\exists K \in \mathbb{N}$, $\forall k > K$,

$$E(B_{k+2}) - E(B_{k+1}) = -\eta \text{tr} \left(B_{k+1}^T F(B_{k+1})^T \mathcal{T}_{B_{k+1}}^2 F(B_{k+1}) B_{k+1} \right) + o(\eta) \quad (\text{B.23})$$

as $B_{k+1}^T F(B_{k+1})^T \mathcal{T}_{B_{k+1}}^2 F(B_{k+1}) B_{k+1} = (\mathcal{T}_{B_{k+1}} F(B_{k+1}) B_{k+1})^T (\mathcal{T}_{B_{k+1}} F(B_{k+1}) B_{k+1})$ is a nonnegative definite matrix and all its eigenvalues can be denoted as $\lambda_1 \geq 0, \dots, \lambda_i \geq 0, \dots, \lambda_M \geq 0$. Let $\lambda^* = \sum_i \lambda_i$, we have

$$E(B_{k+2}) - E(B_{k+1}) = -\eta \lambda^* + o(\eta) \leq 0 \quad (\text{B.24})$$

Note that we have $E(B_k) \geq 0$. Then from (B.23), the convergence of $E(B_k)$ is now proven for a sufficiently small η . On the other side, since the value of the cost function $E(B_k)$ keeps strictly non-increasing as the iteration goes on, the network controllability is naturally guaranteed at each iteration step k . Because once (A, B_k) becomes uncontrollable, the Gramian matrix W_{B_k} becomes invertible. This leads to an infinite $E(B_k)$, which is in contradiction to the non-increasing of $E(B_k)$ during the iteration process. $\square$

Appendix C

Proofs for Chapter 6

C.1 Proof of Lemma 6.3

Based on Lemma 2.1, the derivative can be expressed as

$$\begin{aligned} & \frac{\partial \text{tr}(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f})}{\partial A_{ij}} \\ &= -\delta_{mn} \cdot \frac{\partial [W_A^{-1}]_{mu}}{\partial A_{ij}} \cdot [e^{At_f}]_{uz} \cdot [\mathbf{x}_0 \mathbf{x}_0^T]_{zc} \cdot [e^{A^T t_f}]_{cn} \\ & \quad + \delta_{mn} \cdot [W_A^{-1}]_{mu} \cdot \frac{\partial [e^{At_f}]_{uz}}{\partial A_{ij}} \cdot [\mathbf{x}_0 \mathbf{x}_0^T]_{zc} \cdot [e^{A^T t_f}]_{cn} \\ & \quad + \delta_{mn} \cdot [W_A^{-1}]_{mu} \cdot [e^{At_f}]_{uz} \cdot [\mathbf{x}_0 \mathbf{x}_0^T]_{zc} \cdot \frac{\partial [e^{A^T t_f}]_{cn}}{\partial A_{ij}} \end{aligned} \tag{C.1}$$

By computing $\frac{\partial[W_A^{-1}]_{mu}}{\partial A_{ij}}$, $\frac{\partial[e^{At_f}]_{uz}}{\partial A_{ij}}$ and $\frac{\partial[e^{A^T t_f}]_{cn}}{\partial A_{ij}}$ based on Lemmas 6.1 and 6.2, respectively, we obtain

