# MATRIX MINOR-BASED UPPER BOUNDING FORMULATIONS FOR DESIGNING WEIGHTED NETWORKS WITH MAXIMUM ALGEBRAIC CONNECTIVITY

A Thesis

by

NEELKAMAL SOMISETTY

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

| | |
|---|---|
| Chair of Committee, | Swaroop Darbha |
| Committee Members, | Sivakumar Rathinam |
| | Sergiy Butenko |
| | Harsha Nagarajan |
| Head of Department, | Andreas A. Polycarpou |

August 2021

Major Subject: Mechanical Engineering

ABSTRACT

In this thesis, the problem of maximizing algebraic connectivity is considered; an instantiation of this problem in the context of mechanical systems is as follows: We are given $n$ masses and a set $E$ of springs with each spring having three attributes - (1) cost, (2) the pair of masses it can connect, and (3) stiffness. The problem is to build a structure within a specified budget $B$ so that (a) it is connected, and (b) is as stiff as possible in the sense that the smallest non-zero natural frequency of the mechanical system is as high as possible. Algebraic connectivity in graph theory is an analog of the smallest non-zero natural frequency for such a connected, mechanical structure.

This problem may be thought of as a canonical problem in discrete system realization theory. It has several engineering applications in emerging areas such as control and localization of Unmanned Aerial Vehicles under resource constraints, air transportation systems, inference network design, among others. It is an NP-hard problem and, consequently, is non-trivial. This problem can be posed as a Mixed-Integer Semi-Definite Program (MISDP). Since it is a computationally difficult problem, developing formulations with tighter relaxations for the MISDP are useful as they can provide tight bounds, which in turn determine the computational time required by the Branch and Bound (B&B) solvers. For problem instances where it is difficult to determine the optimal solutions in a reasonable time, the upper bounds help establish posterior sub-optimality bounds for feasible solutions from heuristic methods. The primary novelty of this thesis is the refinement of prior MISDP formulation by adding constraints based on positive semi-definiteness of principal minors, and the subsequent relaxation of MISDP to find tighter upper bounds for the optimum.

The contributions of this thesis to the literature are as follows: (a) development of a relaxation of the MISDP based on $2 \times 2$ principal minors for upper bounding algebraic connectivity, (b) development of tighter upper bounds by implementing the iterative cutting plane algorithm on higher-order principal minors, and (c) development of a variant of the MISDP formulation based on the structure of optimal networks which results in a good feasible solution with better computational efficiency.

DEDICATION

To my Parents, *S.Suresh Kumar* and *S.Lakshmi Madhuri*.

## ACKNOWLEDGMENTS

First and foremost, I am grateful for the opportunity to collaborate with Dr.Swaroop Darbha, my research advisor. Special thanks to Dr.Harsha Nagarajan for guiding me through the project. I'd like to express my gratitude to Dr.Sivakumar Rathinam and Dr.Sergiy Butenko for being valuable members of my committee.

I'd want to thank all the members of the research group, past and present, for their valuable insights. Dr.Sai Krishna Kanth Hari, Dr.Mengke Liu, and Christopher Montez are among them. I'm also grateful to my friends Sonitha Reddy Yandra, Jaikrishna Soundararajan, Rangeesh Venkatesan, Kiran Adhithya Ramakrishnan, Sai Praneeth Devunuri, Riya Khurana, and Phani Arvind Vadali for supporting me through the course of study and also encouraging me to pursue my passions outside of academics.

Finally, I want to convey my gratefulness to my grandparents, parents, my sister Meghana Nandyalam, and all my cousins in the U.S. for their unwavering love and support.

## CONTRIBUTORS AND FUNDING SOURCES

### Contributors

This work was supported by a thesis committee consisting of Dr.Swaroop Darbha [advisor] and Dr.Sivakumar Rathinam of the Department of Mechanical Engineering, Dr.Sergiy Butenko of the Department of Industrial & Systems Engineering and Dr.Harsha Nagarajan from Los Alamos National Laboratory.

### Funding Sources

Graduate study was supported by the project, “TRIAD: Modeling and Discrete Optimization for Robust Network Design,” sponsored by Texas A&M University System & Los Alamos National Laboratory under the award M2101343-01-470411-00001.

