ANALYSIS OF CONTROLLABILITY FOR TEMPORAL NETWORKS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Babak Ravandi

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Fatma Mili, Chair

College of Computing and Informatics, University of North Carolina at Charlotte

Dr. John A Springer, Co-Chair

Department of Computer and Information Technology, Purdue University

Dr. Ioannis Papapanagiotou

Platform Engineering, Netflix

Dr. Marcus Rogers

Department of Computer and Information Technology, Purdue University

Dr. Joaquín Goñi

School of Industrial Engineering, Purdue University

Approved by:

Dr. Kathryne A Newton

Associate Dean for Graduate Programs, Purdue Polytechnic Institute

Dedicated

to my family, my parents and my brother for their unconditional kindness and support.

ACKNOWLEDGMENTS

The author extends his highest appreciation and respect to his advisor, Professor Fatma Mili, for her continued support and help throughout his doctoral program. He is in her debt forever. The author also thanks his co-advisor Professor John Springer for his instructions. The author appreciates Professor Marcus Rogers, Ioannis Papapanagiotou, and Joaquín Goñi for accepting to be in his committee. The author gives special thanks to Professor Guity Ravai for her kindness and years of support.

TABLE OF CONTENTS

Page

LIST OF TABLES viii

LIST OF FIGURES ix

ABBREVIATIONS x

ABSTRACT xi

1 INTRODUCTION 1

1.1 Scope 1

1.2 Significance 3

1.3 Research Questions 4

1.4 Limitations 4

1.5 Assumptions 5

1.6 Delimitations 5

2 REVIEW OF RELEVANT LITERATURE 6

2.1 Complex Systems Theory and Network Thinking 6

2.1.1 Temporal Networks 8

2.2 Controllability of Time-Aggregated Networks 9

2.3 Controllability of Temporal Networks 11

2.3.1 Independent Path Theorem 13

2.3.2 Maximum Controllable Subspace 14

2.4 Control Characteristics of Static Networks 15

2.5 Summary 17

3 FRAMEWORK AND METHODOLOGY 18

3.1 Approach 18

3.1.1 Objective I: Create Minimum Driver Node Sets 18

3.1.2 Objective II: Identify All MCSs of Nodes in Temporal Networks 19

Page

3.1.3 Objective III: Analysis of the Controllable Space of Nodes from the Controllability Perspective . . . 20
3.2 Novelty . . . 20
3.3 Datasets . . . 21
3.4 Measure of Success . . . 24
3.5 Summary . . . 24
4 IDENTIFYING DRIVER NODES FOR TEMPORAL NETWORKS . . . 25
4.1 Heuristic Approach for Identifying SMDSs . . . 26
4.1.1 Multiple Controllable Subspaces and Branching Process . . . 28
4.1.2 Heuristic Approach Algorithm . . . 31
4.2 Results . . . 34
4.2.1 Identifying Multiple Driver Node Sets . . . 35
4.2.2 Queen Ants . . . 37
4.2.3 Degree Distribution of MDSs and SMDSs . . . 38
4.2.4 Evaluations Using Network Randomizations . . . 39
4.2.5 Synthetic Temporal Networks . . . 42
4.3 Conclusion . . . 43
5 COMPLETE CONTROLLABLE DOMAIN . . . 44
5.1 Exponential Number of MCSs . . . 44
5.2 Algorithm to Identify k MCSs . . . 45
5.3 Computational Complexity . . . 49
5.4 Algorithm Optimization . . . 49
5.5 Evaluation . . . 50
5.6 Conclusion . . . 51
6 ANALYSIS OF COMPLETE CONTROLLABLE DOMAIN . . . 52
6.1 Convergence of MCSs-Intersection on k . . . 53
6.2 MCSs and Driver Nodes Sets . . . 54
6.3 Conclusion . . . 56

Page

7 DISCUSSIONS AND FINDINGS 57
7.1 Control Regimes 58
7.2 Future Recommendations 60
7.3 Conclusion 61
REFERENCES 62

LIST OF TABLES

Table Page

4.1 Numerical results for MDSs and SMDSs. 36

4.2 Queen ants' interactions. 38

4.3 Randomization techniques. 42

LIST OF FIGURES

Figure Page

1.1 Temporal and time-aggregated representation of networks. 2

2.1 Temporal correlations in pathways. 9

2.2 Controlling a simple temporal network. 13

2.3 Identifying a maximum controllable subspace. 14

2.4 Control categories of nodes in static networks (Courtesy Jia et al. [14]). . 16

3.1 Network characteristics of the datasets. 22

3.2 Frequency of interactions in the datasets. 23

4.1 Heuristic approach execution on a temporal network. 27

4.2 Non-unique MCSs. 28

4.3 Branching process for an MCSs-list. 29

4.4 Complete illustration of the branching procedure. 30

4.5 Cardinality of MDSs and SMDSs for ant networks. 35

4.6 Driver nodes degree distributions. 39

4.7 Randomization results. 40

4.8 Synthetic temporal networks. 43

5.1 Driver nodes can have an exponential number of MCSs. 44

5.2 A simple temporal network with 6 MCSs. 46

5.3 Execution of the proposed algorithm on a temporal network. 47

5.4 Average height of n_d-ary trees in real-world temporal networks. 50

5.5 Performance of the branch and bound algorithm. 51

6.1 Convergence of MCSs-intersections on small k 53

6.2 Participation in driver node sets 55

7.1 MDSs and SMDSs characteristics for ants and emails network. 59

ABBREVIATIONS

MDS	Minimum Driver node Set
SMDS	Suboptimal Minimum Driver node Set
MCS	Maximum Controllable Subspace
CCD	Complete Controllable Domain

ABSTRACT

Ravandi, Babak PhD, Purdue University, December 2019. Analysis of Controllability for Temporal Networks . Major Professors: Fatma Mili and John A. Springer.

Physical systems modeled by networks are fully dynamic in the sense that the process of adding edges and vertices never ends, and no edge or vertex is necessarily eternal. Temporal networks enable to explicitly study systems with a changing topology by capturing explicitly the temporal changes. The controllability of temporal networks is the study of driving the state of a temporal network to a target state at deadline t_f within $\Delta t = t_f - t_0$ steps by stimulating key nodes called driver nodes. In this research, the author aims to understand and analyze temporal networks from the controllability perspective at the global and nodal scales. To analyze the controllability at global scale, the author provides an efficient heuristic algorithm to build driver node sets capable of fully controlling temporal networks. At the nodal scale, the author presents the concept of Complete Controllable Domain (CCD) to investigate the characteristics of Maximum Controllable Subspaces (MCSs) of a driver node. The author shows that a driver node can have an exponential number of MCSs and introduces a branch and bound algorithm to approximate the CCD of a driver node. The proposed algorithms are evaluated on real-world temporal networks induced from ant interactions in six colonies and in a set of e-mail communications of a manufacturing company. At the global scale, the author provides ways to determine the control regime in which a network operates. Through empirical analysis, the author shows that ant interaction networks operate under a distributed control regime whereas the e-mails network operates in a centralized regime. At the nodal scale, the analysis indicated that on average the number of nodes that a driver node always controls is equal to the number of driver nodes that always control a node.

1. INTRODUCTION

The vision and perspective of this dissertation is provided in this chapter. In addition, the significance of the proposed research in the study of structural controllability is established. Moreover, the research questions, assumptions, scope, and limitations, and delimitations of this research are presented.

1.1 Scope

Networks are abundant around us. Their fingerprint exists within our biology (network of blood vessels and neurons), our technologies (Internet, roads, and transportation networks), and our languages (network of syntactic dependencies). Understanding the structure and behavior of such networks significantly contributes to our understanding of complex systems. The term network represents graphs that model physical systems (a graph is the mathematical representation of a network). A graph is a simple structure with two elements: vertices (nodes) that represent the components of a system and edges (links) that represent the interactions between those components. Researchers use networks to model the structure and dynamics of physical systems.

This research is focused on the controllability aspects of temporal networks. In general, the controllability of complex networks is the study of input and output characteristics of graphs [1, 2]. These complex systems receive stimuli (inputs) and produce effects (outputs) by interacting with an environment. Temporal networks capture the timescale of interactions between a system's components in the structure of networks [3]. In contrast, static networks aggregate the order of these interactions within the intervals in which they occurred [4]. As illustrated in Fig. 1.1, two different temporal networks can have an identical static network. Indeed in Fig. 1.1 (a-b), the

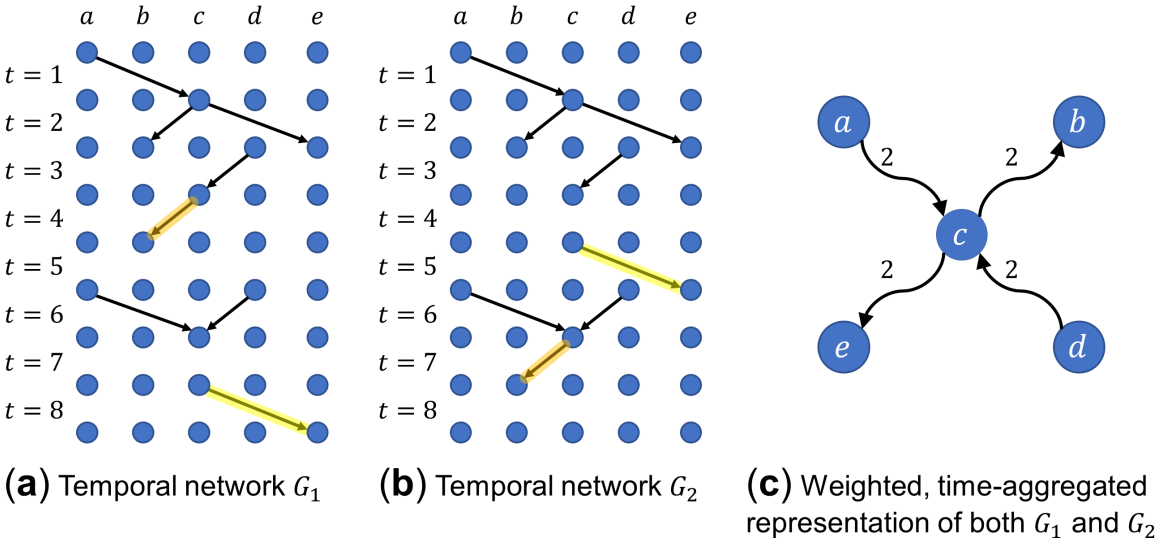

Fig. 1.1. Temporal and time-aggregated representation of networks.

temporal networks G_1 and G_2 differ only in the timescale of two links $c \to b$ and $c \to e$. However, their time-aggregated network over time intervals $1 \leq t \leq 8$ is captured by the same static graph depicted in Fig. 1.1 (c). Note that the static graph contains a path $a \to b \to c$, even though there is no temporal pathway from $a \dashrightarrow c$.

The controllability of temporal networks is the study of driving the state of a network to a targeted state at deadline t_f within $\Delta t = t_f - t_0$ steps by applying stimulations to key nodes called *driver nodes*. Pósfai and Hövel [5] applied the definition of structural controllability to temporal networks under the linear time-varying dynamics. They provided a method to efficiently determine a single Maximum Controllable Subspace (MCS), denoted by $M_C(D, t_f, \Delta t) \subseteq V$, for a set of driver nodes $D \subseteq V$. This research aims to understand and analyze the behavior of temporal networks from the controllability perspective.

1.2 Significance

Complex networks have widely been used to characterize numerous physical and phenomenological systems in the world. The applications of complex networks cover many scientific disciplines including applied and abstract physics [6, 7], chemistry [8], biology [9, 10] as well as social sciences [11]. Controlling such systems has been a motivation for the scientific community [12].

The main contributions of this work are the following two algorithms: a) the author introduced a heuristic approach to create driver node sets capable of controlling temporal networks, and b) the author developed a branch and bound algorithm to determine all MCSs of a driver node. A few applications of the proposed methods in this dissertation are as follows:

- The algorithms to create driver node sets and identify all MCSs can be extended to design networks with an expected control behavior. For instance, designing networks that rely on many nodes to be controlled or in contrast designing networks that rely on a few nodes to become controllable.

- One can study the evolution of MCSs in time to detect emergent behaviors in the context of a complex system. Such a study could shed light on the emergence of distributed and centralized control regimes in complex networks. A network is in a distributed control regime if many groups of nodes with a small size can control the network. On the other hand, in a centralized control regime only a few groups of nodes with many members can control the network.

- Another application of being able to obtain all MCSs is creating a broker to influence the network toward a targeted behavior by applying intervention to a network structure. For example, such a broker can be used to tailor friendship suggestions in social networks to weaken the polarization phenomenon in these networks.

A broader significance of this work is its contribution in understanding the control characteristics of complex systems. For instance, studying the ways these systems self-organize and control themselves through interactions with their environments. As mentioned above, many complex systems operate in a distributed or a centralized control regime [13, 14]. The findings in this research contribute in formalizing and understanding these control regimes. Furthermore, the developed methods in this research could be extended to switch between these control regimes by applying targeted interventions in the structure of networks. It is this process that is likely to change the nature of networks over time.

1.3 Research Questions

The proposed research attempts to answer the following questions:

1) Is it possible to create an algorithm to find minimal-sized driver node sets that can make temporal networks structurally controllable?

2) Can an algorithm be designed to determine all MCSs of nodes in temporal networks?

3) If the above is true, does the analysis of all MCSs reveal the overall behavior of physical systems modeled with temporal networks?

1.4 Limitations

- For temporal networks, the analytical approaches to find the Minimum Driver node Sets (MDSs) are on the order of $O(2^N)$ and currently such sets can be identified only by brute-force. Hence, except for very small networks, it is not always feasible to find an MDS of temporal networks. Therefore, this work may pursue heuristic approaches to identify suboptimal MDSs when the goal is to achieve full controllability.

1.5 Assumptions

The following assumptions are considered in this project:

- This work focused on systems that are governed by discrete time-varying linear dynamics [5, 15]. Most real systems are driven by nonlinear dynamics, but their controllability is structurally similar to linear systems in many aspects [16].
- This work used the definition of structural controllability for temporal networks proposed by Pósfai and Hövel in [5].

1.6 Delimitations

The following delimitations are applied to this study:

- Six temporal networks of ant colonies [17] and e-mail communication of a manufacturing company [18] are incorporated to conduct analysis and perform evaluations. The characteristics of these datasets enable performing computationally feasible experimental trials on temporal networks that have a variety of sizes and behaviors.
- The analysis of computational complexity for the proposed algorithms are relaxed based on the empirical results.
- The correctness of proposed methods in this research is based on both analytical proofs and empirical evidences.

2. REVIEW OF RELEVANT LITERATURE

This section presents a brief introduction of network science, time-aggregated networks, temporal networks, and the structural controllability concepts. An overview of recent advancements in the structural controllability of complex networks is discussed. Also, a brief overview of the author's research in the domains of complex systems and network science is provided.

2.1 Complex Systems Theory and Network Thinking

Most systems consist of many parts that are working together to form capabilities and functionalities. For example, languages consist of distinct components such as verbs and nouns that form a system that we use to communicate our thoughts and ideas with each other. Studying the individual parts that create a system is immensely important for understanding systems. On the same token, studying the collective behaviors of a system is an equally important aspect to understand the emergent behaviors of a system, that is, the phenomena that emerge from interactions between the individual parts of a system [19]. For instance, consider language as a complex system that is evolved over thousands of years. Why do some grammatical structures lose their popularity over time, but others become dominant? To answer these complex questions, we would need to analyze systems as a whole and study the relationships and interactions between their parts. The complex systems theory offers a framework to conceptualize, study, and evaluate the emergent behaviors of systems [19]. In other words, scientists view systems' functions as a set of phenomena and they try to discover and explain the mechanisms that drive the emergence of those phenomena. Hence, the complex systems theory provides a holistic perceptive to study and analyze a system of interest. For example, Will et al. [20] treated the

circulatory system as space-filling fractal networks, and they introduced a theory that provides an explanation for the well-known metabolic scaling relationship observed in many species of animals (i.e., $\textit{metabolic-rate} \propto \textit{body-mass}^{3/4}$).

Interconnectivity, scalability, robustness, and fractal geometry are a few common themes within complex systems [19]. Network thinking provides new perspectives to analyze and understand complex systems by capturing these themes in modeling. A complex network is a web of often many interconnected entities with topological characteristics that are non-trivial and do not happen in simple networks such as lattices and random networks. Scientists widely use complex networks to characterize many physical and phenomenological phenomena in natural and human-made systems. Network thinking is applied to a variety of scientific disciplines including applied and abstract physics [6], chemistry [8], sustainability in ecosystem management [21], and biology [9, 10].

Based on the complex systems theory and network science, the author conducted research in various domains. In [22], the author introduced the Block Software Defined Storage (BSDS) framework to enable self-organization and self-adaptation in cloud block storage infrastructure. To understand the political polarization in social networks, in [23] the author developed an agent-based model to investigate scenarios that either increase or decrease levels of polarization in social networks. Moreover, the author proposed strategies to decrease the polarization issue in social networks. On the topic of food security, in [24] the author proposed an agent-based model to simulate human consumption behavior in all-you-can-eat facilities and quantify the impact of plate size on food waste. The results showed reducing plate size can decrease food waste by 30% without impacting the quality of service. In [25], the author proposed a network-based approach to model and analyze the entire cardiovascular tree as a complex system. The author showed how network science can provide a new perspective to look at the coronary angiography images. Lastly, in [26] the author introduced a heuristic approach to control temporal networks.

2.1.1 Temporal Networks

To create a network from a physical system, we observe the system during multiple observational intervals and record the interactions between its components. Depending on the frequency of observational intervals taken from a system, two types of networks can be introduced: a) time-aggregated (also called static) networks, and b) temporal networks. Time-aggregated networks capture the existence of interactions (or connections) between components of a system, and they ignore the temporal correlations by aggregating the observed interactions. Therefore, time-aggregated networks are suitable for modeling complex systems with a fixed topology, which are governed by trivial temporal correlations such as the power grid networks and ecological networks [27, 28].

However, by nature many systems change continuously and the effect of changes is non-commutative; it is essential to study the effect of these changes on the controllability of networks. Temporal networks explicitly represent the order of observed interactions in the structure of networks, which is a necessity when a system is governed by non-trivial temporal correlations. For example, the speed of spreading processes depends on the temporal correlations [29,30]. Such complex systems with non-trivial temporal correlations range from systems of communication and transportation to biological systems such as neural circuits [31–34].