$$\begin{aligned}
& \frac{\partial \text{tr}(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f})}{\partial A_{ij}} \\
&= -\delta_{mn} \cdot \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} A^{k_0}]_{mi} \cdot [A^{n_0-1-k_0} B B^T e^{A^T t} W_A^{-1}]_{ju} \right) \right) dt \cdot [e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f}]_{un} \\
&\quad - \delta_{mn} \cdot \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} e^{At} B B^T (A^{n_0-1-k_0})^T]_{mj} \cdot [(A^{k_0})^T W_A^{-1}]_{iu} \right) \right) dt \cdot [e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f}]_{un} \\
&\quad + \delta_{mn} \cdot \left[[W_A^{-1}]_{mu} \cdot \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \left(\sum_{k_0=0}^{n_0-1} \left([A^{k_0}]_{ui} \cdot [A^{n_0-1-k_0}]_{jz} \right) \right) \cdot [\mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f}]_{zn} \right] \\
&\quad + \delta_{mn} \cdot \left[[W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T]_{mc} \cdot \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \left(\sum_{k_0=0}^{n_0-1} \left([A^{k_0}]_{in}^T \cdot [A^{n_0-1-k_0}]_{cj}^T \right) \right) \right]
\end{aligned} \tag{C.2}$$

From Lemma 2.1, (C.2) can be rewritten as the following form of $[\bullet]_{ij}$ by rearranging the orders of the terms:

$$\begin{aligned}
& \frac{\partial \text{tr}(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f})}{\partial A_{ij}} \\
&= - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} A^{k_0}]_{ni} \cdot [A^{n_0-1-k_0} B B^T e^{A^T t} W_A^{-1}]_{ju} \right) \right) dt \cdot [e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f}]_{un} \\
&\quad - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([W_A^{-1} e^{At} B B^T (A^{n_0-1-k_0})^T]_{nj} \cdot [(A^{k_0})^T W_A^{-1}]_{iu} \right) \right) dt \cdot [e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f}]_{un} \\
&\quad + [W_A^{-1}]_{nu} \cdot \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \left(\sum_{k_0=0}^{n_0-1} \left([A^{k_0}]_{ui} \cdot [A^{n_0-1-k_0}]_{jz} \right) \right) \cdot [\mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f}]_{zn} \\
&\quad + [W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T]_{nc} \cdot \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \left(\sum_{k_0=0}^{n_0-1} \left([A^{k_0}]_{in}^T \cdot [A^{n_0-1-k_0}]_{cj}^T \right) \right) \\
&= - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([(A^T)^{k_0} W_A^{-T} e^{At_f} x_0 x_0^T e^{A^T t_f} W_A^{-T} e^{At} B B^T (A^T)^{n_0-1-k_0}]_{ij} \right) \right) dt \\
&\quad - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \left(\sum_{k_0=0}^{n_0-1} \left([(A^T)^{k_0} W_A^{-1} e^{At_f} x_0 x_0^T e^{A^T t_f} W_A^{-1} e^{At} B B^T (A^T)^{n_0-1-k_0}]_{ij} \right) \right) dt \\
&\quad + \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \left(\sum_{k_0=0}^{n_0-1} [(A^T)^{k_0} W_A^{-T} e^{At_f} x_0 x_0^T (A^T)^{n_0-1-k_0}]_{ij} \right) \\
&\quad + \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \left(\sum_{k_0=0}^{n_0-1} [(A^T)^{k_0} W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T (A^T)^{n_0-1-k_0}]_{ij} \right)
\end{aligned} \tag{C.3}$$

Again, based on *I-Chain Rule*, we obtain that

$$\begin{aligned}
& \frac{\partial \operatorname{tr}(W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T e^{A^T t_f})}{\partial A} \\
&= - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-T} e^{At_f} x_0 x_0^T e^{A^T t_f} W_A^{-T} e^{At} B B^T (A^T)^{n_0-1-k_0} dt \\
&\quad - \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-1} e^{At_f} x_0 x_0^T e^{A^T t_f} W_A^{-1} e^{At} B B^T (A^T)^{n_0-1-k_0} dt \\
&\quad + \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-T} e^{At_f} x_0 x_0^T (A^T)^{n_0-1-k_0} \\
&\quad + \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T (A^T)^{n_0-1-k_0} \\
&= -2 \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-1} e^{At_f} x_0 x_0^T e^{A^T t_f} W_A^{-1} e^{At} B B^T (A^T)^{n_0-1-k_0} dt \\
&\quad + 2 \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-1} e^{At_f} \mathbf{x}_0 \mathbf{x}_0^T (A^T)^{n_0-1-k_0}
\end{aligned} \tag{C.4}$$