NOMENCLATURE

| | |
|---|---|
| MISDP | Mixed-Integer Semi-Definite Program |
| MISOCP | Mixed-Integer Second Order Conic Program |
| MICP | Mixed-Integer Convex Program |
| MILP | Mixed-Integer Linear Program |
| SDP | Semi-Definite Program |
| PSD | Positive Semi-Definite |
| IP | Integer Program |
| LP | Linear Program |

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

# 1. INTRODUCTION

## 1.1 Algebraic connectivity

Algebraic connectivity plays a vital role in addressing an open problem in system realization theory, which has relevance for several engineering applications. A simple case of network synthesis problem, which is NP-hard [1] can be described as follows: given a weighted graph and a constant $q$, find a connected sub-graph with at most $q$ edges such that the smallest non-zero eigenvalue (or the algebraic connectivity) of the weighted Laplacian of the sub-graph is maximized.

Complex networks are encountered in various applications, such as in bio-medicine, altering the dynamic response of discrete and continuous systems, Very Large Scale Integrated (VLSI) circuits, as well as the co-ordination of the multi-agent systems like Unmanned Air/Ground Vehicles (UAV), to name a few. Robustness/rigidity of networks is a crucial concept in the study of complex networks. For example, in the formation control of connected networks, algebraic connectivity represents a robustness measure and characterizes the collective ability of the formation to maintain a desired interconnection despite the presence of significant errors in measurements, communication delays, and bounded perturbations on the system. Algebraic connectivity as a metric for robustness has gained considerable interest both from the graph-theoretic perspective [2] and an engineering perspective [3–6].

In robot localization applications, the network of robot-to-robot exteroceptive measurements is represented by a weighted graph called the Relative Position Measurement Graph (RPMG). The weight of an edge $\{i, j\}$ of RPMG, is a function of the noise co-variance in the relative position measurement between the $i^{th}$ and the $j^{th}$ robots in the collection. Each robot has a measurement of its velocity contaminated by a zero-mean Gaussian process with known co-variance. The problem is to pick at most $q$ relative position sensors to obtain the best possible accuracy in estimating the robot's position. It has been shown that the estimation error co-variance of the collection using a Kalman filter is a decreasing function of the algebraic connectivity of the Dirichlet Laplacian associated with RPMG and an increasing function of the velocity measurement noise co-variances [4]. Therefore, the positioning accuracy can be improved via topology synthesis by picking a graph corresponding to the algebraic connectivity of the Dirichlet Laplacian subject to any resource constraints that may be present. Also, algebraic connectivity plays a critical role in determining the transient response and string stability of vehicular formations [7]. Optimal topology synthesis for vehicular formations via maximizing algebraic connectivity is difficult but essential for the stability of the motion of vehicles and faster convergence rate in consensus problems.

In air transportation, the airport network connectivity must be robust to an unpredictable node or link failures arising from airline budget cuts, weather hazards, or economic policies [8]. Recently, several studies have shown that networks with higher algebraic connectivity are more robust towards route failures that may be caused due to bad weather, ground delay, and flow programs, and flight delays/cancellations [9, 10]. The problem in distributed inference networks [11] requires an additional constraint on wiring costs given by the sum of the smallest set of eigenvalues of the Laplacian to be less than a specified bound. A similar problem arises in the design of ad-hoc relay networks for Unmanned Vehicles (UVs) with an area coverage constraint [12]. An additional constraint on the graph's diameter is considered for maximizing the robustness of an air transportation network under limited legs itinerary constraints in [13, 14].

This network synthesis problem can be formulated as a Mixed-Integer Semi-Definite Program (MISDP), which is non-trivial to solve. It is compounded by the rapid increase in the size of the problem. Even for instances of moderate size involving eight nodes, if one were asked to pick only seven edges to form a connected structure, there are 262144 ($8^6$) combinations (for a graph with $n$ nodes, there are $n^{n-2}$ connected structures with $n-1$ edges). Furthermore, it is complicated due to the non-smooth and non-linear relationship between algebraic connectivity and the edge choices and weights. Hence, this thesis aims to develop formulations to produce tight bounds in a reasonable time, which in turn determine the computational time required by the Branch and Bound (B&B) solvers.