To capture the dynamic changes in topology of networks, the temporal correlations of links must be explicit. The author illustrates an example in Fig. 2.1 by a simple temporal network with three time-respecting paths (marked with distinct colors) in observational interval $1 \leq t \leq 2$. A *time-respecting path* is a path that follows the temporal orders of connections. In Fig. 2.1 (a), the time-aggregated network allows information to spread from node v_1 to node v_4 through $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$ path. However, considering the order of interactions, it is not physically possible to have a path from node v_1 to node v_4. Hence, the time-layered representation can be utilized to visualize the temporal correlations as presented in 2.1 (b). The number of layers in

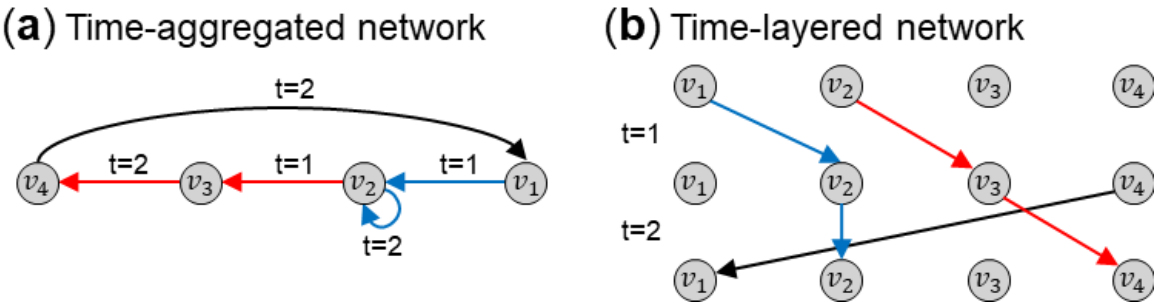

Fig. 2.1. Temporal correlations in pathways.

a time-layered network is equal to the number of observational intervals and each layer has a copy of network nodes. The links between time-layers represent the interactions between nodes observed within an interval. Hence, time-layered networks explicitly encode the temporal correlations between interactions.

This research focuses on the controllability of temporal networks. Many studies in this area are centered on networks with a fixed topology [35–38]. In contrast, this work shifts focus from the fixed topology to the structural controllability of temporal networks. The structural controllability theorem shows that many control related questions can be answered by only analyzing the structure of networks [1, 2, 39]. In [40], Li et al. showed how temporal networks have fundamental advantages concerning the energy and time needed to achieve control. They presented how to harvest the changing topology of complex systems to improve the efficiency of control. Furthermore, in [5], Pósfai and Hövel introduced the fundamental tools to analyze the structural controllability of temporal networks.

2.2 Controllability of Time-Aggregated Networks

A time-aggregated (i.e., static) network denoted by graph $G(V, E)$ consists of a set of vertices (i.e., nodes) $V = \{v_1, v_2, ...v_n\}$ and a set of edges $E = \{e_1, e_2, ...e_m\}$.

Each edge $e = (v_i, v_j, w) \in E$ is a triplet connecting two vertices with weight w (i.e., strength of connection). For directed networks the triplets are ordered (i.e., $v_i \rightarrow v_j$), whereas in undirected networks the edges have no order (i.e., v_i — v_j). An adjacency matrix $A \in \mathbb{R}^{N \times N}$ is the algebraic representation of networks describing the *wiring* of a system between N vertices (components) and their strength of interactions $w_{ij} \in A$. Time-aggregated networks make the assumption that patterns of connections are fixed over time (i.e., the topology of network is fixed).

The structural controllability of time-aggregated networks is applied to the systems that are modeled by a set of nodes with time-varying state variables. A set of directed edges captures the relationship between these nodes. Also, a protocol governs the dynamics of interactions between the nodes that changes the nodes state variables over time. These systems receive stimuli by $u(t)$ from an environment, and they produce outputs. To conclude, the structural controllability of complex networks studies the behaviors and relationships between the structural connections and external stimuli.

Nonlinear dynamics govern many real systems, however, in many aspects, their controllability is structurally similar to that of linear systems [16]. Therefore, recent studies mostly focused on the linear time-invariant dynamics (LTI) that govern a system as presented in Equation 2.1 [2]. In this equation, the network of N vertices is represented by its adjacency matrix A. Each vertex holds a variable x of interest that varies with time, called its state. The network is connected to the external world from a subset of its vertices (captured by matrix B) through time varying stimuli captured by vector u.

$$x(t+1) = A^{\top} x(t) + Bu(t) \tag{2.1}$$

Time-varying vector $x(t) \in \mathbb{R}^N$ captures the state of networked system at interval t. The effect of external stimuli is modeled by the second part of Equation 2.1. The matrix $B \in \mathbb{R}^{N \times M}$ describes the connections between a controller and M driver nodes. The controller imposes stimuli to the network by controlling the driver nodes,

and the time-varying input vector $u(t) \in \mathbb{R}^M$ captures the strength of these stimuli. Therefore, reaching control depends on identifying a set of driver nodes that can fully control the network if driven by external stimuli. Also, many control problems are interested in identifying a Minimum Driver node Set (MDS) with its size denoted by N_c.

The system in Equation 2.1 is controllable if one can identify a set of driver nodes capable of steering the state of network from an initial state to a target state over a finite time. According to Kalman's rank condition, control is possible if the control matrix $C \in \mathbb{R}^{N \times NM}$ has full rank, i.e., $rank(C) = N$ in Equation 2.2 [41].

$$C = (B, AB, A^2B, ...A^{N-1}B) \tag{2.2}$$

To apply Kalman's rank condition the weights of all links must be known, but for many real-world networks this is not practical. Fortunately, the a priori requirement to have exact weights of links can be bypassed by the structural controllability theorem. That is, assuming Equation 2.1 governs the system's dynamic, the system $< A, B >$ is *structurally controllable* if by replacing nonzero weights in A and B the controllability matrix becomes full rank. Hence, the system's links needs to be represented by nonzero entries in $< A, B >$. In [1], Lin showed that a structurally controllable system is controllable for almost any configuration of weights. However, we need to test Kalman's rank condition on $O(2^N)$ driver node sets to find an MDS. Moreover, the problem of identifying an MDS is reduced to the maximum matching problem on directed networks [42]. Hence, an MDS can be identified with $O(\sqrt{N}E)$ computation where N is the number of nodes, and E is the number of edges.

2.3 Controllability of Temporal Networks

Temporal networks enable modeling systems with changing topology over time. Also, the efficiency of control (regarding time and energy) can be increased by utilizing the changes in system's wirings [40]. Moreover, including time in the definition

of structural controllability enables to explicitly study the time necessary to reach controllability. However, for the time-aggregated networks, the study of time needed to reach control only receives implicit treatment [43]. The aforementioned motivate the study of controllability on temporal networks.

Another significant characteristic of complex systems that is traditionally ignored in time-aggregated networks is modeling self-interactions [44, 45]. Often natural systems benefit from a passive stability that enables the capacity to retain information over time [46, 47]. The framework introduced by Pósfai and Hövel in [5] enabled the use of self-loops to model the state retention aspect. The author illustrates an example of state retention in Fig. 2.1, where the single self-loop enabled node v_1 to influence node v_2 at interval $t = 2$.

A directed temporal network denoted by $T(V, E)$ consists of a set of vertices $V = \{v_1, v_2, ...v_n\}$ and a set of directed temporal edges $E = \{e_1, e_2, ...e_m\}$ with timestamps. Each temporal edge $e = (v_i, v_j, w_{ij}, t) \in E$ connects two vertices $v_i \rightarrow v_j$ at observational interval t and the strength of connection is represented by weight w_{ij}.

To conceptualize control, the dynamics that govern the interactions between nodes and capture the changing topology of temporal networks are needed to be defined. This study assumes the system is governed by discrete time-varying linear dynamics as shown in Equation 2.3 [5,15]. Although nonlinear processes drive many real-world systems, the linear systems can provide good approximations [2, 16].

$$x(t+1) = A^{\top}(t)x(t) + B(t)u(t) \tag{2.3}$$

The time-varying vector $x\,(t) \in \mathbb{R}^N$ captures the state of nodes and N denotes the number of nodes. The state of network at interval $t+1$ is defined by Equation 2.3 as a sum of two terms: a) the effect of system's internal dynamics applied at interval t, and b) the effect of external stimuli applied to driver nodes. The weighted adjacency matrix $A\,(t) \in \mathbb{R}^{N\times N}$ encodes system's wiring at interval t. A controller stimulates driver nodes $d_i \in D \subseteq V$ at interval t, which propagates the injected signal values to

interval $t+1$. In this research, an *intervention point* is denoted by the pair $(d_i, t+1)$ to indicate stimulating a driver node. Also, the matrix $B(t) \in \mathbb{R}^{N \times N_I(t)}$ captures the intervention points where $u(t) \in \mathbb{R}^{N_I(t)}$ represents the strength of stimuli and function $N_I(t)$ yields the number of intervention points at t.

2.3.1 Independent Path Theorem

Assuming Equation 2.3 governs the dynamics of a system, based on the independent path theorem a temporal network is structurally controllable within a desired Δt steps at deadline (target) t_f if there exists $|V|$ number of independent time-respecting paths originated by intervention points and end on all nodes $v_i \in V$ at t_f [5]. Independent time-respecting paths are a set of paths that do not cross each other on any time layers within a given interval (i.e., vertex disjoint paths). An example is presented in Fig. 2.2, where the independent time-respecting paths are marked with the solid red edges. In Fig. 2.2 (a), the deadline to reach control is $t_f = 1$ and at least three driver nodes are required to gain control. But, in Fig. 2.2 (b), to reach control at $t_f = 2$ two driver nodes are required. Also, the self-loop of node v_2 enables state retention by which node v_1 influences v_2 over interval $t = 2$.

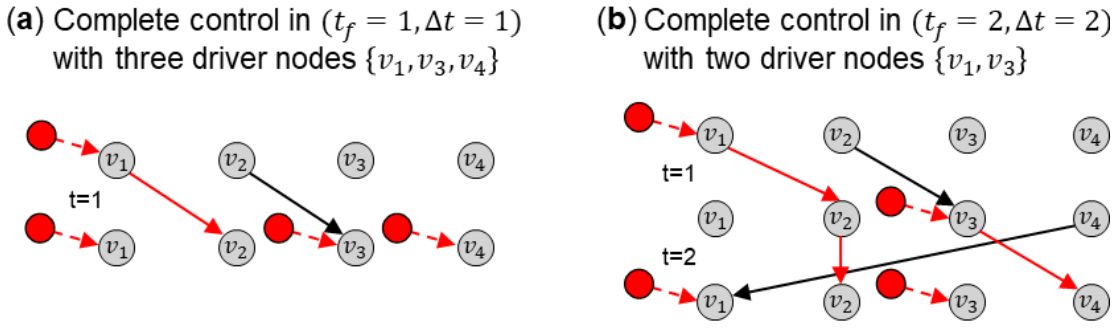

Fig. 2.2. Controlling a simple temporal network.

2.3.2 Maximum Controllable Subspace

The Maximum Controllable Subspace (MCS) is denoted by $M_C(D, t_f, \Delta t) \subseteq V$ that identifies two properties of driver node set $D \subset V$: 1) the maximum number of nodes, namely N_m, that can be controlled by D, and 2) a set of nodes with cardinality N_m that are controllable by D (i.e., an MCS of D) [5]. We can efficiently find an MCS using Ford-Fulkerson algorithm by finding the maximum flow in the time-layered

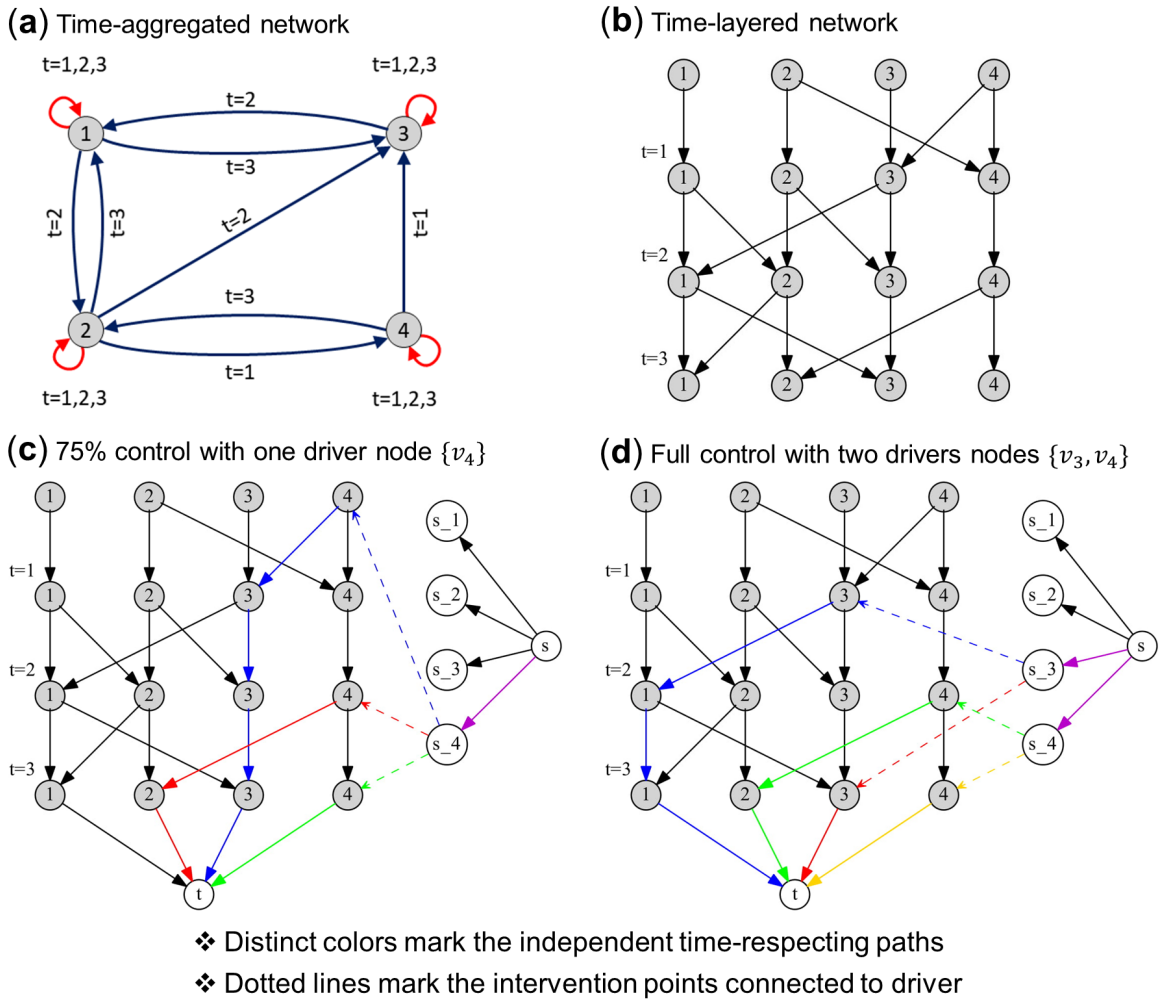

Fig. 2.3. Identifying a maximum controllable subspace.

networks [48, 49]. The procedure of finding an MCS is: 1) create a source node and connect it to all drive nodes in D within the time layers, 2) create a sink node and only connect all node in the deadline layer to it, and 3) limit the flow of each node to one (i.e., at most a flow of one can pass through each node). Hence, a driver node set can have multiple MCSs since a network can have multiple maximum flows. The author provides an example in Fig. 2.3 where the identified independent time-respecting paths are marked by distinct colors. The colored edges are the edges with non-zero flow, which are identified using a maximum flow algorithm. Also, to improve visualization, the connections from source node s to the driver nodes are grouped, and the intervention points are presented with dotted edges. Driver node set $D_1 = \{v_4\}$ can control at most 75% of the network as illustrated in Fig. 2.3 (c). However, as illustrated in Fig. 2.3 (d) driver set $D_2 = \{v_3, v_4\}$ can reach full control.

2.4 Control Characteristics of Static Networks

Nodes in a network can be characterized based on their capacity and level of contribution to control the network. This section aims to provide a summary of the literature on characterization of nodes and networks based on the controllability of static networks.

By considering the concept of controlling a network using the minimum number of driver nodes, it is possible to categorize nodes based on their intersections with multiple MDSs. As mentioned earlier, the problem of identifying a single MDS in static networks is reduced to the maximum matching problem [2, 42]. Equally important, most networks have many MDSs (i.e., multiple maximum matchings) in which nodes intersect or disjoint with. In [14], Jia et al. build upon the existence of multiple MDSs and introduce three categories of nodes: a) *critical nodes* that belong to all MDSs (always selected as drivers), b) *intermittent nodes* representing the nodes that at least appear in one MDS, and c) *redundant nodes* that are not selected as a driver in any MDS. Furthermore, they discovered a bimodality in the fraction of

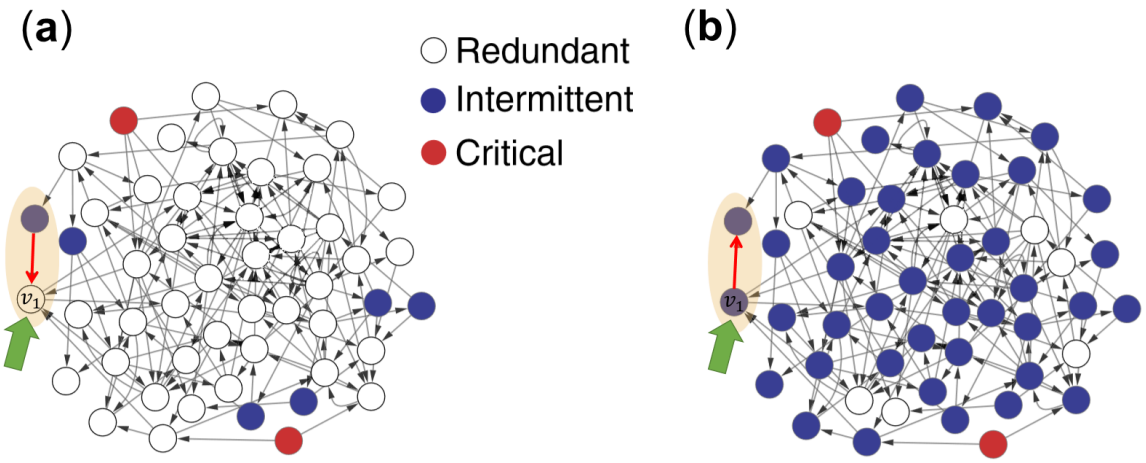

Fig. 2.4. Control categories of nodes in static networks (Courtesy Jia et al. [14]).

redundant nodes in complex networks. This bimodality signals the existence of two control regimes, namely the *distributed* and *centralized* regimes based on the tendency of nodes to appear in MDSs. When only a small fraction of nodes can form an MDS, the network can be considered to operate under a centralized control regime; whereas a distributed control regime indicates that many nodes can form an MDS. For example, the author presented two static networks with different control regimes in Fig. 2.4. As illustrated, by changing the direction of a single edge, the control category of node v_1 (marked with the green arrow) changed from redundant to intermittent.