Thus, this lemma holds. $\square$

Author's Publications

Journal Articles

- G. Li*, **J. Ding*** and C. Wen, “Controlling Directed Networks with Evolving Topologies”, *IEEE Trans. on Control of Network Systems*, accepted. (*contributed equally)
- **J. Ding**, G. Li and C. Wen, “Key node selection in optimal control of complex networks”, *Physica A: Statistical Mechanics and its Applications*, 486, 251-261, 2017.
- G. Li*, **J. Ding*** and C. Wen, “Open-Loop Optimal Control of Directed Networks with Selectable Control Inputs”, *IEEE Trans. on Cybernetics*, under third round review. (*contributed equally)
- **J. Ding**, C. Wen and G. Li, “Key node selection of complex networks via convex optimization”, submitted to *IEEE Trans. on Cybernetics*.
- **J. Ding**, C. Wen and G. Li, “Sparsity-promoting optimal control in complex networks”, to be submitted.
- **J. Ding**, C. Wen, G. Li and C. Chua, “Locality Sensitive Batch Feature Extraction For high-dimensional Data”, *Neurocomputing*, 171, 664-672, 2016.
- G. Li, **J. Ding**, C. Wen and J. Pei, “Optimal Control of Complex Networks based on matrix differentiation”, *Europhysics Letters*, 115(6), 68005, 2016.
- G. Li, C. Wen, W. Wei, Y. Xu, **J. Ding**, G. Zhao and L. Shi, “Trace ratio criterion for feature extraction in classification”, *Mathematical Problems in Engineering*, vol. 2014, pp. 725204, 2014.

Conference Proceedings

- **J. Ding**, C. Wen and G. Li, “Optimal Control of Weighted Networks Based on Node Connection Strength”, *The 26th IEEE International Symposium on Industrial Electronics*, 2017.
- **J. Ding**, G. Li, C. Wen and C. Chua, “Min-Max Discriminant Analysis Based On Gradient Method For Feature Extraction”, *13th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, 2014.
- **J. Ding**, C. Wen and G. Li, “Sparsity-inspired optimal control in complex networks”, *The 27th IEEE International Symposium on Industrial Electronics*, 2018.

Bibliography

- [1] Justin Ruths and Derek Ruths. Control profiles of complex networks. *Science*, 343(6177):1373–1376, 2014.
- [2] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
- [3] Sameer Tilak, Nael B Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36, 2002.
- [4] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdury. Wireless multimedia sensor networks: A survey. *Wireless Communications, IEEE*, 14(6):32–39, 2007.
- [5] Hassan Farhangi. The path of the smart grid. *Power and Energy Magazine, IEEE*, 8(1):18–28, 2010.
- [6] S Massoud Amin and Bruce F Wollenberg. Toward a smart grid: power delivery for the 21st century. *Power and Energy Magazine, IEEE*, 3(5):34–41, 2005.
- [7] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.
- [8] Guido Caldarelli. Scale-free networks: complex webs in nature and technology. *OUP Catalogue*, 2007.
- [9] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.
- [10] Steven H Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