## 1.2 A review on the Laplacian matrix

A graph $G$ is represented as $G(V, E, w)$, where $V$ is a set of vertices, $E(\subset V \times V)$ denotes a set of edges, and $w : E \to \Re_+$ is a weight function. Let $n = |V|$ denote the number of vertices in graph $G$ and let $I_n \in R^{n\times n}$ be the identity matrix. Without any loss of generality, one can simplify the problem by allowing at most one edge between two vertices, and we can number the vertices arbitrarily. Let $i, j \in V$ and let $e_i$, $e_j$ correspond to the $i^{th}$, $j^{th}$ columns of $I_n$. Let $w_{ij}$ gives the weight of the edge $\{i, j\}$. If $w_1$ , $w_2$ are two vectors in the same vector space, we denote their tensor product by $w_1 \otimes w_2$ (or $w_1 w_2^T$ more informally) and their scalar or dot product by $w_1 \cdot w_2$ (or $w_1^T w_2$).

The graph Laplacian of $G$ is defined as:

$$L := \sum_{e=\{i,j\}\in E} w_{ij}(e_i - e_j) \otimes (e_i - e_j). \tag{1.1}$$

The component of $L$ in the $i^{th}$ row and $j^{th}$ column is given by $L_{ij}$ as shown below:

$$L_{i,j} = \begin{cases} -w_{ij}, & \text{if } i \neq j, \{i,j\} \in E, \\ \sum_{j:\{i,j\}\in E} w_{ij}, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{1.2}$$

The graph interpretation of a spring-mass system and an electrical system is shown in the latter part of this section, where the Laplacian matrix of the equivalent weighted graph of these systems is derived.

### 1.2.1 Spring-mass system and its equivalent Graph Laplacian

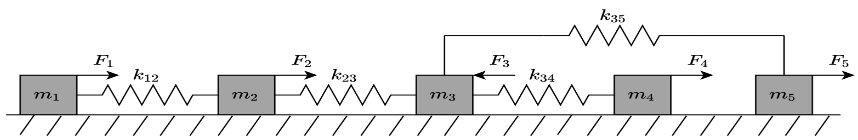


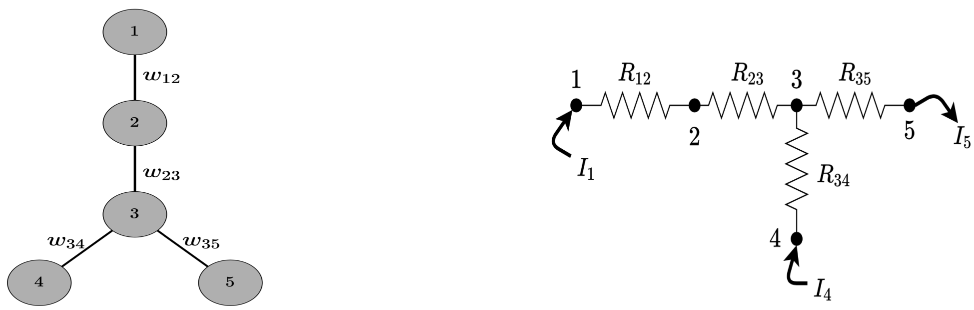

Figure 1.1: Spring-mass system.

As shown in Figure (1.1), the spring mass system is a simple five degree of freedom vibratory system. At equilibrium, every spring is assumed to have no deflection and is linear with a stiffness constant of $k_{ij}$ if it connects masses $m_i$ and $m_j$. If $y_i$ represents the displacement of the $i^{th}$ mass from its equilibrium position, by applying Newton's laws, we get the following equations of motion:

$$\begin{pmatrix} m_1 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 \\ 0 & 0 & 0 & m_4 & 0 \\ 0 & m_2 & 0 & 0 & m_5 \end{pmatrix} \begin{pmatrix} \ddot{y_1} \\ \ddot{y_2} \\ \ddot{y_3} \\ \ddot{y_4} \\ \ddot{y_5} \end{pmatrix} + \underbrace{\begin{pmatrix} k_{12} & -k_{12} & 0 & 0 & 0 \\ -k_{12} & k_{12}+k_{23} & -k_{23} & 0 & 0 \\ 0 & -k_{23} & k_{23}+k_{34}+k_{35} & -k_{34} & -k_{35} \\ 0 & 0 & -k_{34} & k_{34} & 0 \\ 0 & 0 & -k_{35} & 0 & k_{35} \end{pmatrix}}_{\text{Stiffness matrix}} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ -F_3 \\ F_4 \\ F_5 \end{pmatrix} \tag{1.3}$$