Furthermore, Jia et al. [14] proposed that if $\Delta n_r = n_r - n_r^T > 0$ then a network operates under a centralized control regime, and if $\Delta n_r < 0$ then it operates under a distributed regime where n_r denotes the number of redundant nodes and n^T refers to the transpose of a network in which all the edges are reversed.

2.5 Summary

The controllability of complex networks, and particularity temporal networks, is an emerging field. This chapter introduced the relevant literatures in the field and provided the background needed for the coming chapters. To summarize, this chapter introduced the following points: 1) the definition of controllability for both time-aggregated and temporal networks, and 2) the past attempts to reveal the behavior of complex networks from the controllability perspective.

3. FRAMEWORK AND METHODOLOGY

The previous chapter summarized the current achievements in the literatures and introduced the innovation and general scope of this research. In this chapter, the author dives into the methodologies and frameworks used to develop this study.

3.1 Approach

This research consists of three objectives and quantitative approaches that are perused. Objective I aims to build driver node sets that can make a temporal network structurally controllable. In objective II, the researcher aims to design an algorithm to identify all MCSs of a driver node. Objective III focuses on analyzing the MCSs of individual driver nodes to investigate the overall behavior of temporal networks from the controllability perspective.

3.1.1 Objective I: Create Minimum Driver Node Sets

As discussed in Chapter 2 finding an MDS for temporal networks is computationally prohibitive. Hence, the researcher proposes a heuristic approach for identifying and using driver nodes in temporal networks. The heuristic algorithm creates multiple Suboptimal Minimum Driver node Sets (SMDSs) with size $N_s \geq N_c$ with the computation complexity of $O(N^3 + N^2E + N\Delta t)$. The concept of SMDS focuses on efficiently creating a set of drivers that can fully control the network. This research builds on the concept of SMDS to analyze the overall behavior of temporal networks from the controllability perspective.

3.1.2 Objective II: Identify All MCSs of Nodes in Temporal Networks

The researcher proposes the below algorithmic approach to identify multiple MCSs of a single node, denoted by u. Function $M_c(G, u)$ finds an MCS of u in time-layered network G using the maximum flow approach that was introduced in Section 2.3.2. Chapter 5 draws on the below algorithm to develop an approach capable of identifying all MCSs of a driver node.

1. Find an MCS of node u denoted by $M = M_c(G, u) \subseteq V$ in the time layered network G where V is the set of network nodes.
2. Create the set $\mathcal{M} = \{M\}$ to store all MCSs of u.
3. For every node $v \in M$ create an augmented graph G^v where v is removed from deadline layer t_f.
4. find an MCS of u denoted by $M' = M_c(G^v, u)$ and if $|M'| = |M|$ then record M' is a new MCS of u, i.e., $\mathcal{M} = \mathcal{M}.append(M')$.
5. Repeat steps 3-4 until all nodes in $v \in m$ are exhausted.
6. Return $\mathcal{M}$ which contains multiple MCSs of node u.

The above algorithm enables finding the intersection and union between MCSs of a single node u, denoted by $I^u = \bigcap_{M \in \mathcal{M}} M$ and $U^u = \bigcup_{M \in \mathcal{M}} M$ for node $u \in V$. The goal is to analyze I and U for all nodes in temporal networks and investigate whether they follow a pattern. One way to analyze all MCSs of nodes is using metrics such as ranking the nodes with respect to I and U. For example, we can rank nodes based on their frequency in MCSs of the other nodes. By using similar approaches, this research aims to explore the domain of information that can be provided using the above algorithm.

Novelty of Identifying Multiple MCSs

The proposed algorithmic approach leads to the creation of a new concept in the controllability of temporal networks. The author refers to this concept as the Complete Controllable Domain (CCD) for a driver node set. The CCD of nodes reflects the degree to which the system's entities are capable of influencing control. Hence, the third objective of this research focuses on identifying the properties of CCD.

3.1.3 Objective III: Analysis of the Controllable Space of Nodes from the Controllability Perspective

The author defines CCD as the union of all MCSs of a driver node, and the goal of this objective is exploring the characteristics of CCD of individual driver nodes. An approach toward this exploration is investigating the intersection between MCSs of nodes and the driver node sets (i.e., MDSs and SMDSs as identified in Chapter 4). From the controllability perspective, this analysis is likely to reveal the overall behavior of temporal networks. Moreover, the possible findings from exploring all MCSs of nodes could contribute to improving the formalization of the CCD concept and identifying its potential applications.

3.2 Novelty

The novelty of this research is in providing efficient algorithms to study the behavior of temporal networks from the structural controllability perspective. This research introduces the concept of SMDS to efficiently identify driver node sets that can make a temporal network controllable. Next, the author introduces the concept of CCD for a driver node by designing an algorithm that can identify k MCSs of a driver node. The concepts of SMDS and CCD could be used to determine the function of a system

from its structure, which has a great importance in understanding the underlying mechanisms that govern complex systems.

3.3 Datasets

This research performs a series of experiments on real-world temporal networks that are induced from ants' interactions [17] and e-mail communications of a manufacturing company [18]. Six temporal networks created from independent ant colonies are utilized to develop methods, conduct analysis, and perform evaluations. Blonder and Dornhaus [17] captured the interaction between ants of six colonies and provided timestamped directed edges, which record the ants who initiate pairwise communications by touching antennas. The characteristics of these datasets (e.g., a small number of minimum driver nodes) enable performing various experiments that are computationally feasible. Each ant network has a unique temporal and structural characteristics as presented in Fig. 3.1 and Fig. 3.2. For example, ant colonies 1-1 and 1-2 have a low average inter-event (1.63 and 1.67 seconds respectively), and colonies 3-1 and 3-2 have a large average inter-event (15.87 and 15.83 seconds).

The main reason for selecting these datasets is the nature of physical systems that they represent. The ant colonies are capable of self-organization; that is, an ant colony does not depend on a leader to organize itself [19]. In contrast, the emails dataset reflects the structure of communications in a hierarchical organization that relies on leaders to be managed. In a hierarchical organization, it is common to observe that a few numbers of employees (e.g., managers) are the most influential components in managing the organization. Hence, the selected datasets enable to analyze the proposed algorithms in this dissertation on systems with both self-organizing and hierarchical traits.

(a) Ants 1-1

Number of Nodes:	89
Time-stamped links:	1911
Links ÷ Nodes:	21.47
Time-layers:	883
Observation period:	[0 s, 1438 s]
Observation length:	1438 s
Avg. inter-event dt:	1.63 s
Min/Max inter-event dt:	1/15 s

(b) Ants 1-2

Number of Nodes:	72
Time-stamped links:	1820
Links ÷ Nodes:	25.28
Time-layers:	1048
Observation period:	[0 s, 1749 s]
Observation length:	1749 s
Avg. inter-event dt:	1.67 s
Min/Max inter-event dt:	1/12 s

(c) Ants 3-1

Number of Nodes:	11
Time-stamped links:	78
Links ÷ Nodes:	7.09
Time-layers:	72
Observation period:	[12 s, 1139 s]
Observation length:	1127 s
Avg. inter-event dt:	15.87 s
Min/Max inter-event dt:	1/97 s

(d) Ants 3-2

Number of Nodes:	6
Time-stamped links:	104
Links ÷ Nodes:	17.33
Time-layers:	92
Observation period:	[2 s, 1425 s]
Observation length:	1423 s
Avg. inter-event dt:	15.64 s
Min/Max inter-event dt:	1/533 s

(e) Ants 6-1

Number of Nodes:	33
Time-stamped links:	652
Links ÷ Nodes:	19.76
Time-layers:	537
Observation period:	[1 s, 1918 s]
Observation length:	1917 s
Avg. inter-event dt:	3.58 s
Min/Max inter-event dt:	1/34 s

(f) Ants 6-2

Number of Nodes:	32
Time-stamped links:	367
Links ÷ Nodes:	11.47
Time-layers:	312
Observation period:	[2 s, 1755 s]
Observation length:	1753 s
Avg. inter-event dt:	5.64 s
Min/Max inter-event dt:	1/58 s

(g) E-mail Communications of a mid-sized manufacturing company

Number of Nodes:	109
Time-stamped links:	250
Links ÷ Nodes:	2.30
Time-layers:	224
Observation period:	[1262671200 s, 1285758000 s]
Observation length:	23086800 s
Avg. inter-event dt:	103528.25 s
Min/Max inter-event dt:	3600/500400 s

Fig. 3.1. Network characteristics of the datasets.

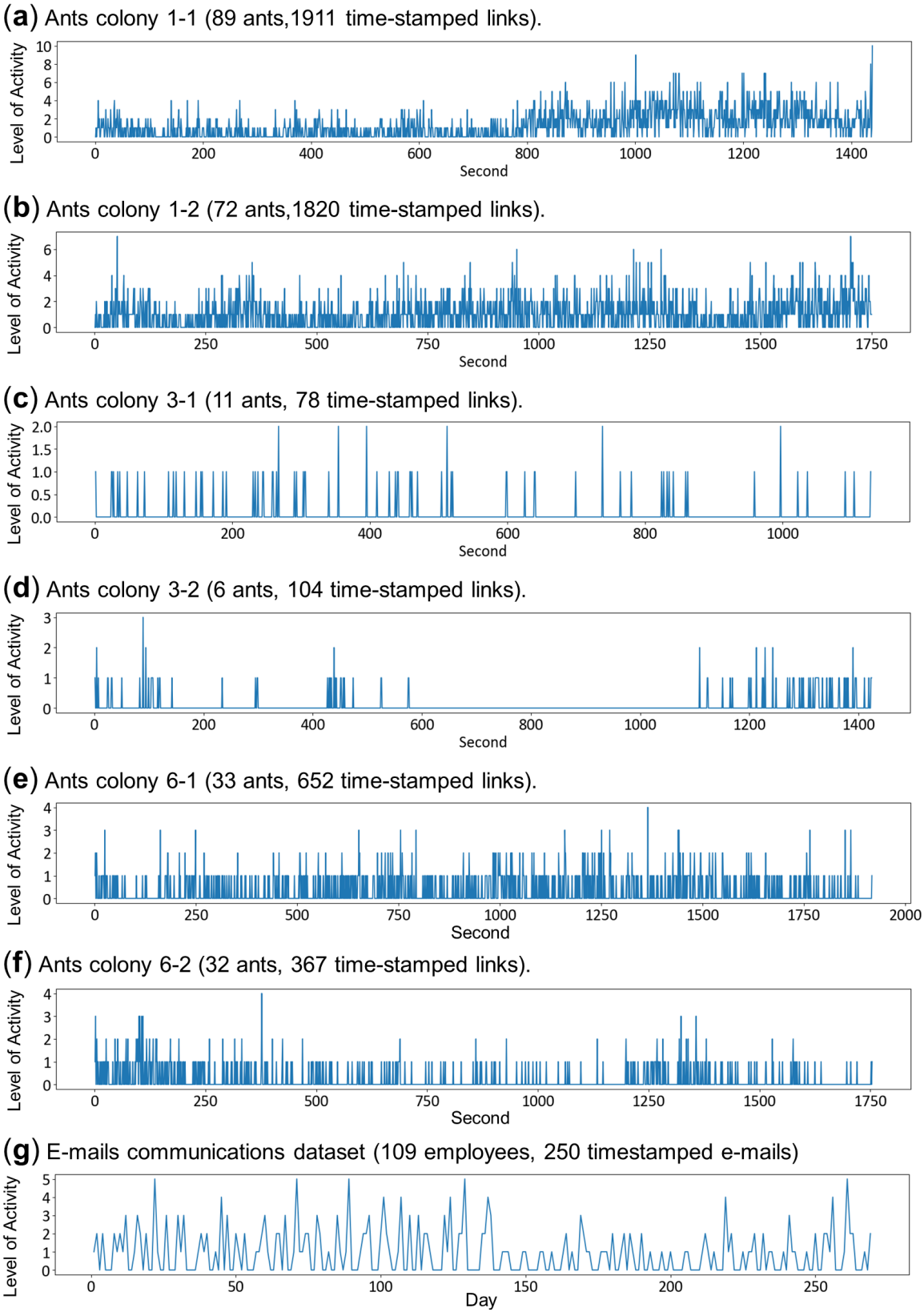

Fig. 3.2. Frequency of interactions in the datasets.

3.4 Measure of Success

The measures of success are as the following:

- Develop an approach to build driver node sets that can make a temporal network structurally controllable.
- Design an algorithm to determine all MCSs of a driver node.
- Analyze the controllable space of individual driver nodes and draw intuitions using the methods and metrics as discussed in Objectives I, II, and III.
- From the controllability perspective, connect the analysis of MCSs with the characteristics of the physical systems represented by the aforementioned datasets in Section 3.3. In other words, analyze the controllable space of nodes to understand the overall behavior of the temporal networks created from those datasets.

3.5 Summary

This chapter introduced the research framework and objectives of the proposed study to answer the proposed research questions in Section 1.3. Also, the datasets used in this study are introduced along with the measurements of success.

4. IDENTIFYING DRIVER NODES FOR TEMPORAL NETWORKS

Finding an MDS for temporal networks is computationally prohibitive due to two difficulties: a) it involves searching many combinations of driver nodes and intervention points in order of $O(2^{N\Delta t})$ configurations where $\Delta t = t_f - t_0$ represents the expected number of steps to reach structural controllability at deadline t_f, and b) the rank of temporal controllability matrix $C(t_0, t_f)$ in Equation 4.1 needs to be calculated to test a configuration. Also, the latter step requires $O(A(t)^{\Delta t})$ number of matrix multiplications. This research assumes that the start interval is $t_0 = 0$. There is no loss of generality as we can modify the start interval to gain control within any desired period in $(t_f - \Delta t, t_f]$.

$$\begin{aligned} C(t_0, t_f) = \Big[& A(t_f - 1)A(t_f - 2) \dots A(t_0 + 1)B(t_0); \\ & \dots; A(t_f - 1)B(t_f - 2); B(t_f - 1) \Big] \end{aligned} \tag{4.1}$$

The operator $[X; Y]$ concatenates two matrices hence $C(t_0, t_f) \in \mathbb{R}^{N \times N_I}$ where $N_I = \sum_t N_I(t)$ returns the total number of interventions. Assuming the discrete time-varying linear dynamics (Equation 2.3) governs the system, by a proper selection of intervention points the system $< A(t), B(t) >$ becomes controllable if $rank(C(t_0, t_f)) = N$. Since, the linear rank of $C(t_0, t_f)$ represents the number of variables that can be independently controlled.

Constructing an MDS with size N_C that passes the aforementioned rank condition is computationally excessive as it involves minimizing the number of driver nodes and maximizing the number of times the drivers are used. Hence, the author proposes a heuristic algorithm to efficiently find Suboptimal Minimum Driver node Sets (SMDSs) with size $N_s \geq N_C$ that can fully control temporal networks.

4.1 Heuristic Approach for Identifying SMDSs

Pósfai and Hövel proposed the independent path theorem [5] to formalize the structural controllability for temporal networks. That is, a subset of network nodes $C \subseteq V$ at deadline t_f is controllable within Δt steps iff there exists $|C|$ independent time-respecting paths such that a) they start from intervention points, and b) they end on all nodes in C at t_f layer. Hence, a temporal network is structurally controllable if there exists $|V|$ number of independent time-respecting paths originated from intervention points and end on t_f layer. Hence, the author proposes a heuristic algorithm to find $|V|$ number of such independent-paths. This is equivalent with identifying an SMDS, denoted by D, where its MCS covers all networks nodes, i.e., $M_C(D, t_f, \Delta t) = V$ such that $M_C \subseteq V$ identifies an MCS of D within $t_f - \Delta t < t \leq t_f$. This leads to the emergence of Properties 1 and 2 that connect the concept of MCS to the MDS and SMDS concepts.

Property 1 $\sum_{d \in MDS} |M_C(\{d\}, t_f, \Delta t)| \geq |V|$ *for any arbitrary MDS.*

Property 2 *For any driver set* $D' \subseteq V$ *if* $\sum_{d \in D'} |M_C(\{d\}, t_f, \Delta t)| \geq |V|$*, then* D' *is a potential SMDS.*

The heuristic approach transforms a sorted list, denoted by $\mathbb{C}$, that contains an MCS of each node to an SMDS. This transformation is based on applying Property 2 to $\mathbb{C}$. Formally, a set S_{v_k} denotes an MCS of v_k such that $S_{v_k} = M_C(\{v_k\}, t_f, \Delta t) \subseteq V$. The author proposes to create a possible SMDS, denoted by D', from an MCSs-list $\mathbb{C}$ in the following steps:

1. Create an MCSs-list such that $\mathbb{C} = [S_{v_1}, S_{v_2}, \ldots, S_{v_{k=N}}]$; that is for each node store an arbitrary MCS.

2. Sort $\mathbb{C}$ in the decreasing order of its elements size (the size of MCSs). That is, $|S^1| \geq |S^2| \geq \cdots \geq |S^N|$ for every $S_{v_k} \in \mathbb{C}$ where the position of elements in $\mathbb{C}$ are denoted by the superscripts.