- [11] Shi Gu, Richard F Betzel, Marcelo G Mattar, Matthew Cieslak, Philip R Delio, Scott T Grafton, Fabio Pasqualetti, and Danielle S Bassett. Optimal trajectories of brain state transitions. *NeuroImage*, 148:305–317, 2017.
- [12] Shi Gu, Fabio Pasqualetti, Matthew Cieslak, Qawi K Telesford, B Yu Alfred, Ari E Kahn, John D Medaglia, Jean M Vettel, Michael B Miller, Scott T Grafton, et al. Controllability of structural brain networks. *Nature communications*, 6, 2015.
- [13] Jayanth R Banavar, Amos Maritan, and Andrea Rinaldo. Size and form in efficient transportation networks. *Nature*, 399(6732):130, 1999.
- [14] Roger Guimera, Stefano Mossa, Adrian Turtzchi, and LA Nunes Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities’ global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799, 2005.
- [15] GC D’ans and DC Gazis. Optimal control of oversaturated store-and-forward transportation networks. *Transportation Science*, 10(1):1–19, 1976.
- [16] Wenbo Shi, Xiaorong Xie, Chi-Cheng Chu, and Rajit Gadh. Distributed optimal energy management in microgrids. *IEEE Transactions on Smart Grid*, 6(3):1137–1146, 2015.
- [17] Cameron Nowzari, Victor M Preciado, and George J Pappas. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems*, 36(1):26–46, 2016.
- [18] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.
- [19] Ching Tai Lin. Structural controllability. *Automatic Control, IEEE Transactions on*, 19(3):201–208, 1974.
- [20] Alex Olshevsky. Minimal controllability problems. *IEEE Transactions on Control of Network Systems*, 1(3):249–258, 2014.
- [21] Yang-Yu Liu and Albert-Laszló Barabási. Control principles of complex networks. *arXiv preprint arXiv:1508.05384*, 2015.
- [22] Sérgio Pequito, Soummya Kar, and A Pedro Aguiar. Minimum cost input/output design for large-scale linear structural systems. *Automatica*, 68:384–391, 2016.
- [23] Sérgio Pequito, Guilherme Ramos, Soummya Kar, A Pedro Aguiar, Ramos, and Jaime. The robust minimal controllability problem.

- [24] Alex Olshevsky. Minimum input selection for structural controllability. In *2015 American Control Conference (ACC)*, pages 2218–2223. IEEE, 2015.
- [25] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004.
- [26] Rembrandt Bakker, Thomas Wachtler, and Markus Diesmann. CoCoMac 2.0 and the future of tract-tracing databases. *Frontiers in Neuroinformatics*, 6, 2012.
- [27] Tom Gommans, Duarte Antunes, Tijs Donkers, Paulo Tabuada, and Maurice Heemels. Self-triggered linear quadratic control. *Automatica*, 50(4):1279–1287, 2014.
- [28] John C Doyle, Keith Glover, Pramod P Khargonekar, and Bruce A Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic control*, 34(8):831–847, 1989.
- [29] Kristian Hengster Movric and Frank L Lewis. Cooperative optimal control for multi-agent systems on directed graph topologies. *Automatic Control, IEEE Transactions on*, 59(3):769–774, 2014.
- [30] BP Molinari. The time-invariant linear-quadratic optimal control problem. *Automatica*, 13(4):347–357, 1977.
- [31] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1): 3–20, 2002.
- [32] Thang Nguyen and Zoran Gajic. Solving the matrix differential riccati equation: a lyapunov equation approach. *IEEE Transactions on Automatic Control*, 55(1): 191–194, 2010.
- [33] Ruth F Curtain. Riccati equations for second order spatially invariant partial differential systems. *Automatica*, 48(1):139–144, 2012.
- [34] Juanjuan Xu and Huanshui Zhang. Sufficient and necessary open-loop stackelberg strategy for two-player game with time delay. *IEEE transactions on cybernetics*, 46(2):438–449, 2016.
- [35] Tony Stillfjord. Low-rank second-order splitting of large-scale differential riccati equations. *IEEE Transactions on Automatic Control*, 60(10):2791–2796, 2015.