The stiffness matrix in equation (1.3) is the Laplacian matrix of its equivalent weighted graph shown in Figure (1.2a).

### 1.2.2 Electrical system and its equivalent Graph Laplacian




(a) Weighted graph (b) Electrical network

Figure 1.2: A weighted graph and its equivalent form as a electrical network.

An electrical network consisting of four resistors and five junctions is shown in Figure (1.2), which can be equivalently represented as a graph. Here, the junctions and resistors represent the vertices and edges, respectively. The weight of the edge is equivalent to the inverse of resistor's resistance.

The problem is to find out the voltages $V_1, V_2, V_3, V_4$ and $V_5$ at the corresponding vertices in the electrical network shown in Figure (1.2b). The amounts of current entering and leaving the network is known. Applying Ohm's law and Kirchhoff's current balance law at all the vertices, we have the following set of linear equations:

$$\frac{1}{R_{12}}(V_1 - V_2) = I_1, \tag{1.4a}$$

$$-\frac{1}{R_{12}}(V_1 - V_2) + \frac{1}{R_{23}}(V_2 - V_3) = 0, \tag{1.4b}$$

$$-\frac{1}{R_{23}}(V_2 - V_3) + \frac{1}{R_{35}}(V_3 - V_5) - \frac{1}{R_{34}}(V_4 - V_3) = 0, \tag{1.4c}$$

$$\frac{1}{R_{34}}(V_4 - V_3) = I_4, \tag{1.4d}$$

$$-\frac{1}{R_{35}}(V_3 - V_5) = -I_5. \tag{1.4e}$$

These equations can be expressed in the matrix form as shown:

$$\underbrace{\begin{pmatrix} \frac{1}{R_{12}} & -\frac{1}{R_{12}} & 0 & 0 & 0 \\ -\frac{1}{R_{12}} & \frac{1}{R_{12}} + \frac{1}{R_{23}} & -\frac{1}{R_{23}} & 0 & 0 \\ 0 & -\frac{1}{R_{23}} & \frac{1}{R_{23}} + \frac{1}{R_{34}} + \frac{1}{R_{35}} & -\frac{1}{R_{34}} & -\frac{1}{R_{35}} \\ 0 & 0 & -\frac{1}{R_{34}} & \frac{1}{R_{34}} & 0 \\ 0 & 0 & -\frac{1}{R_{35}} & 0 & \frac{1}{R_{35}} \end{pmatrix}}_{\text{Admittance matrix}} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{pmatrix} = \begin{pmatrix} I_1 \\ 0 \\ 0 \\ I_4 \\ -I_5 \end{pmatrix} \tag{1.5}$$

The admittance matrix in equation (1.5) is the Laplacian matrix of the graph in Figure (1.2a).

### 1.3 Algebraic connectivity as an objective function

Algebraic connectivity is chosen as an objective of maximization in this thesis. Motivation is provided through a linear mechanical system application where maximizing algebraic connectivity is intuitive.

Let the mechanical system consist of $n$ identical masses, and $|E|$ springs, where masses and linear springs represent the nodes and edges of a graph with the edge weights as the stiffness coefficients of the springs. The algebraic connectivity of the graph corresponds to the smallest non-zero natural frequency of the discrete mechanical system. Let $M, L$ respectively represent the mass and stiffness matrices respectively. The components of $L$ depend on the topology, $\mathbf{x}$, of connections of masses with the aid of springs. Let $e_0$ denote a vector, with every component being unity. If $\delta$, $F$ represent respectively the vectors of displacements and forces acting on the masses, then the governing equations corresponding to a given topology $\mathbf{x}$ may be compactly expressed as:

$$M\ddot{\delta} + L(\mathbf{x})\delta = F. \tag{1.6}$$

Let $F' = \{F : \|F\|_2 \leq 1, F \cdot e_0 = 0\}$ where $\|F\|_2$ represents the 2-norm of $F$. The condition $F \cdot e_0 = 0$ implies that the net force acting on the system of masses is zero; hence, the centroid of the system remains unchanged.