3. Starting from the first MCS $S^{i=1}$, do $S^j \leftarrow S^j \setminus S^i$ for $1 \leq i < j \leq N$.

4. Repeat Step 3 for $i = 2 \ldots N$.

5. Consider the nodes with non-empty controllable subspace as potential driver nodes. That is, a potential SMDS denoted by $D' = \{v_k | S_{v_k} \in \mathbb{C}, S_{v_k} \neq \emptyset\}$ is derived from $\mathbb{C}$.

6. Mark D' as an actual SMDS, denoted by $D = D'$ if $M_C(D', t_f, \Delta t) = V$. Otherwise, D' cannot fully control the network hence it is not an SMDS.

The author illustrates an execution of the proposed heuristic in Fig. 4.1 on a simple temporal network. The step by step execution of heuristic approach on MCSs-list $\mathbb{C}$ is presented in Fig. 4.1 (a) and resulted in transforming $\mathbb{C}$ to an SMDS $\{v_1, v_4\}$. Moreover, Fig. 4.1 (b) illustrates the independent time-respecting paths originated from the identified driver nodes (marked with red color). Also, the identified SMDS is an MDS since at least two driver nodes are required to fully control the network.

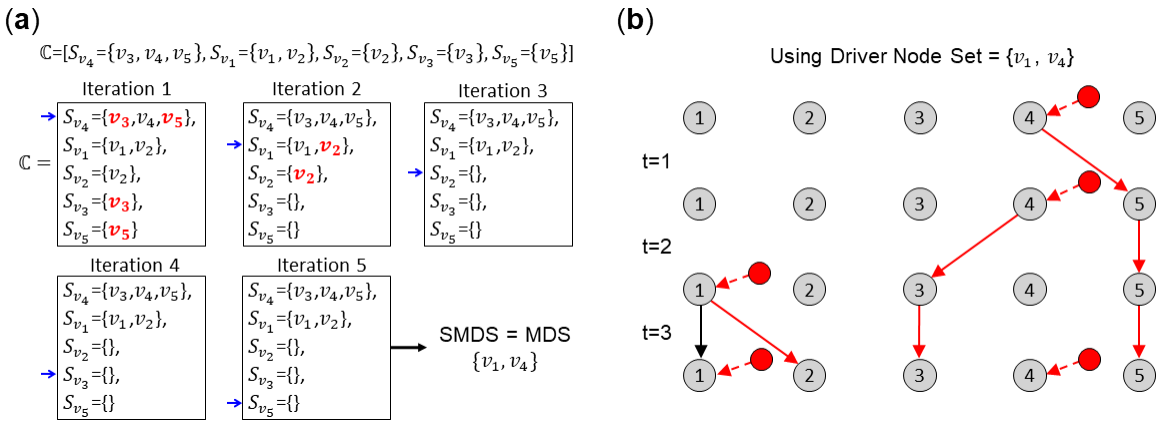

Fig. 4.1. Heuristic approach execution on a temporal network.

4.1.1 Multiple Controllable Subspaces and Branching Process

Temporal pathways limit the number of independent time-respecting paths, especially when multiple driver nodes are selected. This limitation results in the following inequality $M_C(\{v_1, v_2\}, t_f, \Delta t) \neq M_C(\{v_1\}, t_f, \Delta t) \cup M_C(\{v_2\}, t_f, \Delta t)$, i.e., an MCS of multiple driver nodes is not equal to the union of an MCS of each driver node separately. Moreover, a node does not necessarily have a unique MCS due to the existence of various temporal pathways. The author illustrates an example in Fig. 4.2 by adding a new temporal link (marked with color blue) to the temporal network of Fig. 4.1 (b). All possible MCSs of each node are presented in Fig. 4.2 by S_{v_i}. The new link expands the space of MCSs of nodes v_4 and v_5 hence the maximum flow approach could return any of the possible MCSs. For instance, function $M_C(v_5, t_f = 3, \Delta t = 3)$ could return any of the two MCSs of node v_5 as illustrated in Fig. 4.2. Suppose $M_C(v_5, t_f = 3, \Delta t = 3) = \{v_2, v_5\}$ and $M_C(v_4, t_f = 3, \Delta t = 3) = \{v_1, v_3, v_4\}$. Now, accounting for the temporal path-

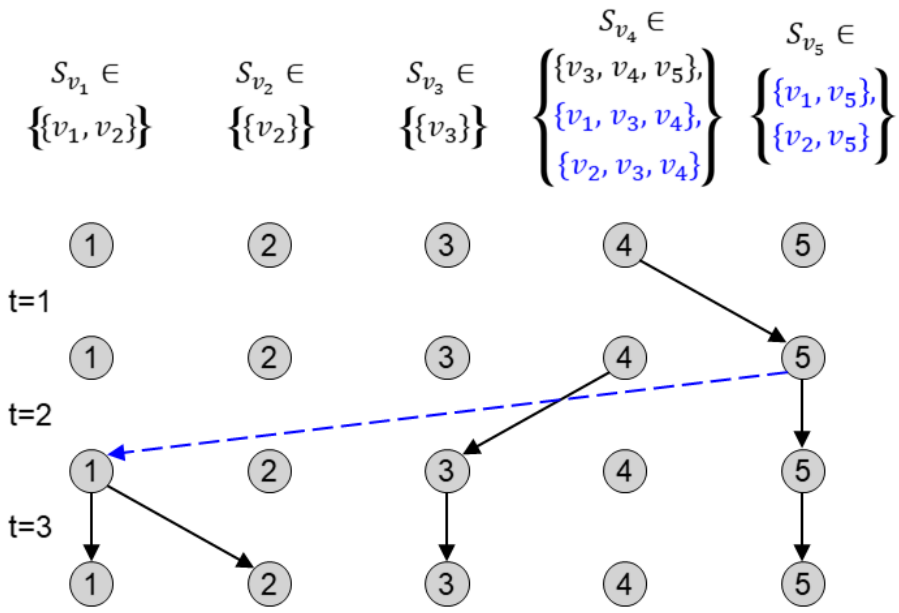

Fig. 4.2. Non-unique MCSs.

ways $M_C(\{v_5, v_4\}, t_f = 3, \Delta t = 3) \in Z = \{\{v_1, v_3, v_4, v_5\}, \{v_2, v_3, v_4, v_5\}\}$. However, $\{v_2, v_5\} \cup \{v_1, v_3, v_4\} \notin Z$.

The aforementioned inequality can affect the size of an SMDS generated by the heuristic approach since the heuristic does not have any control on which MCS is selected by a maximum flow algorithm. Hence, the author introduces a branching procedure to generate multiple SMDSs, which increase the chance of finding at least one MCS. Also, having multiple SMDSs can be used to characterize the overall behavior of temporal networks from the controllability perspective as discussed in Chapters 6 and 7. This procedure creates multiple stems of the original MCSs-list in the following way:

1. On each iteration of the heuristic if $S^i \cap S^j \neq \emptyset$ for $j > i$ then create a new stem $\mathbb{C}' \leftarrow \mathbb{C}$.

2. Swap the position S^i and S^j in $\mathbb{C}'$.

3. Run the heuristic approach on $\mathbb{C}'$ without creating new stems.

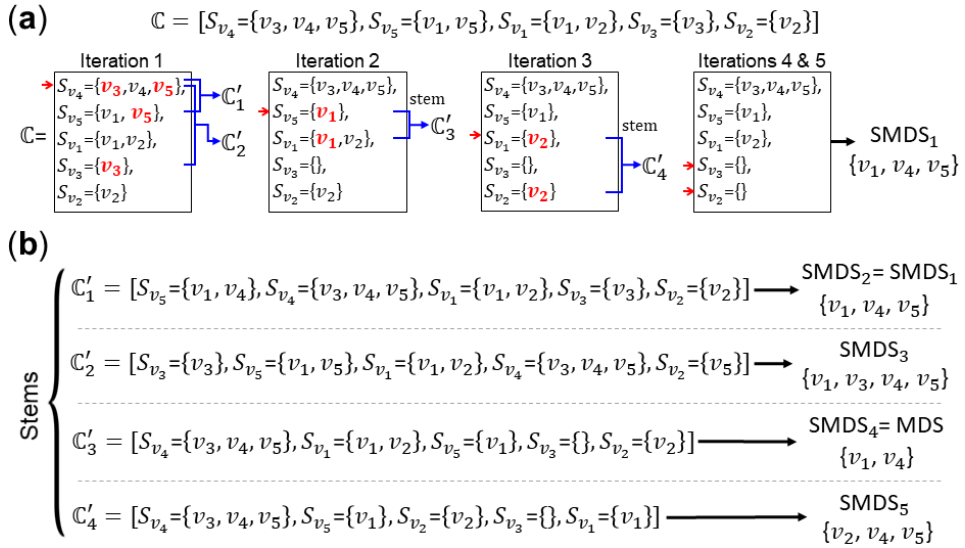

Fig. 4.3. Branching process for an MCSs-list.

The author illustrates an example of the branching process on the temporal network of Fig. 4.2 that resulted in creating four stems. In Fig. 4.3 (a) the red arrows mark an element of MCSs-list $\mathbb{C}$ that is being processed, and the blue arrows indicate the creation of a new stem. The procedure produced five SMDSs with various sizes and identified an MDS. Moreover, in Fig. 4.4 the author illustrates the complete execution of the heuristic approach for each stem.

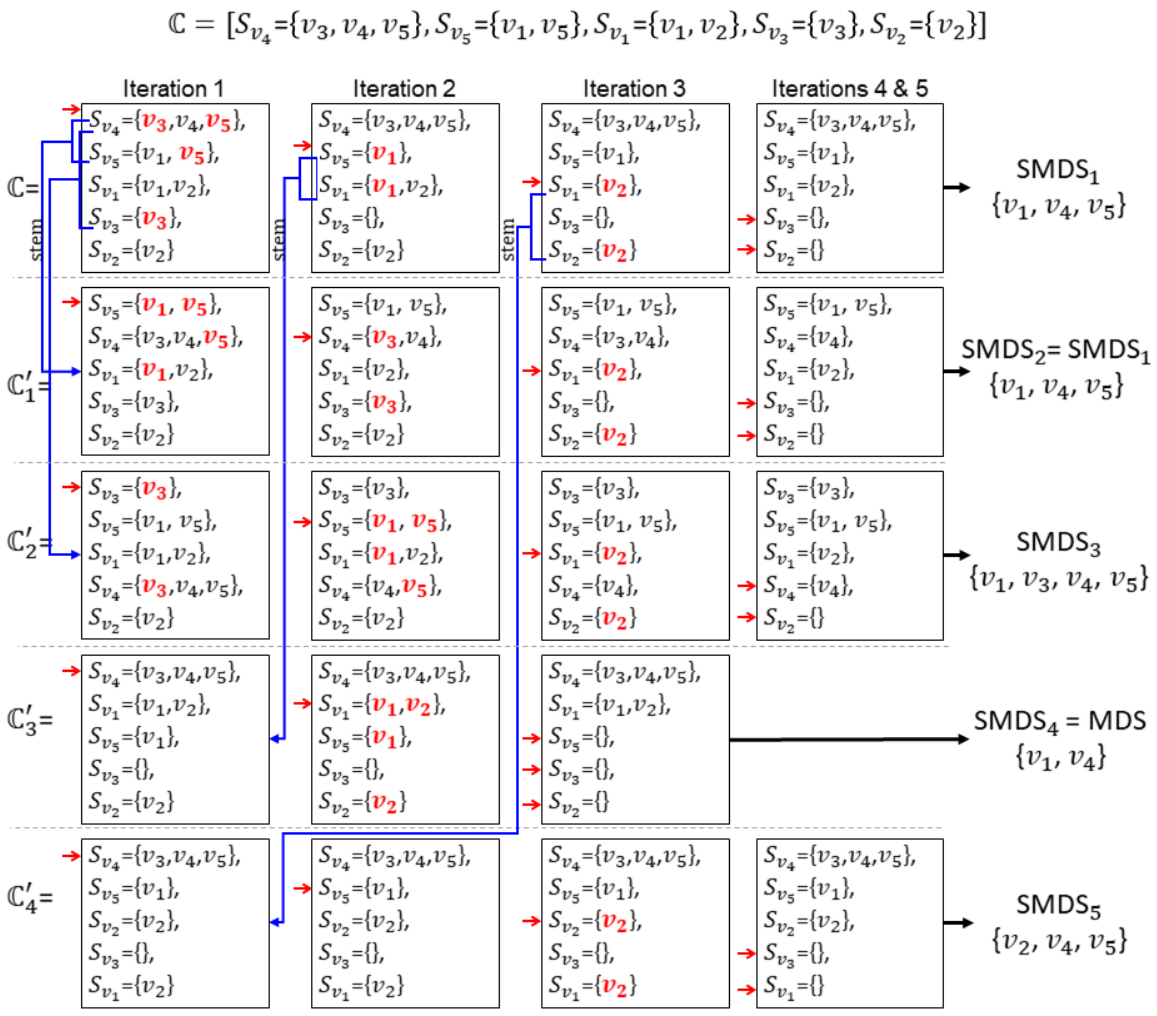

Fig. 4.4. Complete illustration of the branching procedure.

4.1.2 Heuristic Approach Algorithm

The author formally introduces the proposed heuristic approach in two algorithms. First, Algorithm 1 initializes the process to create an SMDS in three phases: 1) the *initialization* phase to declare the required variables, 2) the *creation* phase that employs Algorithm 2 to transform and MCSs-list to a potential SMDS, denoted by D', and create stems of $\mathbb{C}$, and 3) the *selection* phase that validates the potential SMDSs and return an SMDS with the smallest size. In practice the expected computational cost of the heuristic is $O(N^3 + N^2E + N\Delta t)$. The computational cost of each phase is as the following:

(1) *Initialization phase*: This phase (Algorithm 1 Lines 3-5) initializes the MCSs-list using the maximum flow approach on the time-layered network, which requires $O(N\Delta t + E)$ computation, where E is the number of network edges. The Ford-Fulkerson algorithm is used with computation cost of $O(Ef^*)$ where f^* is value of the maximum flow [48, 49]. In the time layered network, the flow capacity of all edges is equal to one, hence $f^* = N$ and finding an MCS is in the order of $O(NE)$. Therefore, to create an MCSs-list, N number of MCSs needs to be identified that results in the computation cost of $O(N^2)$ (Line 4 of Algorithm 1). Next, the heuristic approach sorts MCSs-list $\mathbb{C}$ in the descending order of its elements cardinality that requires $O(N \log N)$ computations. Therefore, the computational complexity of *Initialization* phase is in order of $O(N^2E + N\Delta t)$.

(2) *Creation phase*: This phase (Algorithm 1 Lines 5-11) utilizes Algorithm 2 for two purposes. First, transform an MCSs-list to a potential SMDS that cost $O(N^2)$ computation due to the nested loops in Lines 5 and 6 of Algorithm 2. Second, generate stems of $\mathbb{C}$ that is denoted by $\mathbb{C}_{stem}$ in Algorithm 1. At most a maximum of number of $\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} 1 = O(N^2)$ stems can be generated from $\mathbb{C}$, i.e., for each iteration of Algorithm 2 all $S^{j>i}$ must have a common node with S^i where $1 \leq i < j \leq N$. Therefore, the computational cost of *creation* phase is in order of $O(N^4)$ as Algorithm 2 needs to be executed for each stem.

(3) *Selection phase*: This phase (Algorithm 1 Lines 12-17) validates the created potential SMDSs in the ascending order of their cardinality (Line 12 of Algorithm 1) to return a smallest SMDS. The validation requires running a maximum flows algorithm to ensure a potential SMDS can yield flow N. Hence, at worse all identified potential SMDSs needs to be validated resulting in $O(N^2E)$ computations.

Algorithm 1 Build an SMDS

1: **Inputs:** (1) V: Nodes, (2) t_f: Deadline layer, (3) Δt: Expected steps.
2: **Output:** An SMDS denoted by D.
3: **procedure** FIND_SMDS($V, t_f, \Delta t$)
//Initialization phase
4: $\mathbb{C} \leftarrow [S_{v_k} = M_C(\{v_k\}, t_f, \Delta t) | v_k \in V]$
5: **Sort** $\mathbb{C}$ in a descending order of its elements cardinality
//Creation phase
6: $D', \mathbb{C}_{stem} \leftarrow$ Select_Drivers($\mathbb{C}$, True) // Calls Algorithm 2
7: $P \leftarrow D'$
8: **for each** $\mathbb{C}'$ **in** $\mathbb{C}_{stem}$ **do**
9: $D' \leftarrow$ Select_Drivers($\mathbb{C}'$, False)
10: $P \leftarrow P.append(D')$
11: **end for**
12: **Sort** P in an ascending order of its elements size
//Selection phase
13: **for each** D' **in** P **do**
14: **if** $M_C(D', t_f, \Delta t) == V$ **then**
15: **return** $D \leftarrow D'$ //An actual SMDS is found
16: **end if**
17: **end for**
18: **end procedure**

Algorithm 2 Selecting Driver Nodes Heuristic

Inputs: (1) $\mathbb{C}$: An MCSs-list,

(2) *stem*: Flag to enable the branching process to create stems.

Outputs: (1) D': a possible SMDS,

(2) $\mathbb{C}_{stem}$: stems created from $\mathbb{C}$ if parameter $stem ==$ True.

procedure SELECT_DRIVERS($\mathbb{C}, stem$)

    $\mathbb{C}_{stem} \leftarrow \{\}$

    for each S^i **in** $\mathbb{C}$ **;** $i = 1, 2 \ldots |\mathbb{C}|$ **do**

        for each S^j **in** $\mathbb{C}$ **;** $j = i+1, i+2 \ldots |\mathbb{C}|$ **do**

            if *stem* **is** True **and** $|S^i \cap S^j| > 0$ **then**

                $\mathbb{C}' \leftarrow \mathbb{C}$

                $\mathbb{C}' \leftarrow$ Swap the location of S^i and S^j in $\mathbb{C}'$

                $\mathbb{C}_{stem} \leftarrow (\mathbb{C}_{stem} \cup \mathbb{C}')$

            end if

            $S^j \leftarrow S^j \setminus S^i$

        end for

    end for

    $D' \leftarrow \{v_k | S_{v_k} \in \mathbb{C}, S_{v_k} \neq \{\emptyset\}\}$

    return $D', \mathbb{C}_{stem}$

end procedure

The theoretical computational cost of the heuristic approach is $O(N^4)$. However, in practice the computational cost is a factor of N lower. In Section 4.2.1 the empirical analysis showed that the number of stems is in order of $O(N)$, although the theoretical worst case is $O(N^2)$. Hence, based on the empirical analysis the computational cost of algorithms is closer to $O(N^3 + N^2E + N\Delta t)$.