- [36] Manuel Jiménez-Lizárraga, Michael Basin, Victoria Rodríguez, and Pablo Rodríguez. Open-loop nash equilibrium in polynomial differential games via state-dependent riccati equation. *Automatica*, 53:155–163, 2015.
- [37] Gang Yan, Jie Ren, Ying-Cheng Lai, Choy-Heng Lai, and Baowen Li. Controlling complex networks: How much energy is needed? *Physical review letters*, 108(21): 218703, 2012.
- [38] Gang Yan, Georgios Tsekenis, Baruch Barzel, Jean-Jacques Slotine, Yang-Yu Liu, and Albert-Laszlo Barabasi. Spectrum of controlling and observing complex networks. *arXiv preprint arXiv:1503.01160*, 2015.
- [39] Fabio Pasqualetti, Sandro Zampieri, and Francesco Bullo. Controllability metrics, limitations and algorithms for complex networks. *IEEE Transactions on Control of Network Systems*, 1(1):40–52, 2014.
- [40] Tyler Summers, Fabrizio Cortesi, and John Lygeros. On submodularity and controllability in complex dynamical networks. 2014.
- [41] Thomas E Gorochoowski, Mario Di Bernardo, and Claire S Grierson. Evolving dynamical networks: a formalism for describing complex systems. *Complexity*, 17(3):18–25, 2012.
- [42] Pietro DeLellis, Mario di Bernardo, and Maurizio Porfiri. Pinning control of complex networks via edge snapping. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(3):033119, 2011.
- [43] George CMA Ehrhardt, Matteo Marsili, and Fernando Vega-Redondo. Phenomenological models of socioeconomic network dynamics. *Physical Review E*, 74(3): 036106, 2006.
- [44] João da Gama Batista, Jean-Philippe Bouchaud, and Damien Challet. Sudden trust collapse in networked societies. *The European Physical Journal B*, 88(3):55, 2015.
- [45] Daniel Yamins, Steen Rasmussen, and David Fogel. Growing urban roads. *Networks and Spatial Economics*, 3(1):69–85, 2003.
- [46] Danielle Smith Bassett and ED Bullmore. Small-world brain networks. *The neuroscientist*, 12(6):512–523, 2006.
- [47] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

- [48] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.
- [49] Arne D Ekstrom, Michael J Kahana, Jeremy B Caplan, Tony A Fields, Eve A Isham, Ehren L Newman, and Itzhak Fried. Cellular networks underlying human spatial navigation. *Nature*, 425(6954):184–188, 2003.
- [50] Wilson J Rugh. *Linear system theory*, volume 2. prentice hall Upper Saddle River, NJ, 1996.
- [51] Chi-Tsong Chen. *Linear system theory and design*. Oxford University Press, Inc., 1995.
- [52] Fabrizio L Cortesi, Tyler H Summers, and John Lygeros. Submodularity of energy related controllability metrics. In *53rd IEEE Conference on Decision and Control*, pages 2883–2888. IEEE, 2014.
- [53] PC Müller and HI Weber. Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems. *Automatica*, 8(3):237–246, 1972.
- [54] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.
- [55] Paul Erdős and Alfréd Rényi. On random graphs I. *Publ. Math. Debrecen*, 6: 290–297, 1959.
- [56] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.
- [57] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [58] Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.
- [59] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.
- [60] Guoqi Li, Pei Tang, Ziyang Meng, Changyun Wen, Jing Pei, and Luping Shi. Matrix function optimization problems. *IEEE Transactions on Neural Networks and Learning Systems*, submitted.