For a given graph, let $v_1, v_2, \ldots, v_n$ be the eigenvectors of $L(\mathbf{x})$ corresponding to eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then, $L(\mathbf{x})$ can be represented as:

$$L(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i v_i \otimes v_i. \tag{1.7}$$

Since $L(\mathbf{x})e_0 = 0$, $e_0$ is in the null space of $L(\mathbf{x})$ and hence, $\lambda_1 = 0$ and $v_1 = \frac{e_0}{\sqrt{e_0 \cdot e_0}}$. This eigenvector corresponds to a rigid body mode where all the displacements of all masses are same and the deflections in the springs are zero. A system is connected if and only if there exists at most a single rigid body mode. We can thus gather that the second smallest eigenvalue is positive ($\lambda_2 > 0$).

**Lemma 1.3.1.** *Let $\delta_s$ be the vector of displacements of masses of the mechanical system due to the forcing function $F$ . If $x \in \mathbf{x}$ , and the initial value of average displacement and velocity of all masses is zero, then [4, 12]:*

$$max_{F \in F'} ||\delta_s||_2 = \frac{1}{\lambda_2(L(x))}. \tag{1.8}$$

*Proof.* Since $F$ is a constant force, $\delta_s$ is a vector of constants and hence satisfies the equation shown:

$$L(x)\delta_s = F.$$

Let $F$ be decomposed along the eigenvectors $v_2, v_3, \ldots, v_n$ as:

$$F = \textstyle\sum_{i=2}^{n} \alpha_i v_i,$$

so that

$$\alpha_j = v_j \cdot F = v_j \cdot L(x)\delta_s = L(x)v_j \cdot \delta_s = \lambda_j v_j \cdot \delta_s.$$

From the assumption that the initial average displacement and velocity of all masses is zero, it follows that the average displacement and velocity of masses is zero through-out as:

$$e_0 \cdot [M\ddot{\delta} + L(x)\delta] = e_0 \cdot F = 0 \quad \Rightarrow e_0 \cdot \ddot{\delta} = 0.$$

Hence, $\delta_s$ cannot have a component along $v_1$ (equivalently, $e_0$). Since $x \in \mathbf{x}$,

$$\delta_s = \textstyle\sum_{j=2}^{n}(v_j \cdot \frac{F}{\lambda_j})v_j, \Rightarrow ||\delta_s||_2^2 = \sum_{j=2}^{n} \left(\frac{\alpha_j}{\lambda_j}\right)^2 \leq \frac{\sum_{j=2}^{n} \alpha_j^2}{\lambda_2^2} = \frac{1}{\lambda_2^2}.$$

Since the maximum is achieved when $F = v_2$, it follows that,

$$max_{F \in F'} ||\delta_s||_2 = \tfrac{1}{\lambda_2(L(x))}.$$

The maximum value of the 2-norm of forced response of the mechanical system can be minimized when $\lambda_2(L(x))$ is a maximum. For this reason, algebraic connectivity (or the second smallest eigenvalue of $L(x)$) is maximized. $\square$

## 1.4 Literature review

The problem of the maximization of algebraic connectivity is a simplified version of the system realization problem, which has been open for the past five decades. Maas first considered the problem of finding the desired graph with maximum algebraic connectivity in [15]. However, it was shown to be NP-hard recently [1]. Since this is an NP-hard problem, various algorithms to obtain optimal solutions for small-sized problems and heuristics are proposed in the literature. Special cases of the algebraic connectivity problem like edge design [16] and edge rewiring [17] to maximize algebraic connectivity are studied. Given a graph topology (say a spanning tree), choosing the weights of the edges in the topology so that its algebraic connectivity is maximized has been studied. Variants of this problem can be posed as a convex program subject to linear matrix inequality constraints, and iterative algorithms have been developed to solve them [18]. The problem of maximization of algebraic connectivity has recently received attention in the UAV literature; for example, a few of the relevant references are [7,19,20]. This problem has also gotten significant attention in the field of air transportation networks [9,10,21].