4.2 Results

The author evaluated the proposed heuristic approach on empirical and synthetic temporal networks. The empirical networks are induced from e-mail communications in a manufacturing company and ants interactions in six colonies [17, 18, 50]. These datasets represent complex systems with different underlying control mechanisms. For example, often the organization of companies rely on a few key employees such as managers. On the other hand, many naturally evolved systems such as ant colonies are capable of self-organization without any leader. Hence, the author expects to observe these traits from the structural controllability perspective.

A strong state retention assumption is enforced to all evaluations, i.e., all nodes have self-loops within time intervals. This causes no loss of generality as the model can be simply modified to limit the state retention property. Also, a strong state retention configuration results in a highly dense time-layered network that expands the space of temporal pathways and the connectivity of nodes. For both ants and e-mails datasets, the temporal networks are created within the entire available periods of observations, i.e., t_f is set to the last recorded observation and t_0 to the first observation in the order of timestamps. For the e-mails dataset, the observations' timestamps were discretized with one-hour resolution, i.e., the time distance between each time layer is one hour. The six ant interactions datasets were not discretized hence the time distance between each layer varies depending on the observations' timestamps. For more information on the temporal characteristics of the datasets refer to Section 3.3.

Moreover, the author was able to identify all MDSs for the six ant colony networks using a brute force approach by utilizing Property 1. However, due to exponential combination of driver sets, the brute force approach was not able to identify any MDS for the e-mails network. For the ant colony networks, the author presented the minimum number of driver nodes in Fig. 4.5, denoted by N_c, and the smallest number of driver nodes identified by the heuristic approach is denoted by N_s. The

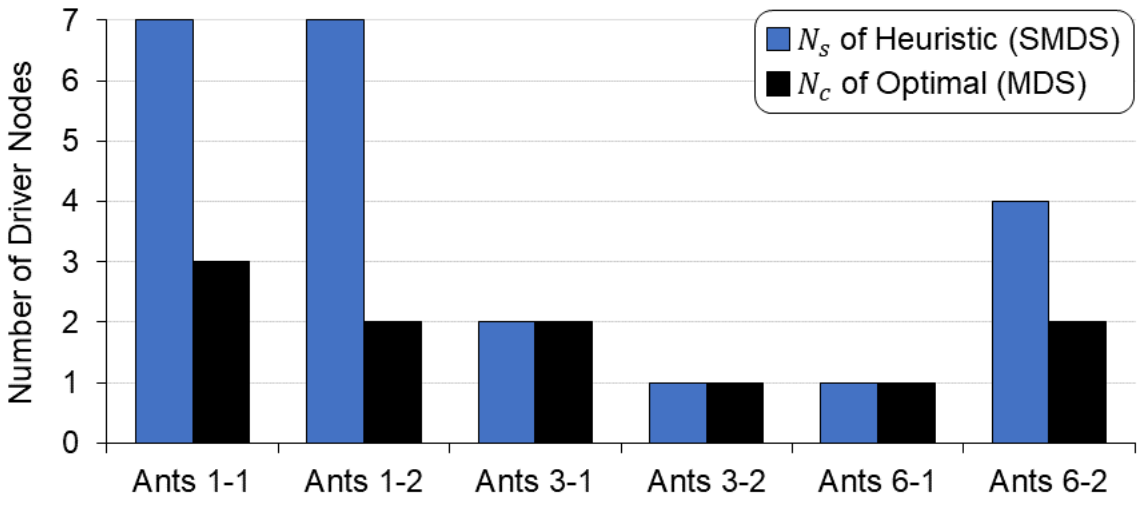

Fig. 4.5. Cardinality of MDSs and SMDSs for ant networks.

heuristic approach successfully identified SMDSs. Also, the heuristic found optimal solutions (MDSs) in the networks of colonies 3-1, 3-2, and 6-1.

4.2.1 Identifying Multiple Driver Node Sets

Many networks have more than a unique MDS with size N_c, which expands the space of nodes participating in MDSs. This resulted in categorizing network nodes into three groups namely *critical*, *intermittent*, and *redundant* [13,14,51]. The *critical* nodes belong to all MDSs, i.e., they are the intersection of all MDSs. In contrast, the *redundant* nodes do not appear in any MDS. A node is categorized as *intermittent* if it belongs to at least one MDS.

With a minor modification, the heuristic algorithm can produce multiple SMDSs of size $N_s \geq N_c$. Algorithm 1 can validate and return all created SMDSs at Step 15. In this dissertation, the author analyzes multiple SMDSs to further study the behavior of networks. However, this dissertation does not aim to establish definitions to characterize driver nodes for temporal networks.

The numerical results for both MDSs and SMDSs are presented in Table 4.1. For SMDSs the minimum size N_s and $N_s + 1$ are considered in all analysis. MDSs are identified by a brute force approach (column N_{MDSs}), but for the e-mail dataset no MDSs were identified as the combinatorial space is in order of $\binom{109}{40} = 109!/(40!69!)$ if $N_c \approx 40$. The column N_{SMDSs} shows the number of identified SMDSs with size

Table 4.1.
Numerical results for MDSs and SMDSs.

Network			MDSs (Optimal)					SMDSs (Suboptimal)						
Name	N	E	N_c	N_{MDSs}	f_i	f_r	f_c	N_{Stems}	N_s	N_{SMDSs}	$\lvert P\rvert$	f_i'	f_r'	f_c'
Ants 1-1	89	1911	3	153	0.59	0.41	0	231	7	2	2	0.02	0.91	0.07
									8	11	11	0.13	0.81	0.06
											Average:	0.07	0.86	0.06
Ants 1-2	72	1820	2	21	0.20	0.80	0	198	7	6	6	0.15	0.82	0.03
									8	19	19	0.26	0.72	0.02
											Average:	0.2	0.77	0.02
Ants 3-1	11	78	**2**	7	0.54	0.46	0	17	**2***	1	1	0	0.82	0.18
									3	6	6	0.64	0.27	0.09
											Average:	0.32	0.54	0.13
Ants 3-2	6	104	**1**	5	0.83	0.17	0	11	**1***	5	5	0	0.83	0.17
									2	5	5	1.0	0	0
											Average:	0.5	0.41	0.08
Ants 6-1	33	652	**1**	3	0.09	0.91	0	50	**1***	3	3	0.09	0.91	0
									2	19	19	0.54	0.46	0
											Average:	0.31	0.68	0
Ants 6-2	32	367	2	10	0.28	0.72	0	44	4	5	5	0.22	0.78	0
									5	14	14	0.53	0.47	0
											Average:	0.37	0.62	0
E-mail	109	250	Too big to compute					56	51	6	6	0.08	0.50	0.42
									52	17	17	0.25	0.39	0.36
											Average:	0.16	0.44	0.39

* At least one MDS is identified by the heuristic approach.

N_s and $N_s + 1$. Also, to present the efficiency of the proposed algorithm, column $|P|$ reports the total number of potential SMDSs, i.e., the number of stems of $\mathbb{C}$ plus one. This shows all potential SMDSs did pass the validation test and were actual SMDSs. Finally, for the heuristic approach the columns f_i', f_r', and f_c' present the fraction of *intermittent*, *redundant*, and *critical* nodes. Similarly, for MDSs the columns f_i, f_r, and f_c present the fractions of nodes.

In Section 4.1.2, the author mentioned that the number of stems is expected to be in the order of $O(N)$ based on the empirical results. This is presented by the column N_{stems} of Table 4.1 signaling that the number of stems linearly grows in relation with N. Also, for the ant networks 3-1, 3-2, and 6-1 the heuristic approach identified MDSs (i.e., where $N_s = N_c$). The possibility of identifying MDSs by the heuristic may be explained by the degree of uniqueness between MCSs of individual nodes in a network. For instance, the frequency of a node appearing in MCSs of other nodes. Such analysis requires an efficient approach to identify all MCSs of a node, which is the focus of Chapters 5 and 6.

4.2.2 Queen Ants

The analysis of structural controllability showed queen ants tend to avoid becoming a driver node in both MDSs and SMDSs. In nature, the main responsibility of queen ants is reproduction and she does not contribute in the organization of the colony (e.g., temperature management and decision to forage for food). Table 4.2 presents the fraction of edges connected to a queen ant and her participation in the driver node sets. The queen ants have a high in-degree and out-degree in the time-aggregated networks. This indicate that although many interactions with the queen ants were recorded, a queen rarely was selected as a driver node.

Table 4.2.

Queen ants' interactions.

Network	N	Queen	in-degree	out-degree	Interactions	Fraction in MDSs	Fraction in SMDSs
Ants1-1	89	Q1	42	30	23	4%	0%
Ants1-2	72	Q1	17	23	21	0%	30%
Ants3-1	11	Q1	3	4	3	0%	0%
Ants3-2	6	Q1	10	7	4	0%	10%
Ants6-1	33	Q1	23	0	11	0%	0%
		Q2	14	26	15	0%	0%
Ants6-2	32	Q1	10	6	9	0%	0%

4.2.3 Degree Distribution of MDSs and SMDSs

For the time-aggregated network of ant colonies, the author presented the average in/out/total degree distributions of driver nodes in Fig. 4.6. This analysis tests whether driver nodes tend to have a large or a small degree in temporal networks. The results showed driver nodes tend to have a large degree in both MDSs and SMDs. Also, the out-degree of driver nodes is slightly higher than their in-degree. Similarly, Hu et al. [52] found that messenger nodes tend to have a large degree in the structural observability problem (a dual problem of structural controllability). These results are in contrast with static networks where it is shown that driver nodes tend to have a small degree and avoid hubs [2].

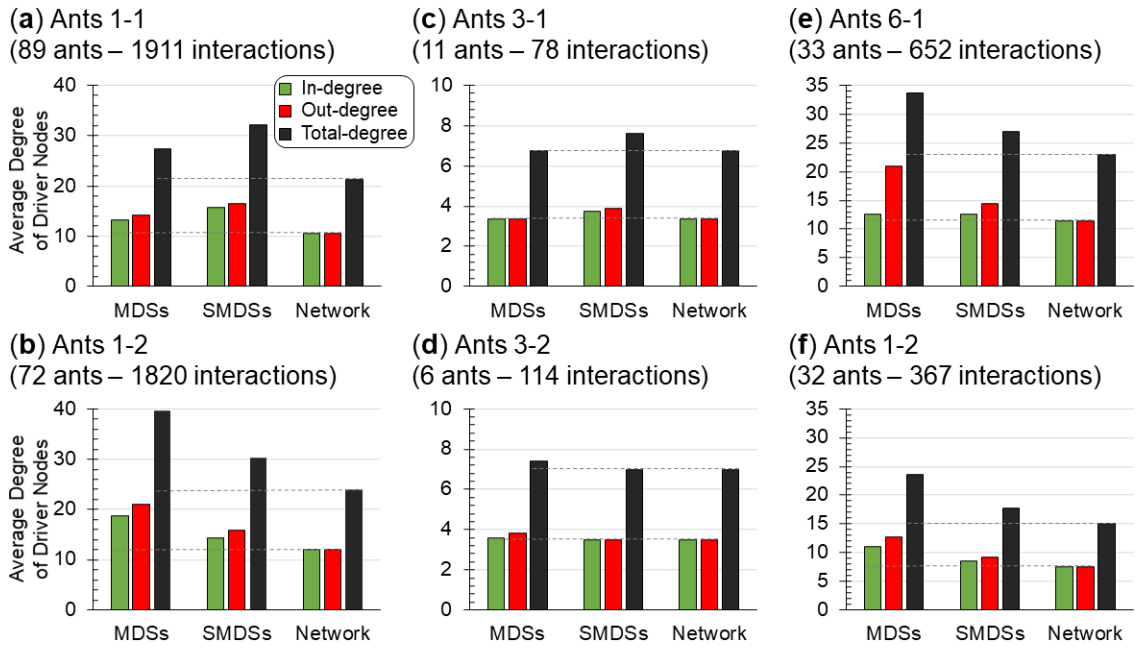

Fig. 4.6. Driver nodes degree distributions.

4.2.4 Evaluations Using Network Randomizations

The author utilizes randomization techniques to identify the most influential characteristics of temporal networks on their structural controllability. Five randomization techniques are employed and they are formally introduced in the next section. The author illustrates the effect of each randomization technique on N_s and N_c in Fig. 4.7 and all results were averaged over 10 realizations.

The main observation is that degree distribution is the most influential factor on the structural controllability of temporal networks. In Fig. 4.7 (d), DPN (Degree Preserved Network) has the least difference on N_c of the randomized and original networks. As discussed in the next section, DPN preserves both the degree distributions between time-layers and the time-aggregated network, but it eliminates the network structure. Also, DPN similarly behaves on the heuristic approach (i.e., N_s) except for colony 1-1.

Focusing on N_s, in Fig. 4.7 (b-c), RT (Random Time) and RP (Randomly Permuted) has the smallest delta on N_s that signals the influence of degree distribution and the structure of networks. In Fig. 4.7 (a), the fluctuations of N_c and N_s present the effect of network topology assessed by the RE (Randomized Edges) technique. Furthermore, using RN (Random Network) in Fig. 4.7 (e), the largest delta between N_s and N_c is observed since RN eliminates the degree distributions between each time-layer and the time-aggregated network. Lastly, the e-mails network behaves similarly against all randomization techniques.

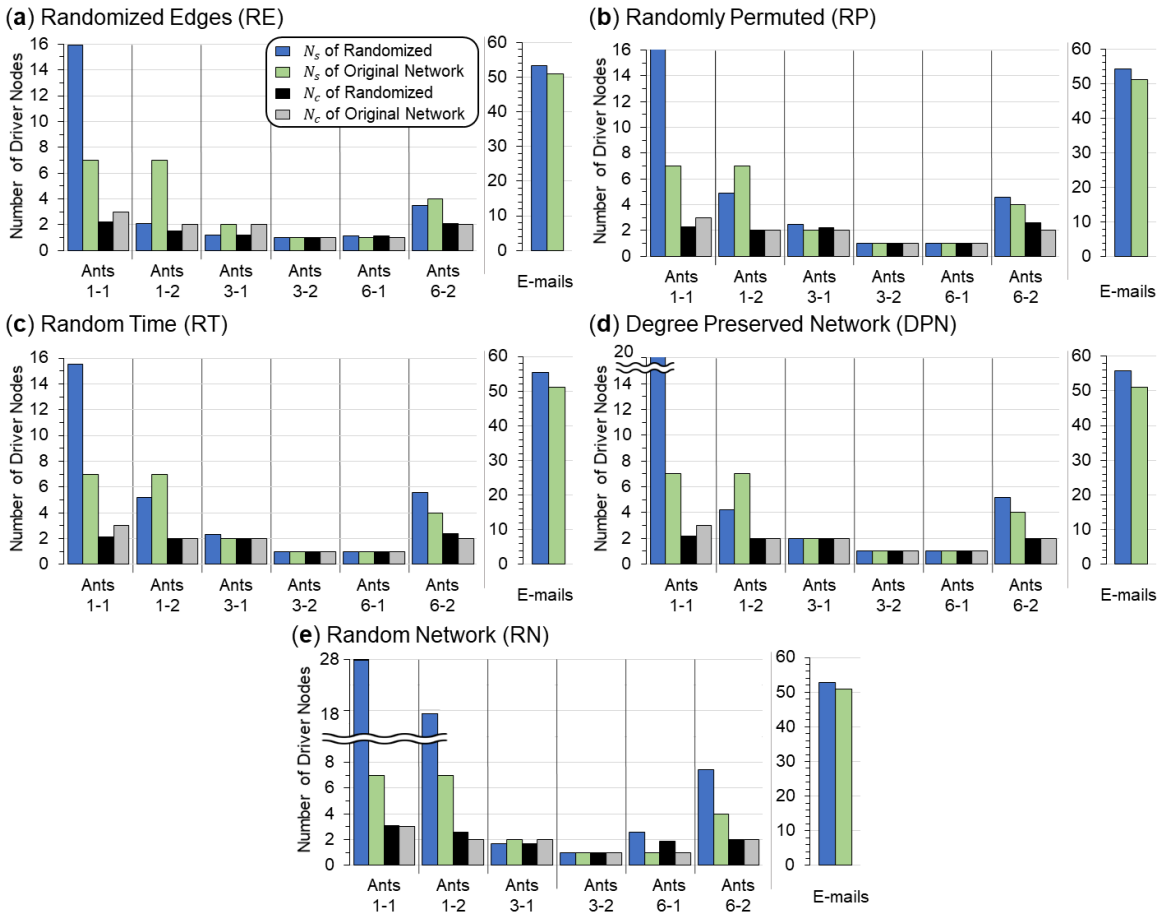

Fig. 4.7. Randomization results.

Randomization Procedures

The author utilized the randomization techniques in Table 4.3 where the green and red cells indicate preservation and elimination of a network's characteristic respectively. Holme and Saramäki [3] provided a comprehensive overview of methods for randomizing temporal networks. The following provides a summary of the utilized randomization techniques:

Randomized Edges (RE): The effect of network topology can be studied using this method. RE assumes that edges govern the temporal relations (i.e., time of contacts) instead of the vertices, and it changes who contacts whom. The sequence of contacts is preserved since the timestamps of edges will not be changed. Also, the temporal correlations associated with edges are preserved (e.g., average degree fluctuations). An algorithm to implement RE is introduced in [3].

Randomly Permuted (RP): RP shuffles the timestamps of edges and it retains the structure of time-aggregated network and the number of contacts associated with each edge. However, RP eliminates all temporal correlations (e.g., causal events) and it changes the degree distribution of each time layer.

Random Time (RT): RT assigns timestamps to temporal links. Hence, RT eliminates both local (e.g., simultaneous events) and global (e.g., overall fluctuations in the degree distributions) correlations. But, RT retains the structure of time-aggregated network.

Degree Preserved Network (DPN): DPN randomly rewires the networks between each time-layer given their corresponding original degree sequences. Hence, the timestamps of edges will not change. DPN only preserves the degree distribution between time-layers. However, DPN eliminates all other structural and temporal correlations.