- [61] Tyler H Summers, Fabrizio L Cortesi, and John Lygeros. On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1):91–101, 2016.
- [62] Guoqi Li, Wuhua Hu, Gaoxi Xiao, Lei Deng, Pei Tang, Jing Pei, and Luping Shi. Minimum-cost control of complex networks. *New Journal of Physics*, 18(1):013012, 2015.
- [63] Ken Binmore and Joan Davies. *Calculus: concepts and methods*. Cambridge University Press, 2002.
- [64] Karim M Abadir and Jan R Magnus. *Matrix algebra*, volume 1. Cambridge University Press, 2005.
- [65] Erik W Grafarend and Rey-Jer You. Fourth order taylor–kármán structured covariance tensor for gravity gradient predictions by means of the hankel transformation. *International Journal on Geomathematics*, pages 1–24, 2015.
- [66] Jan R Magnus, Heinz Neudecker, et al. Matrix differential calculus with applications in statistics and econometrics. 1995.
- [67] Randal J Barnes. Matrix differentiation. 2014.
- [68] Mahdi Jalili, Omid Askari Sichani, and Xinghuo Yu. Optimal pinning controllability of complex networks: Dependence on network structure. *Physical Review E*, 91(1):012803, 2015.
- [69] Wenwu Yu, Guanrong Chen, and Jinhua Lü. On pinning synchronization of complex dynamical networks. *Automatica*, 45(2):429–435, 2009.
- [70] Fu Lin, Makan Fardad, and Mihailo R Jovanovic. Algorithms for leader selection in stochastically forced consensus networks. *IEEE Transactions on Automatic Control*, 59(7):1789–1802, 2014.
- [71] Andrew Clark, Basel Alomair, Linda Bushnell, and Radha Poovendran. Minimizing convergence error in multi-agent systems via leader selection: A supermodular optimization approach. *IEEE Transactions on Automatic Control*, 59(6):1480–1494, 2014.
- [72] Ya-xiang Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151(1):249–281, 2015.
- [73] Jie Sun and Adilson E Motter. Controllability transition and nonlocality in network control. *Physical Review Letters*, 110(20):208701, 2013.

- [74] Guoqi Li, Ning Ning, Kiruthika Ramanathan, Wei He, Li Pan, and Luping Shi. Behind the magical numbers: hierarchical chunking and the human working memory capacity. *International journal of neural systems*, 23(04):1350019, 2013.
- [75] Fernand Gobet, Peter CR Lane, Steve Croker, Peter CH Cheng, Gary Jones, Iain Oliver, and Julian M Pine. Chunking mechanisms in human learning. *Trends in cognitive sciences*, 5(6):236–243, 2001.
- [76] Zhijian Chen and Nelson Cowan. Chunk limits and length limits in immediate recall: a reconciliation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(6):1235, 2005.
- [77] James Deese and Roger A Kaufman. Serial effects in recall of unorganized and sequentially organized verbal material. *Journal of experimental psychology*, 54(3):180, 1957.
- [78] Robert H Logie. *Visuo-spatial working memory*. Psychology Press, 2014.
- [79] Eddy J Davelaar, Yonatan Goshen-Gottstein, Amir Ashkenazi, Henk J Haarmann, and Marius Usher. The demise of short-term memory revisited: empirical and computational investigations of recency effects. *Psychological review*, 112(1):3, 2005.
- [80] Nikos Gorgoraptis, Raquel FG Catalao, Paul M Bays, and Masud Husain. Dynamic updating of working memory resources for visual objects. *The Journal of Neuroscience*, 31(23):8502–8511, 2011.
- [81] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440–442, 1998.
- [82] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464:1025–1028, 2010.
- [83] Albert-László Barabási. Scale-free networks: a decade and beyond. *science*, 325(5939):412–413, 2009.
- [84] Giulia Menichetti, Luca DallAsta, and Ginestra Bianconi. Network controllability is determined by the density of low in-degree and out-degree nodes. *Physical review letters*, 113(7):078701, 2014.
- [85] Yu-Zhong Chen, Lezhi Wang, Wenxu Wang, and Ying-Cheng Lai. The paradox of controlling complex networks: control inputs versus energy requirement. *arXiv preprint arXiv:1509.03196*, 2015.