From the viewpoint of developing a systematic procedure to solve the algebraic connectivity problem to optimality, different types of cuts have been constructed. References [22,23] deal with the non-linear cuts for solving the mixed-integer second-order conic programs. Since conic programs are special instances of semi-definite programs, it is important to construct efficient cuts for semi-definite programs. Semi-definite cuts are developed in [24] by importing concepts from semi-definite programming, which are observed to be weak cuts. Recent work in [4,12,13,21,25] utilized eigenvector-based cuts to solve the algebraic connectivity problem. The same idea generalizes to other MISDPs as this technique is agnostic to the structure of the algebraic connectivity problem. These cuts were later observed to be efficient on generic MISDPs in [26,27]. However, tools for producing feasible solutions along with their suboptimality bounds within reasonable computational time are lacking. The earlier work in [12, 28] concerns the computation of such upper bounds. Nevertheless, these bounds are still not adequately tight to solve large scale problems effectively. In the context of power systems applications, for the problem of optimal power flow, the importance of the Second-Order Conic Program (SOCP) relaxations resulting from $2 \times 2$ principal minors and the higher-order minors (up to size $3 \times 3$) were considered in [29] and [30] respectively. While in [29], the constraints were SOCP representable, the constraints resulting due to higher-order minors in [30] were represented as non-linear, non-convex polynomials. More recently, [31] discusses the theoretical aspects and the importance of representing Positive Semi-Definite (PSD) constraints using principal minor characterization. However, to the best of our knowledge, the work presented in this thesis will be the first to develop upper bounding formulations based on principal minor-based characterizations for the problem of maximizing algebraic connectivity of weighted graphs. Moreover, we also exploit the higher-order minors of sizes up to $4 \times 4$, without an explicit evaluation of the polynomials corresponding to non-negativity of these minors, but enforce them in a cutting plane fashion using the eigenvector-based cuts. The book on convex optimization [32] provides an excellent overview of the algorithms required to solve convex semi-definite programs.

### 1.5 Research plan

From the literature study, it is clear that solving the problem of maximizing algebraic connectivity in a reasonable time is crucial. In order to design efficient methods for solving this problem, it is posed as an MISDP. As it is non-trivial and computationally difficult to solve this problem, the cutting plane method is utilized to solve the MISDP problem at hand. The basic idea of this method is to find an outer-approximation (relaxation) of the feasible set of the MISDP problem and solve the optimization problem over the outer-approximation (which we refer to as a relaxed MISDP). One may then iteratively refine the outer-approximation until the optimal solution of the outer-approximation is feasible for the MISDP. However, the run time for computing optimal solutions using either eigenvector cuts or semi-definite cuts grows drastically with the size of the problem. For the large instances where finding optimum is difficult, it is vital to compute upper bounds, which can act as a benchmark for comparing the quality of the feasible solutions from heuristic methods. Also, the cutting plane method's effectiveness relies heavily on the tightness of the upper bounds that one can obtain on maximum algebraic connectivity. Therefore, developing formulations for producing the tight upper bounds for the maximum algebraic connectivity in reasonable computational time is crucial.

The primary focus of this thesis is to develop tighter upper bounds for maximizing the algebraic connectivity. Utilizing the properties of the PSD matrices, relaxations of the MISDP formulation are developed to produce upper bounds for the maximum algebraic connectivity. A degree-constrained formulation developed in this thesis aids in the computation of sub-optimal solutions. The motivation behind it is to exploit the common structural feature of the optimal networks which led to a faster convergence.