Random Network (RN): RN replaces the networks between time-layers with Erdős-Reńyi networks that have the exact same number of links as the original networks. Hence, RN eliminates the structure of the time-aggregated network and all temporal correlations. Similar to DPN, RN does not modify the timestamps of interactions.

Table 4.3.

Randomization techniques.

Method	Aggregated network structure[a]	Aggregated network degree distribution	Local degree distributions[b]	Overall rate of events[c]	Average degree fluctuations	Causal chain of events
RE						
RP						
RT						
DPN						
RN						

[a] Retains who contacts whom.

[b] Local refers to the networks between time-layers and their degree distributions.

[c] The level of simultaneous interactions within time-layers. Such as the patterns in amount of daily, weekly, or monthly messages in the e-mail networks.

4.2.5 Synthetic Temporal Networks

The author evaluated the behavior of proposed heuristic approach on synthetic temporal networks generated using Erdős-Reńyi and Barabási–Albert models with various sizes (average degree) [53]. The generation process of synthetic networks was based on generating an Erdős-Reńyi or Barabási–Albert network for each time-layer. In Fig. 4.8, the author illustrates N_c and N_s versus average degree $\langle k \rangle$ of the time-aggregated networks. All temporal networks have 50 time-layers and 15 nodes, and the results were averaged over 10 realizations. As illustrated in Fig. 4.8, the heuristic approach and optimal solutions converged as the size of networks was increased. Also, Barabási-Albert model required more driver nodes compared to Erdős-Reńyi. Moreover, N_s and N_c converged on a slightly smaller $\langle k \rangle$.

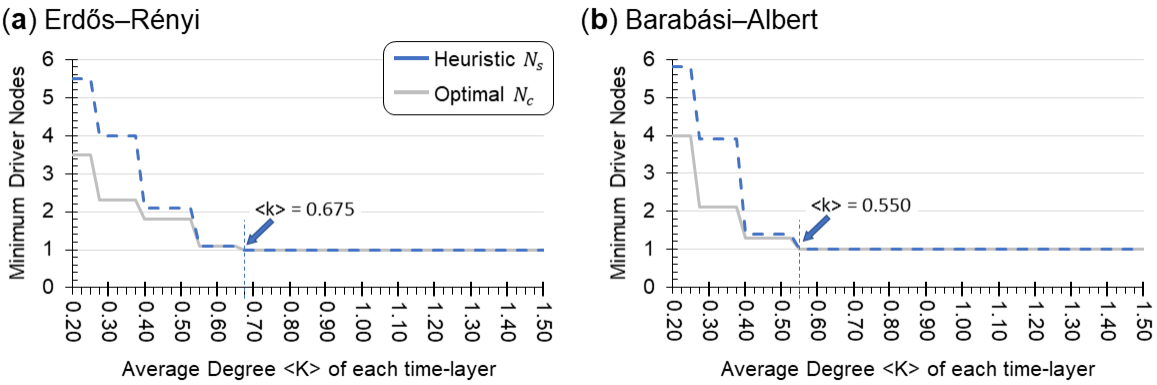

Fig. 4.8. Synthetic temporal networks.

4.3 Conclusion

In this chapter, the author proposed algorithms to identify and use driver node sets that can make a temporal network structurally controllable. Also, a comprehensive analysis of controllability was presented on empirical and synthetic temporal networks. The results indicated that driver nodes tend to have a large degree in temporal networks.

5. COMPLETE CONTROLLABLE DOMAIN

The predominant research on analyzing the controllable space of a driver node focuses on the maximum cardinality of controllable nodes. In this chapter, the author focuses on the multiple controllable subspaces of a driver node with the maximum controllable cardinality. The author introduces the Complete Controllable Domain (CCD) as the union of all MCSs of a driver node and investigates the characteristics of CCD.

5.1 Exponential Number of MCSs

A node, namely $v \in V$, can have an exponential number of MCSs depending on the network size (number of nodes) and the maximum number of controllable nodes by v is denoted by n_d. The author illustrates an example in Figure 5.1, where node v_1 can originate two independent time respecting paths at $t = 1$ and $t = 2$ that can end on any node at $t_f = 2$ (except v_1). Hence, any combinations of size 2 from $\{v_2, v_3, \ldots, v_n\}$ is an MCS of v_1.

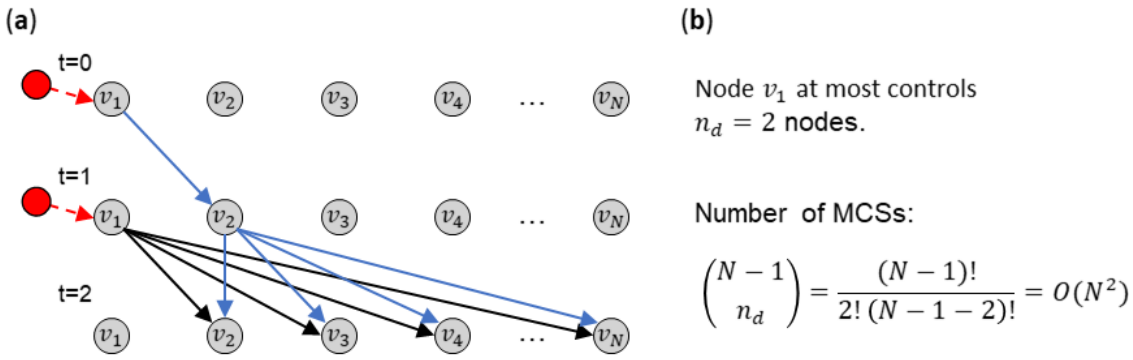

Fig. 5.1. Driver nodes can have an exponential number of MCSs.

Accordingly, Equation 5.1 provides the worst case upper-bound on the number of MCSs for a single node.

$$\binom{N-1}{n_d} = \frac{(N-1)!}{n_d!(N-n_d-1)!} \tag{5.1}$$

Equation 5.1 has the highest possible value if $n_d \approx N/2$. Hence, by using Stirling's approximation for $n!$ the asymptotic value of Equation 5.1 is:

$$\lim_{N\to\infty}\binom{N}{\frac{N}{2}} = \lim_{N\to\infty}\frac{N!}{(\frac{N}{2})!(\frac{N}{2})!} \approx \lim_{N\to\infty}\frac{\sqrt{2\pi N}(\frac{N}{e})^N}{[\sqrt{\frac{2\pi N}{2}}(\frac{\frac{N}{2}}{e})^{\frac{N}{2}}]^2} = \sqrt{\frac{2}{\pi N}}2^N \tag{5.2}$$

5.2 Algorithm to Identify k MCSs

A driver node can have an exponential number of MCSs hence the cardinality of CCD of a driver node can exponentially grow. Therefore, the researcher proposes a branch and bound algorithm to approximate the CCD of a driver nodes (see Algorithm 3). The proposed algorithm obtains k MCSs for a single driver node with computational cost of $O(kn_d^3NE^3)$ (derived in Section 5.3).

The main idea of the algorithm is checking whether a node that belongs to an MCS of a driver can be replaced by another node. Hence, for a driver node $d \in V$, the algorithm blocks the independent time-respecting paths (i.e., flows in the time-layer network) by strategically removing connections to the sink $u \to t$ from time-layered network, denoted by G, and assess if d still has a maximum flow in G. The strategy is to create an n_d-ary tree, namely T, for a driver node d where each node in T has at most n_d children [49]. Each edge in T represents the removal of an edge $u \to t$ in G. That is, the depth of a node in T indicates the number of removed connections to the sink node. In other words, each edge in T divides the possible spectrum of MCSs.

For example, consider the simple temporal network in Fig. 5.2, denoted by T that has 6 MCSs of size $n_d = 2$. The author illustrates an example of execution of Algorithm 3 in Fig. 5.3. For illustration, the black boxes show the possible space of MCSs at each node that can be randomly selected using the maximum flow algorithm.

The highlighted edges and nodes in Fig. 5.3 present a complete n_d-ary tree for the temporal network in Fig. 5.2. On the other hand, the dashed edges present the space that can result in other n_d-ary trees. Moreover, in T a same MCS can be returned by the max flow algorithm for two nodes. An example of this case is marked in Fig. 5.3 by the blue star. However, the children of such nodes represent different spectrum (branches) of MCSs as the combination of edges from the root indicates the removal of a unique set of nodes from the time-layered network.

The bound condition of the proposed branch and bound algorithm is based on the prevention of processing redundant sub-trees of T. A sub-tree, namely T', is considered redundant if there exists another node in T that represents the same combination of edge removals as the root of T'. To prevent creating redundant sub-trees, the algorithm keeps the record of edge removal sets using set B^e defined in Line 10 of Algorithm 3. Figure 5.3 presents the two redundant edge removals (sub-trees) by the dashed ellipses.

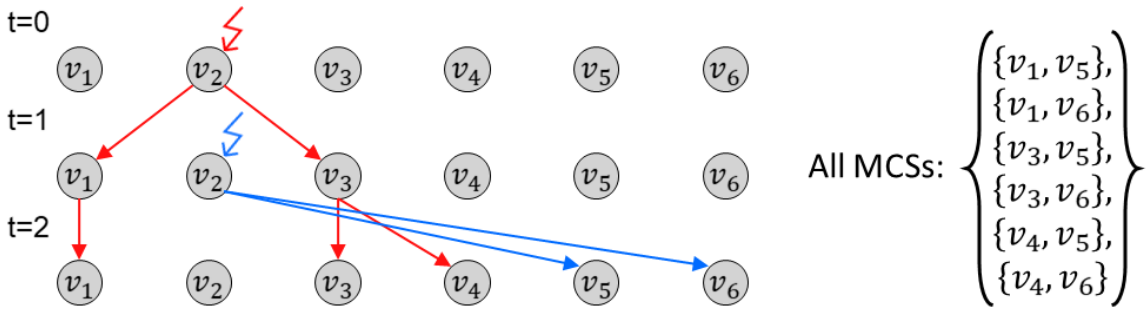

Fig. 5.2. A simple temporal network with 6 MCSs.

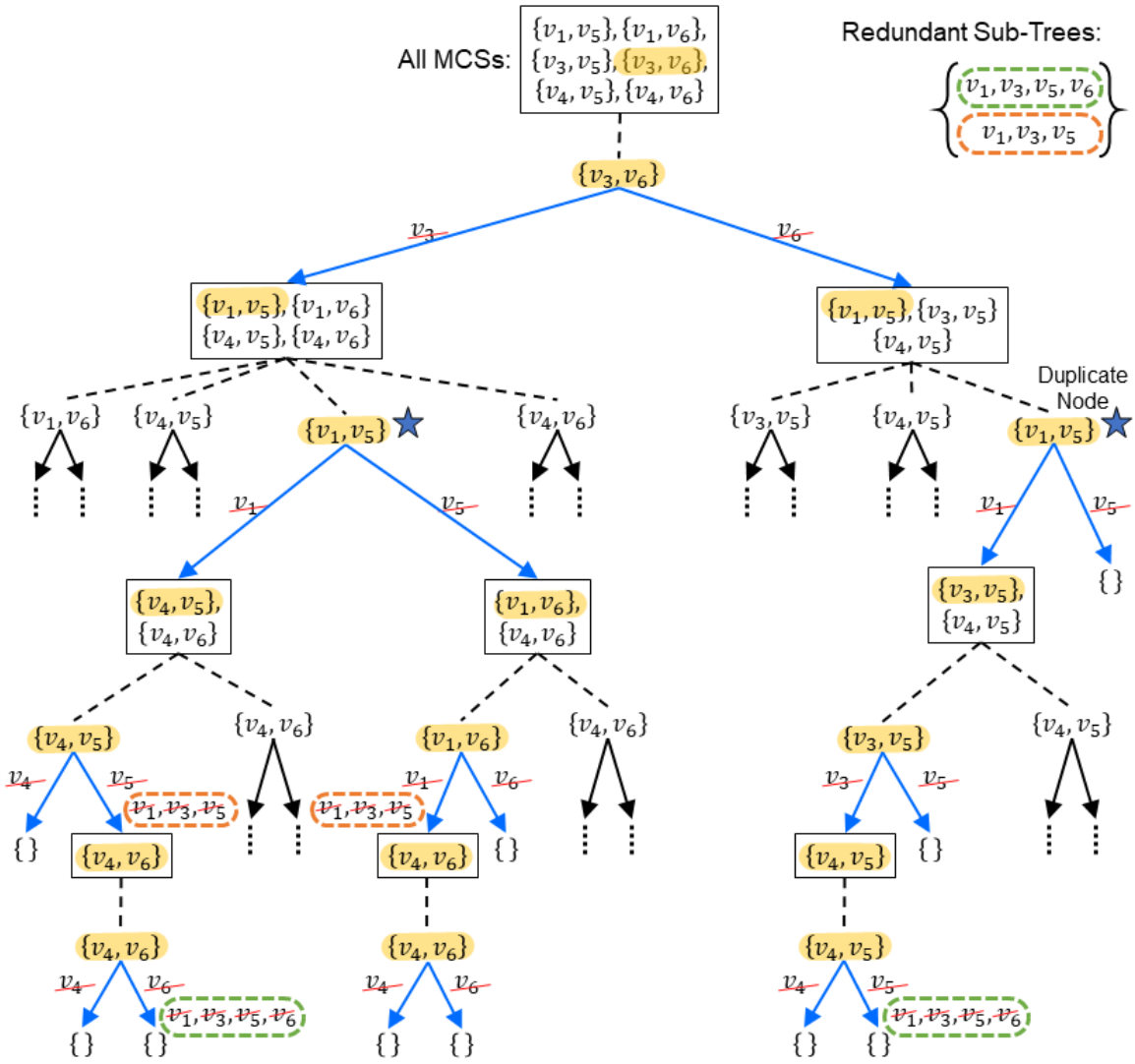

Fig. 5.3. Execution of the proposed algorithm on a temporal network.

Algorithm 3 Obtain k MCSs of a single driver node

Inputs: (1) G: time-layered network, (2) d: a driver node,
(3) k: find at most k MCSs of d, (4) t: the sink node of G

Output: $MCSs$: a set that contains k MCSs of d.

procedure FIND_MCSs(G, d, k, t)
  $M \leftarrow M_C(G, d)$ // M_C finds an MCS of d using the max flow approach
  $n_d \leftarrow |M|$
  $MCSs \leftarrow \{M\}$
  $B \leftarrow [[r]|r \in M, r \neq d]$ // Branches of tree
  $B^e \leftarrow \{\}$ // Evaluated branches
  while true **do**
    if $|MCSs| == k$ **then return** $MCSs$
    $B_l \leftarrow B.popFromHead()$ // BFS order
    if $B_l \in B^e$ **then continue**
    Remove all edges $\{u \rightarrow t|u \in B_l\}$ from G
    $M \leftarrow M_C(G, d)$
    if $n_d == |M|$ **then**
      $B' \leftarrow [B_l + [r]|r \in M, r \neq d]$ // Create new branches
      $B \leftarrow B.append(B')$
      $B^e \leftarrow B^e.append(B')$
      $MCSs \leftarrow MCSs.append(M)$
    end if
    Add all edges $\{u \rightarrow t|u \in B_l\}$ to G
  end while
  return $MCSs$
end procedure

5.3 Computational Complexity

The proposed algorithm uses the Edmonds-Karp algorithm to find maximum flows, which requires $O(NE^2)$ computations. Also, to find k number of MCSs, the algorithm needs to build an n_d-ary tree, namely T, with at least k nodes. Hence the height of T is at least:

$$h = \lfloor \log_{n_d} ((n_d - 1)k) \rfloor \tag{5.3}$$

Therefore, at least $\Omega(n_d^h) = \Omega(n_d k)$ computations are required if all nodes in T are a unique MCS. However, since the tree can have duplicated nodes (as illustrated in Fig. 5.3) its height may grow higher than the minimum. In practice, the researcher observed that building T with height $h \approx 3$ is sufficient to find k MCSs (the empirical evidence is presented in Section 5.5). Hence, in practice the computational cost of this step is $O(kn_d^3)$. Therefore, the total computation cost of the algorithm is $O(kn_d^3 NE^2)$.

5.4 Algorithm Optimization

Algorithm 3 creates an n_d-ary tree in the Breadth-First Search (BFS) order by removing the branches from the head of branch-queue in Line 9 of the algorithm. In contrast, the algorithm can build an n_d-ary tree in the Depth-First Search (DFS) order by removing from the tail of branch-queue. The researcher chose to create the tree in BFS order for two reasons. First, by using DFS order, an n_d-ary tree grows vertically that limits the finding of MCSs to a few branches. In other words, the edge removals combinations will have a large intersection. Second, since a node can have an exponential number of MCSs, the height of an n_d-ary tree is un-bounded. Hence, the proposed computational cost analysis in Section 5.3 is valid if the tree grows horizontally.

5.5 Evaluation

In Section 5.3 the researcher proposed that based on the empirical results it is sufficient to build n_d-ary trees with height ≈ 3 for identifying k number of MCSs. In Fig. 5.4, the author presented the aforementioned empirical observation between the average MCS size of nodes (i.e., n_d) and the average height of n_d-ary trees. The empirical analysis showed on average the height of n_d-ary trees are orders of magnitude smaller than n_d. This is aligned with the proposed worst-case computational complexity of Algorithm 3.