- [86] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.
- [87] Ronald S Burt. Social contagion and innovation: Cohesion versus structural equivalence. *American journal of Sociology*, 92(6):1287–1335, 1987.
- [88] Robert R Christian and Joseph J Luczkovich. Organizing and understanding a winters seagrass foodweb network through effective trophic levels. *Ecological modelling*, 117(1):99–124, 1999.
- [89] Vasileios Tzoumas, Mohammad Amin Rahimian, George J Pappas, and Ali Jad-babaie. Minimal actuator placement with bounds on control effort. *IEEE Transactions on Control of Network Systems*, 3(1):67–78, 2016.
- [90] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [91] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [92] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [93] Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. A forward-backward bregman splitting scheme for regularized distributed optimization problems. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1093–1098. IEEE, 2016.
- [94] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
- [95] Xiaoqun Zhang, Martin Burger, Xavier Bresson, and Stanley Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.
- [96] Guoqi Li, Jie Ding, Changyun Wen, and Jing Pei. Optimal control of complex networks based on matrix differentiation. *EPL (Europhysics Letters)*, 115(6):68005, 2016.
- [97] Le-Zhi Wang, Yu-Zhong Chen, Wen-Xu Wang, and Ying-Cheng Lai. Physical controllability of complex networks. *Scientific reports*, 7, 2017.

- [98] Yu-Zhong Chen, Le-Zhi Wang, Wen-Xu Wang, and Ying-Cheng Lai. Energy scaling and reduction in controlling complex networks. *Royal Society open science*, 3(4): 160064, 2016.
- [99] Siddharth Joshi and Stephen Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.
- [100] Alan J Miller and Nam-Ky Nguyen. Algorithm as 295: A fedorov exchange algorithm for d-optimal design. *Journal of the royal statistical society. series c (applied statistics)*, 43(4):669–677, 1994.
- [101] Henry P Wynn. Results in the theory and construction of d-optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 133–147, 1972.
- [102] Huaqing Li, Guo Chen, Tingwen Huang, Zhaoyang Dong, Wei Zhu, and Lan Gao. Event-triggered distributed average consensus over directed digital networks with limited communication bandwidth. *IEEE transactions on cybernetics*, 46(12):3098–3110, 2016.
- [103] Zhengzhong Yuan, Chen Zhao, Zengru Di, Wen-Xu Wang, and Ying-Cheng Lai. Exact controllability of complex networks. *Nature communications*, 4, 2013.
- [104] Rudolf Emil Kalman. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2): 152–192, 1963.
- [105] Filippo Menolascina, Gianfranco Fiore, Emanuele Orabona, Luca De Stefano, Mike Ferry, Jeff Hasty, Mario di Bernardo, and Diego di Bernardo. In-vivo real-time control of protein expression from endogenous and synthetic gene networks. *PLoS computational biology*, 10(5):e1003625, 2014.
- [106] Amirreza Rahmani, Meng Ji, Mehran Mesbahi, and Magnus Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186, 2009.
- [107] Daron Acemoglu, Asuman Ozdaglar, and Ali ParandehGheibi. Spread of (mis) information in social networks. *Games and Economic Behavior*, 70(2):194–227, 2010.
- [108] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature reviews. Molecular cell biology*, 9(10):770, 2008.

- [109] BY ALBERT-LÁSZLÓ Barabási and Eric Bonabeau. Scale-free networks. *Scientific American*, 288(5):50–59, 2003.
- [110] Christian Commault and Jean-Michel Dion. Input addition and leader selection for the controllability of graph-based systems. *Automatica*, 49(11):3322–3328, 2013.
- [111] Sergio Pequito, Soumya Kar, and A Pedro Aguiar. On the complexity of the constrained input selection problem for structural linear systems. *Automatica*, 62: 193–199, 2015.
- [112] Tao Yang, Jie Lu, Di Wu, Junfeng Wu, Guodong Shi, Ziyang Meng, and Karl Henrik Johansson. A distributed algorithm for economic dispatch over time-varying directed networks with delays. *IEEE Transactions on Industrial Electronics*, 64(6): 5095–5106, 2017.
- [113] Sérgio Pequito, James Svacha, George J Pappas, and Vijay Kumar. Sparsest minimum multiple-cost structural leader. *IFAC-PapersOnLine*, 48(22):144–149, 2015.
- [114] Guoqi Li, Pei Tang, Changyun Wen, and Ziyang Meng. Boundary constraints for minimum cost control of directed networks. *IEEE transactions on cybernetics*, 2016.
- [115] Edda Klipp, Wolfram Liebermeister, Christoph Wierling, Axel Kowald, and Ralf Herwig. *Systems biology: a textbook*. John Wiley & Sons, 2016.
- [116] Anthony M Bloch, Peter E Crouch, and Amit K Sanyal. A variational problem on stiefel manifolds. *Nonlinearity*, 19(10):2247, 2006.
- [117] Charles W Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):93–101, 1986.
- [118] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.
- [119] Peter Lancaster. *Theory of matrices*. Academic Press, New York, 1969.
- [120] Jimmie Gilbert and Linda Gilbert. *Linear Algebra and Matrix Theory*. Academic Press, 2014.
- [121] Dan Kalman. A singularly valuable decomposition: the svd of a matrix. *The college mathematics journal*, 27(1):2–23, 1996.