Moreover, the proposed branch and bound algorithm is evaluated on empirical datasets to investigate the number of bounded sub-trees and duplicate tree nodes. In

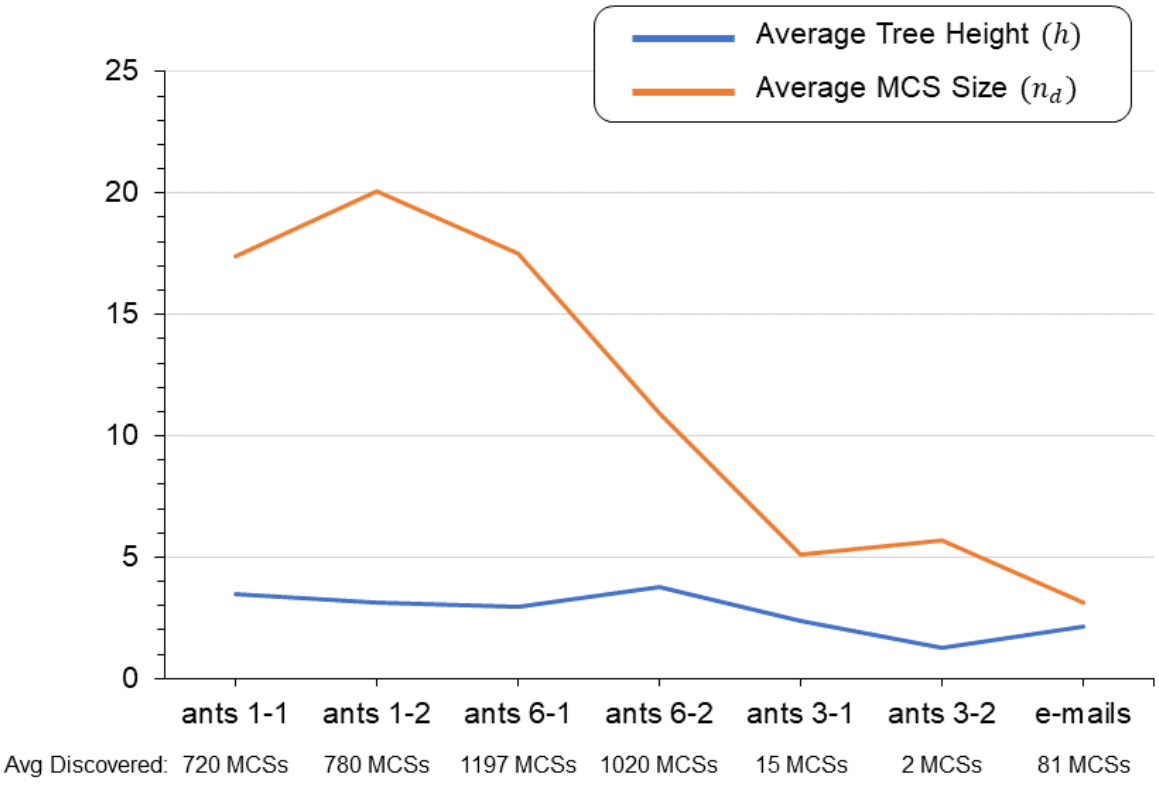

Fig. 5.4. Average height of n_d-ary trees in real-world temporal networks.

Fig. 5.5, the author illustrates the performance of Algorithm 3 (with un-bounded k) on the real-world temporal networks. Overall, the average number of bounded sub-trees (marked with green lines) is close to size of created n_d-ary trees (i.e., average number of tree nodes), which presented the effectives of proposed algorithm.

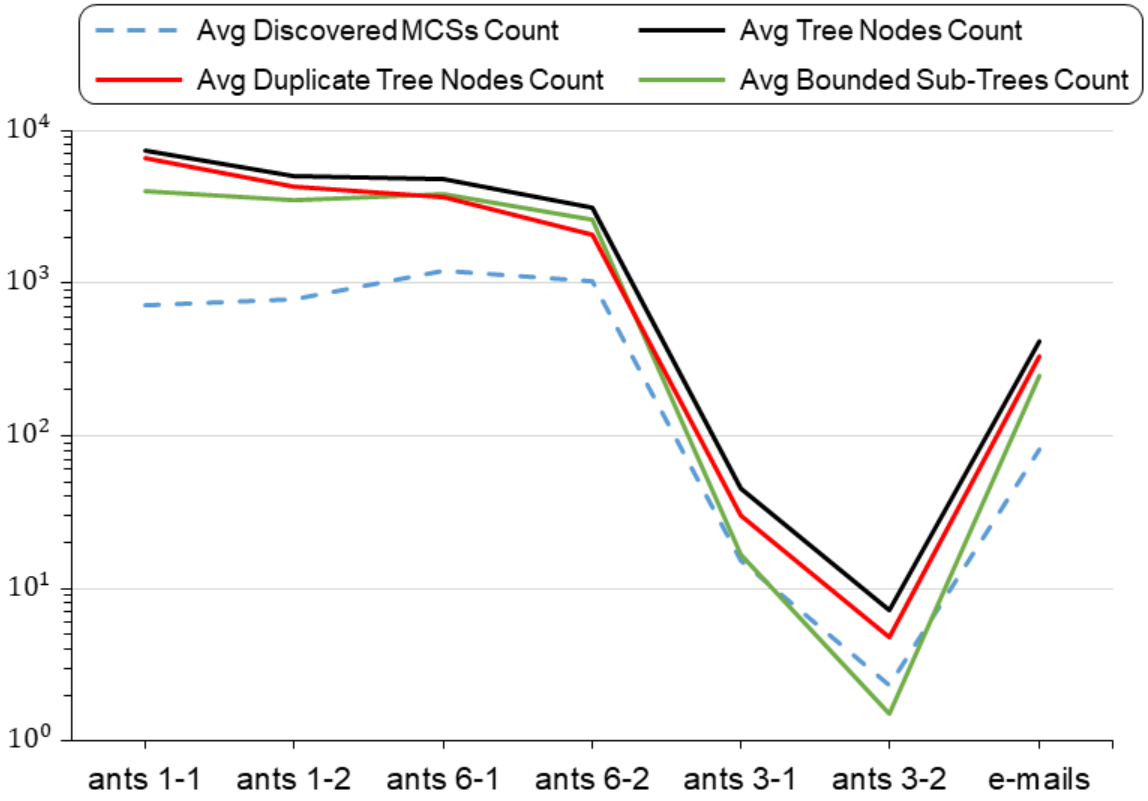

Fig. 5.5. Performance of the branch and bound algorithm.

5.6 Conclusion

In this chapter, the author showed that a node can have an exponential number of MCSs. Hence, a branch and bound algorithm to identify k number of MCSs was introduced and its performance was analyzed on a variety of real-world temporal networks.

6. ANALYSIS OF COMPLETE CONTROLLABLE DOMAIN

The purpose of this chapter is to investigate the characteristics of CCD of individual driver nodes approximated by Algorithm 3. The researcher begins by analyzing the intersection between MCSs of nodes for the real-world temporal networks introduced in Section 3.3. For node $v \in V$, let I_v^k denote the intersection of maximum k MCSs of v. In other words, I_v^k represents the set of nodes that driver node v always controls. Given that a driver node at most can control the full network, and to exclude the driver itself from its MCSs-intersection, we have $0 \leq |I_v^k| \leq N - 1$. Moreover, $\mathcal{I}^k$ denotes the average size of MCSs-intersection for a temporal network as defined in Equation 6.1 where $N = |V|$.

$$\mathcal{I}^k = \frac{1}{N} \sum_{v \in V} |I_v^k| \tag{6.1}$$

Similarly, the frequency of a node, namely v, in MCSs-intersection of all other nodes is denoted by f_v^k as defined by Equation 6.2. In other words, f_v^k is the number of driver nodes that always control v. Since a node can be in MCSs-intersection of at most $N - 1$ driver nodes then $0 \leq f_v^k \leq N - 1$.

$$f_v^k = \sum_{j \in V, j \neq v} \begin{cases} 1, & \text{if } v \in I_j^k; \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

The average frequency of nodes in MCSs-intersection of other nodes for a network is denoted by $\mathcal{E}^k$ as defined in Equation 6.3.

$$\mathcal{E}^k = \frac{1}{N} \sum_{v \in V}^{|} f_v^k \tag{6.3}$$

6.1 Convergence of MCSs-Intersection on k

The researcher evaluated the values of $\mathcal{I}^k$ and $\mathcal{E}^k$ for various k as illustrated in Fig. 6.1. The results were averaged over 5 realizations for each k. At the first glance, $\mathcal{I}^k$ (marked by red solid lines) converges to a fixed value for $k \approx 25$ in the evaluated temporal networks. Hence, this result indicated that it is possible to approximate the CCD of a driver node by having a limited number of its MCSs.

Interestingly, the researcher observed that for all networks $\mathcal{I}^k$ and $\mathcal{E}^k$ converge. This convergence indicates that on average the number of nodes that are always controlled by a driver node is equal to the number of driver nodes that always control

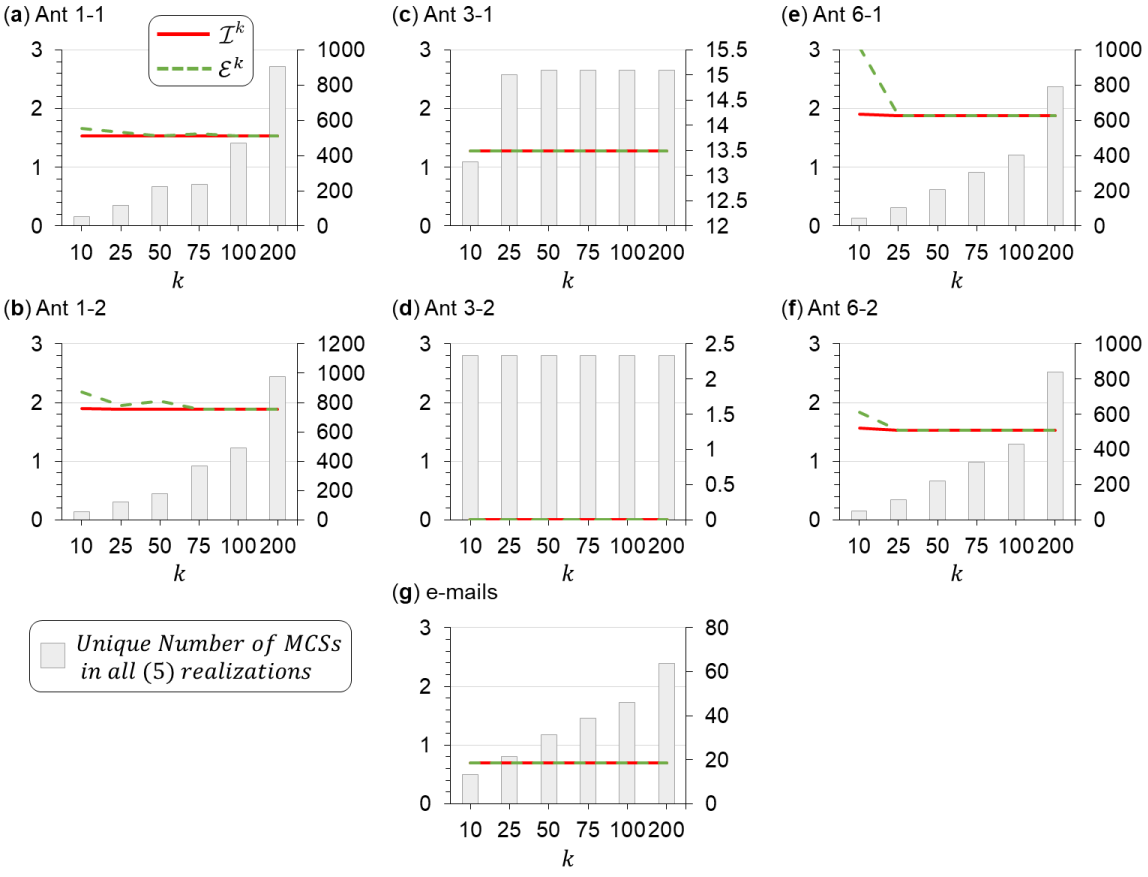

Fig. 6.1. Convergence of MCSs-intersections on small k

a node. Further investigation to explain the reason behind this convergence is needed, however, it is out of the scope of this dissertation. Moreover, the author presented the number of discovered MCSs over 5 realizations of Algorithm 3 using the bar charts in Fig. 6.1. Overall, in the large and medium size ant colonies (i.e., 1-1, 1-2, 6-1, and 6-2) on average over 800 MCSs were discovered indicating that most nodes (ants) had a large CCD and had the potential to influence a large fraction of nodes. However, in the e-mails network the small number of unique MCSs (around 60) indicated that many nodes did not have any chance to control a large fraction of the networks.

6.2 MCSs and Driver Nodes Sets

In Chapter 4, the author proposed a heuristic approach to identify multiple SMDSs for temporal networks. Also, the optimal driver sets (MDSs) were identified for the ant networks. Drawing on the MDSs and SMDSs concepts, the author aims to investigate the relationship between MCSs and the driver node sets. In pursuit of this aim, Equations 6.4 and 6.5 quantify the tendency of MCSs-intersection of node to participate in the driver node sets.

$$\mathcal{D}_M^k = \frac{1}{N} \sum_{v \in N} \sum_{M \in MDSs} \begin{cases} 1, & \text{if } I_v^k \cap M \neq \{\emptyset\}; \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

$$\mathcal{D}_S^k = \frac{1}{N} \sum_{v \in V} \sum_{S \in SMDSs} \begin{cases} 1, & \text{if } I_v^k \cap S \neq \{\emptyset\}; \\ 0, & \text{otherwise} \end{cases} \tag{6.5}$$

Moreover, the author quantifies the frequency of nodes appearing in driver sets based on Equations 6.6 and 6.7.

$$Q_M = \frac{1}{N} \sum_{v \in V} \sum_{M \in MDSs} \begin{cases} 1, & \text{if } v \in M; \\ 0, & \text{otherwise} \end{cases} \tag{6.6}$$

$$Q_S = \frac{1}{N} \sum_{v \in V} \sum_{S \in SMDSs} \begin{cases} 1, & \text{if } v \in S; \\ 0, & \text{otherwise} \end{cases} \tag{6.7}$$

The author illustrates the aforementioned measurements with respect to the driver node sets (MDSs and SMDSs) in Fig. 6.2. Algorithm 3 is executed with $k = 200$ and all results were averaged over 5 realizations. Two main observations can be drawn based on the results depicted in Fig. 6.2. First, in the case of ant networks the proposed heuristic approach (Chapter 4) and optimal solutions behave similarly. For example, as illustrated in Fig. 6.2 if in a colony $D_M^{k=200} > Q_M$ then the corresponding analysis for SMDSs also showed $D_S^{k=200} > Q_S$. Second, overall in ant colonies the MCSs-intersection of nodes more appeared in the driver node sets compared to the level of nodes that appeared in the driver sets. However, in the e-mails networks the MCSs-intersection of nodes tend to less participate in the driver node sets. This behavioral difference between the e-mails and ant networks could be due to the fundamental underlying control mechanisms that these systems operate under. Ant societies self-organize themselves, however, the companies rely on managers to be organized. Chapter 7 continues this discussion by analyzing the control regimes of complex networks.

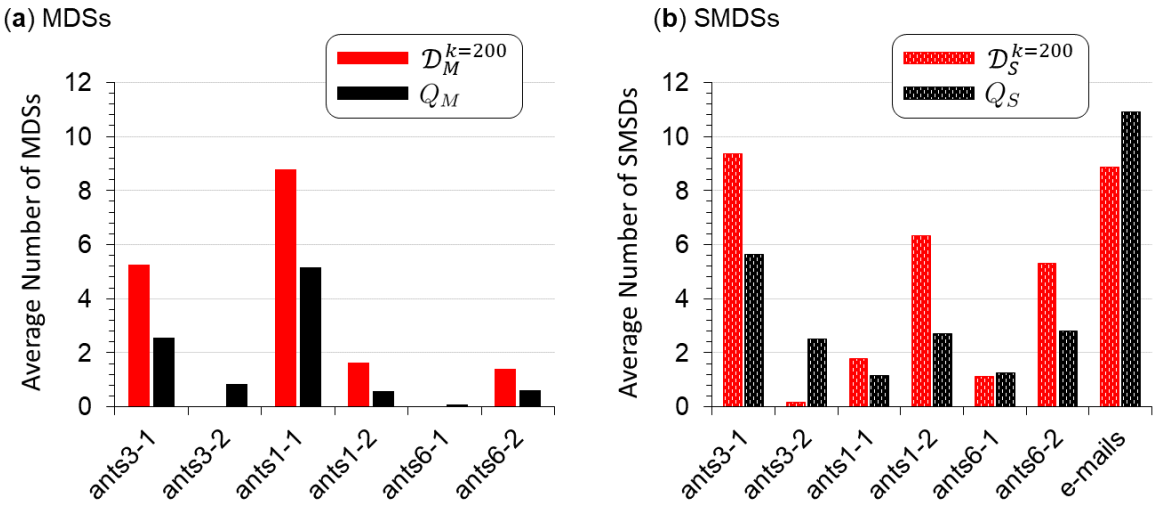

Fig. 6.2. Participation in driver node sets

6.3 Conclusion

This chapter analyzed the behavior of ant colony and e-mail communication networks by investigating multiple MCSs of individual driver nodes and approximating the CCD of driver nodes. In other words, this chapter provided a showcase of employing the proposed algorithm in Chapter 5 that identifies k MCSs of a network. For instance, the empirical results showed the cardinality of MCSs-intersection of driver nodes converged on $k \approx 25$.

7. DISCUSSIONS AND FINDINGS

This dissertation provided a set of algorithms and tools to understand and analyze the controllability of temporal networks. The author enforced a strong state retention assumption to all evaluation by allowing all nodes to have self-loops. This assumption enables nodes to retain their previous state in the absence of interactions between nodes in consequent periods of time. In other words, this assumption lifts the common modeling limitation that the flow of information and influence depends on having a continues chain of observations. Moreover, employing self-loops in the structure of networks is aligned with the behaviors of many physical systems [45]. For instance, in e-mails communications, the lack of connections during weekends will not stop the diffusion of information from the e-mail received before the weekends.

Furthermore, the conducted analysis is aligned with the behavior of complex systems represented by the evaluated datasets. For example, Section 4.2.2 illustrates that queen ants tend to avoid becoming a driver node, which is aligned with the queen's main responsibility to reproduce rather than decision making for the management of her colony. In fact, ant societies without any leader optimize their resources and workforce to maintain both exploration (e.g., to forage) and exploitation (e.g., to transfer and consume discovered food) [19]. An architecture that can foster a highly distributed control regime is required to achieve such optimizations. That is, relatively large groups of ants with a small number of members are capable of quickly spread and diffuse information in their colony.

In contrast with self-organized ant societies, the hierarchical organizations rely on managers to organize the company and spread information. Such a behavior is reflected by the analysis conducted in this dissertation. For instance, the identified SMDSs showed a few groups of employees with many members are capable of fully controlling the network. Also, a large intersection between those SMDSs exists sig-

naling the need for having critical employees (e.g., managers) to drive the spread of information. This indicates that the e-mails network operates under a highly centralized control regime as expected in hierarchical organizations.

7.1 Control Regimes

Researchers attempted to identify the control regime of complex networks based on the control categories of nodes (i.e., *critical*, *intermittent*, and *redundant*). Two general control regimes are commonly proposed namely the *distributed* and *centralized* regimes [13,14,26]. A *centralized* regime indicates that *redundant* nodes are dominant in a network. In contrast, a *distributed* regime indicates that *intermittent* and *critical* nodes are dominant. It is challenging to define a universal definition for the centralized and distributed control regimes due to the various states and size of systems. For example, the evaluated ant colony datasets did not provide any information on the state that a colony was during the data collection such as being in the stat of forage to search for food or in the state of exploitation to transport identified food. Also, the individual nodal dynamics are unknown such as the strength of state retention for an ant (e.g., how long the pheromone trails remain active or how strong is an ant's memory). Moreover, in ant datasets on average 1800 seconds were observed for data collection (Section 3.3).

The control regimes of complex networks are classified mainly based on the fraction of redundant nodes f_r [13, 14]. That is, a large f_r indicates a centralized regime is governing control because a small number of nodes could be selected as driver nodes. To discuss the control regime, Fig. 7.1 illustrated the fraction of nodes for each control type and the number of MDSs and SMDSs (the numerical results are presented in Table 4.1). Based on only considering f_r, ant colony 1-2 should be in a centralized control regime since $f_r(0.8) > f_i(0.2)$ as illustrated in Fig. 7.1. However, the author argues that colony 1-2 operate under a distributed control regime because 21 (N_{MDS}) unique groups with only $N_c = 2$ ants can fully control the colony. Also, consider only

15 ants ($= \lceil f_i * |V| \rceil$) could be selected in minimum driver set (i.e., 15 intermittent nodes). Even assuming that some ants are never exposed to any external interventions (i.e., permanently stay inside their colony), still 15 ants are about 20% of the colony (of 72 ants) that have the ability to control the colony with groups of $N_c = 2$ ants. Therefore, in an attempt to formalize the control regime of complex networks the author proposes Statement 1.

Statement 1 *A centralized regime refers to the capacity that relatively a few groups of nodes with a large intersection can make a network structurally controllable. On the other hand, a distributed control regime indicates that relatively many groups of nodes with a small number of members can make a network structurally controllable.*

Based on Statement 1, except colony 6-1 all ant colonies were in a distributed control regime since relative to their networks size, they have a large number of

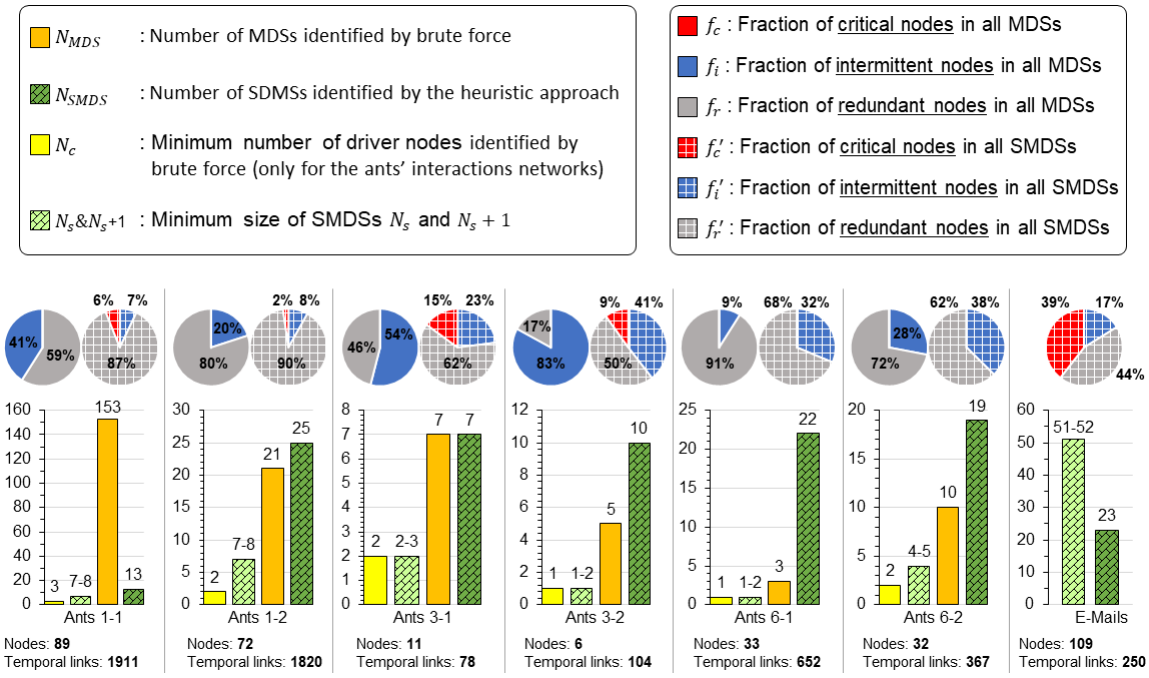

Fig. 7.1. MDSs and SMDSs characteristics for ants and emails network.

MDSs N_{MDS} and a small N_c. In colony 6-1, as presented in Fig. 7.1 $N_c = 1$ and $N_{MDS} = 3$ that is only three ants formed three MDSs out of 33 ants, which clearly demonstrate a centralized control regime. On the other hand, the e-mails network operates under a highly centralized control regime because a large percentage of the nodes belong to the *critical* node category (39% as depicted in Fig. 7.1).

In Fig. 7.1, the author illustrates the number of SMDSs N_{SMDSs} with size N_s and $N_s + 1$ as well as the fractions of nodes control types based on the identifies SMDSs denoted by f_c', f_i', and f_r'. The reasons for providing SMDSs of size N_s and $N_s + 1$ is to improve the accuracy of f_c', f_i', and f_r'. According to Statement 1 based on SMDSs except colony 1-1 all other colonies are in a distributed control regime as 13 SMDSs with size $N_s = 7$ and $N_s = 8$ are identified.

The author notes that applying Statement 1 needs to be done with caution because a colony can be considered to operate under a centralized approach using MDSs, however the same colony can be classified as a distributed control regime based on SMDSs. For example, based on SMDSs colony 6-1 is operating under a centralized regime since $N_c = 1$ and $N_{MDSs} = 3$. But, the heuristic approach identified 22 SMDSs with sizes $Ns = 1$ and $Ns = 2$, which indicate the network is operating under a distributed control regime.

7.2 Future Recommendations

A systematic way to characterize distributed and centralized control regimes is needed. Moreover, the proposed research in this dissertation paves the way to design targeted interventions in the structure of networks to alter their control regime. Similarly, one can design methods to alter the MCSs of a node by tailoring a series of interventions.

7.3 Conclusion

Temporal networks enable to explicitly study systems with changing topologies by capturing the order of interactions (i.e., time dimension) in the network's structure. This dissertation focuses on the structural controllability of temporal networks, which is the study of how to drive the state of networks from an initial state to a targeted state in finite time. In Chapter 4, the author showed finding a Minimum Driver node Set (MDS) is computationally prohibitive for temporal networks. Hence, the author proposed a heuristic approach to efficiently identify Suboptimal Minimum Driver node Sets (SMDSs) based on the Maximum Controllable Subspace (MCS) concept.

In Chapter 5, the author showed a node can have an exponential number of MCSs and proposed a branch and bound algorithm to identify k number of MCSs for a driver node. The algorithm enables to further study the behavior of temporal networks from the controllability perspective. In Chapter 6, the author evaluated the proposed branch and bound algorithm on a variety of temporal networks and observes that the MCSs-intersection of a node converge with identifying $k = 200$ MCSs. Also, the author analyzed the relationship between MCSs-intersections of nodes and driver node sets (MDSs and SMDSs).

REFERENCES

[1] C.-T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[2] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, p. 167, 2011.

[3] P. Holme and J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97 – 125, 2012, temporal Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0370157312000841

[4] I. Scholtes, "When is a network a network?: Multi-order graphical model selection in pathways and temporal networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: ACM, 2017, pp. 1037–1046. [Online]. Available: http://doi.acm.org/10.1145/3097983.3098145

[5] M. Pósfai and P. Hövel, "Structural controllability of temporal networks," *New Journal of Physics*, vol. 16, no. 12, p. 123055, 2014. [Online]. Available: https://doi.org/10.1088%2F1367-2630%2F16%2F12%2F123055

[6] S. Strogatz, *Sync: The emerging science of spontaneous order*. Penguin UK, 2004.

[7] A. E. Motter, C. Zhou, and J. Kurths, "Enhancing complex-network synchronization," *EPL (Europhysics Letters)*, vol. 69, no. 3, p. 334, 2005.

[8] L. Carlucci, G. Ciani, and D. M. Proserpio, "Polycatenation, polythreading and polyknotting in coordination network chemistry," *Coordination Chemistry Reviews*, vol. 246, no. 1-2, pp. 247–289, 2003.

[9] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.

[10] D. Makowiec, "The heart pacemaker by cellular automata on complex networks," *Cellular Automata*, pp. 291–298, 2008.

[11] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, no. 3, pp. 211–223, Aug 2001. [Online]. Available: https://doi.org/10.1023/A:1011122126881

[12] S. Hosoe, "Determination of generic dimensions of controllable subspaces and its application," *IEEE Transactions on Automatic Control*, vol. 25, no. 6, pp. 1192–1196, Dec 1980.

[13] X. Zhang and Q. Li, "Altering control modes of complex networks based on edge removal," *Physica A: Statistical Mechanics and its Applications*, vol. 516, pp. 185 – 193, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437118312901

[14] T. Jia, Y.-Y. Liu, E. Csóka, M. Pósfai, J.-J. Slotine, and A.-L. Barabási, "Emergence of bimodality in controlling complex networks," *Nature Communications*, vol. 4, jun 2013. [Online]. Available: https://doi.org/10.1038/ncomms3002

[15] H. Kwakernaak and R. Sivan, *Linear optimal control systems*. Wiley-interscience New York, 1972, vol. 1.

[16] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[17] B. Blonder and A. Dornhaus, "Time-ordered networks reveal limitations to information flow in ant colonies," *PLOS ONE*, vol. 6, no. 5, pp. 1–8, 05 2011. [Online]. Available: https://doi.org/10.1371/journal.pone.0020298

[18] R. Michalski, S. Palus, and P. Kazienko, "Matching organizational structure and social network extracted from email communication," in *Business Information Systems*, W. Abramowicz, Ed., vol. 87. Springer Berlin Heidelberg, 2011, pp. 197–206.

[19] M. Mitchell, *Complexity: A guided tour*. Oxford University Press, 2009.

[20] G. B. West, J. H. Brown, and B. J. Enquist, "A general model for the origin of allometric scaling laws in biology," *Science*, vol. 276, no. 5309, pp. 122–126, 1997. [Online]. Available: http://science.sciencemag.org/content/276/5309/122

[21] P. T. Benavides, U. Diwekar, and H. Cabezas, "Controllability of complex networks for sustainable system dynamics," *Journal of Complex Networks*, vol. 3, no. 4, pp. 566–583, 2015. [Online]. Available: http://dx.doi.org/10.1093/comnet/cnu051

[22] B. Ravandi and I. Papapanagiotou, "A self-organized resource provisioning for cloud block storage," *Future Generation Computer Systems*, vol. 89, pp. 765 – 776, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17330121

[23] B. Ravandi and F. Mili, "Polarization and coherence in complex networks," *Journal of Computational Social Science*, Feb 2019. [Online]. Available: https://doi.org/10.1007/s42001-019-00036-w

[24] B. Ravandi and N. Jovanovic, "Impact of plate size on food waste: Agent-based simulation of food consumption," *Resources, Conservation and Recycling*, vol. 149, pp. 550 – 565, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921344919302459

[25] B. Ravandi and A. Ravandi, "Network-based approach for modeling and analyzing coronary angiography," *arXiv preprint arXiv:1909.02664*, 2019. [Online]. Available: https://arxiv.org/abs/1909.02664

[26] B. Ravandi, J. A. Springer, and F. Mili, "Identifying and using driver nodes in temporal networks," *Journal of Complex Networks*, 2019. [Online]. Available: http://dx.doi.org/10.1093/comnet/cnz004

[27] S. Arianos, E. Bompard, A. Carbone, and F. Xue, "Power grid vulnerability: A complex network approach," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 1, p. 013119, 2009. [Online]. Available: https://doi.org/10.1063/1.3077229

[28] J. M. Montoya, S. L. Pimm, and R. V. Solé, "Ecological networks and their fragility," *Nature*, vol. 442, p. 259, Jul 2006, review Article. [Online]. Available: http://dx.doi.org/10.1038/nature04927

[29] I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, and F. Schweitzer, "Causality-driven slow-down and speed-up of diffusion in non-markovian temporal networks," *Nature Communications*, vol. 5, p. 5024, Sep 2014, article. [Online]. Available: http://dx.doi.org/10.1038/ncomms6024

[30] A. Rizzo, M. Frasca, and M. Porfiri, "Effect of individual behavior on epidemic spreading in activity-driven networks," *Phys. Rev. E*, vol. 90, p. 042801, Oct 2014. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.90.042801

[31] M. Karsai, M. Kivelä, R. K. Pan, K. Kaski, J. Kertész, A.-L. Barabási, and J. Saramäki, "Small but slow world: How network topology and burstiness slow down spreading," *Phys. Rev. E*, vol. 83, p. 025102, Feb 2011. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.83.025102

[32] R. K. Pan and J. Saramäki, "Path lengths, correlations, and centrality in temporal networks," *Phys. Rev. E*, vol. 84, p. 016105, Jul 2011. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.84.016105

[33] S. I. Dimitriadis, N. A. Laskaris, V. Tsirka, M. Vourkas, S. Micheloyannis, and S. Fotopoulos, "Tracking brain dynamics via time-dependent network analysis," *Journal of Neuroscience Methods*, vol. 193, no. 1, pp. 145 – 155, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165027010004875

[34] T. M. Przytycka, M. Singh, and D. K. Slonim, "Toward the dynamic interactome: it's about time," *Briefings in Bioinformatics*, vol. 11, no. 1, pp. 15–29, 2010. [Online]. Available: http://dx.doi.org/10.1093/bib/bbp057

[35] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, no. 3, p. 035006, 2016.

[36] J. C. Nacher and T. Akutsu, "Analysis of critical and redundant nodes in controlling directed and undirected complex networks using dominating sets," *Journal of Complex Networks*, vol. 2, no. 4, pp. 394–412, 2014. [Online]. Available: http://dx.doi.org/10.1093/comnet/cnu029

[37] M. Egerstedt, "Complex networks: Degrees of control," *Nature*, vol. 473, no. 7346, pp. 158–159, may 2011. [Online]. Available: https://doi.org/10.1038/473158a

[38] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, "Exact controllability of complex networks," *Nature Communications*, vol. 4, p. 2447, Sep 2013, article. [Online]. Available: http://dx.doi.org/10.1038/ncomms3447

[39] X. Liu, L. Pan, H. E. Stanley, and J. Gao, "Controllability of giant connected components in a directed network," *Phys. Rev. E*, vol. 95, p. 042318, Apr 2017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.95.042318

[40] A. Li, S. P. Cornelius, Y.-Y. Liu, L. Wang, and A.-L. Barabási, "The fundamental advantages of temporal networks," *Science*, vol. 358, no. 6366, pp. 1042–1046, 2017. [Online]. Available: http://science.sciencemag.org/content/358/6366/1042

[41] R. Kalman, "Mathematical description of linear dynamical systems," *Journal of the Society for Industrial and Applied Mathematics Series A Control*, vol. 1, no. 2, pp. 152–192, 1963. [Online]. Available: https://doi.org/10.1137/0301010

[42] K. Murota, *Matrices and matroids for systems analysis*. Springer Science & Business Media, 2009, vol. 20.

[43] S. Pequito, V. M. Preciado, A.-L. Barabási, and G. J. Pappas, "Trade-offs between driving nodes and time-to-control in complex networks," *Scientific Reports*, vol. 7, p. 39978, jan 2017. [Online]. Available: https://doi.org/10.1038/srep39978

[44] C. Zhao, W.-X. Wang, Y.-Y. Liu, and J.-J. Slotine, "Intrinsic dynamics induce global symmetry in network controllability," *Scientific Reports*, vol. 5, pp. 8422 EP –, Feb 2015, article. [Online]. Available: http://dx.doi.org/10.1038/srep08422

[45] N. J. Cowan, E. J. Chastain, D. A. Vilhena, J. S. Freudenberg, and C. T. Bergstrom, "Nodal dynamics, not degree distributions, determine the structural controllability of complex networks," *PLOS ONE*, vol. 7, no. 6, pp. 1–5, 06 2012. [Online]. Available: https://doi.org/10.1371/journal.pone.0038398

[46] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte, "Memory in network flows and its effects on spreading dynamics and community detection," *Nature Communications*, vol. 5, pp. 4630 EP –, Aug 2014, article. [Online]. Available: http://dx.doi.org/10.1038/ncomms5630

[47] L. Zino, A. Rizzo, and M. Porfiri, "Modeling memory effects in activity-driven networks," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 4, pp. 2830–2854, 2018. [Online]. Available: https://doi.org/10.1137/18M1171485

[48] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian journal of Mathematics*, vol. 8, no. 3, pp. 399–404, 1956.

[49] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[50] "Manufacturing emails network dataset," http://konect.uni-koblenz.de/networks/radoslaw_email, accessed: 2018-06-03.

[51] X. Zhang, T. Lv, and Y. Pu, "Input graph: the hidden geometry in controlling complex networks," *Scientific Reports*, vol. 6, no. 1, nov 2016. [Online]. Available: https://doi.org/10.1038/srep38209

[52] Z.-L. Hu, Z. Shen, S. Cao, B. Podobnik, H. Yang, W.-X. Wang, and Y.-C. Lai, "Locating multiple diffusion sources in time varying networks from sparse observations," *Scientific Reports*, vol. 8, no. 1, p. 2685, 2018. [Online]. Available: https://doi.org/10.1038/s41598-018-20033-9

[53] M. Newman, *Networks: an introduction*. Oxford university press, 2010.