- [122] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *Signal Processing Conference, 2007 15th European*, pages 606–610. IEEE, 2007.
- [123] Chaoming Song, Shlomo Havlin, and Hernan A Makse. Self-similarity of complex networks. *Nature*, 433(7024):392–395, 2005.
- [124] Sean J Bennett and Renjie Liu. Basin self-similarity, hacks law, and the evolution of experimental rill networks. *Geology*, 44(1):35–38, 2016.
- [125] Hanhua Chen, Hai Jin, and Shaoliang Wu. Minimizing inter-server communications by exploiting self-similarity in online social networks. *IEEE Transactions on Parallel and Distributed Systems*, 27(4):1116–1130, 2016.
- [126] IE Leonard. The matrix exponential. *SIAM review*, 38(3):507–512, 1996.
- [127] Jon Dattorro. *Convex optimization & Euclidean distance geometry*. Lulu. com, 2010.
- [128] Sever S Dragomir. A survey on cauchy-bunyakovsky-schwarz type discrete inequalities. *J. Inequal. Pure Appl. Math*, 4(3):1–142, 2003.
- [129] Daan Bloembergen, Bijan Ranjbar-Sahraei, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. Influencing social networks: An optimal control study. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, pages 105–110. IOS Press, 2014.
- [130] Sergei N Dorogovtsev and José FF Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. OUP Oxford, 2013.
- [131] Ann E Krause, Kenneth A Frank, Doran M Mason, Robert E Ulanowicz, and William W Taylor. Compartments revealed in food-web structure. *Nature*, 426(6964):282–285, 2003.
- [132] E. Almaas, B Kovacs, T Vicsek, ZN Oltvai, and A-L Barabási. Global organization of metabolic fluxes in the bacterium escherichia coli. *Nature*, 427(6977):839–843, 2004.
- [133] Neli Blagus, Lovro Šubelj, and Marko Bajec. Self-similar scaling of density in complex real-world networks. *Physica A: Statistical Mechanics and its Applications*, 391(8):2794–2802, 2012.

-
- [134] Mohammad Babakmehr, Marcelo Godoy Simoes, Michael B Wakin, Ahmed Al Durra, and Farnaz Harirchi. Smart-grid topology identification using sparse recovery. *IEEE Transactions on Industry Applications*, 52(5):4375–4384, 2016.
 - [135] Xiaofan Wu and Mihailo R Jovanovic. Sparsity-promoting optimal control of consensus and synchronization networks. In *American Control Conference (ACC), 2014*, pages 2936–2941. IEEE, 2014.
 - [136] Curt Schurgers, Vlasios Tsiatsis, and Mani B Srivastava. Stem: Topology management for energy efficient sensor networks. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, pages 3–3. IEEE, 2002.
 - [137] Richard C Dorf and Robert H Bishop. *Modern control systems*. Pearson, 2011.
 - [138] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2013.