# 2. MAXIMIZATION OF ALGEBRAIC CONNECTIVITY

In this section, the problem at hand is formulated as an MISDP and the algebraic connectivity of the Laplacian matrix is chosen as the objective function to maximize. Variants of this formulation are solved to optimality along the line using different techniques as the computation of solutions for combinatorial problems can be sensitive to the mathematical formulation of the problem. All optimization problems have been programmed using JuMP v0.21.3 [33] in Julia v1.3.1 [34]. All computational results presented in this thesis are computed with Mosek 9.2.16 [35] as the convex Semi-Definite Program (SDP) solver and Gurobi 8.1.1 [36] as a Mixed-Integer Linear Program (MILP) solver on a laptop with a 2.9 GHz 6-Core Intel Core i9 processor and 16 GB of RAM. This section is organized as follows: (1) formulation of the problem as an MISDP is first presented. (2) The MISDP is then solved by relaxing the constraints and the quality of the solutions are discussed, (3)Implementation of the cutting plane techniques on the relaxed MISDP to obtain optimal solutions for the original MISDP is presented, followed by the (4) introduction of a Mixed-Integer Convex Program (MICP) solver named Pajarito.jl [37] which will act as a reference for the performance of the formulations developed in next section.

## 2.1 Problem formulation of maximizing algebraic connectivity

An undirected graph $G$ is represented as $G(V, E, w)$, where $V$ is a set of vertices of the graph, $E \subset V \times V$ denotes a set of edges connecting vertices, and $w$ is a weight matrix, where $w_{ij}$ denotes the weight of the edge $e = \{i, j\} \in E$. If there is no edge connecting vertices $i, j \in V$, then the corresponding weight $w_{ij}$ is set to $\infty$. Let $x$ represent the vector of choice variables, $x_{ij}$ and $x_{ii} = 0$. Here, $x_{ij} \in \{0, 1\}$, which is a binary variable, refers to whether the edge is present or not. In simple words, if $x_{ij} = 1$, then this implies that edge is present in the network; otherwise, it is not. Without any loss of generality, one can simplify the problem by allowing at most one edge between two vertices, and also, we can number the vertices arbitrarily. Let $i, j \in V$ and let $e_i$, $e_j$ correspond to the $i^{th}$, $j^{th}$ columns of identity matrix $I_n$ where $n$ denotes the number of vertices in the graph.

As mentioned in the section (1.2), $L_{ij}$ is defined as:

$$L_{ij} := w_{ij}(e_i - e_j) \otimes (e_i - e_j). \tag{2.1}$$

The Laplacian matrix of the weighted graph is expressed as:

$$L(x) := \sum_{i \leq j, \{i,j\} \in E} x_{ij} L_{ij}. \tag{2.2}$$

Let $\lambda_1(L(x))(= 0) \leq \lambda_2(L(x)) \leq \cdots \leq \lambda_n(L(x))$ represents the eigenvalues of $L(x)$ and $v_1, v_2, \ldots, v_n$ be the corresponding eigenvectors. Let $q$, which is a positive integer, upper bounds the number of edges to be chosen. Then the problem of the maximizing algebraic connectivity is posed as:

$$\gamma^* = \quad \max \ \lambda_2(L(x)), \tag{2.3a}$$

$$s.t., \quad \sum_{i \leq j, \{i,j\} \in E} x_{ij} \leq q, \tag{2.3b}$$

$$x_{ij} \in \{0, 1\}, \quad \forall \{i, j\} \in E. \tag{2.3c}$$

The problem formulation in equation (2.3) is a non-linear binary program, which is difficult to compute. In the remainder of this section, the non-linear binary program is equivalently formulated as an MISDP using Eigen decomposition of $L(x)$.

### 2.1.1 Mixed-integer semi-definite program

The Laplacian matrix of the weighted graph $L(x)$ can be decomposed using its eigenvectors and eigenvalues, as shown:

$$L(x) := \sum_{i=1}^{n} \lambda_i(x)(v_i(x) \otimes v_i(x)) \tag{2.4a}$$

$$= \sum_{i=1}^{n} \lambda_i v_i v_i^T. \tag{2.4b}$$

For simplicity, the tensor product of $v_1$ and $v_2$ is equivalently represented as $v_1 v_2^T$ in equation (2.4). Let $e_0$ be the eigenvector corresponding to $\lambda_1(L(x)) = 0$ where $e_0 = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} e_i$, such that $e_0 \cdot e_0 = 1$. The equation (2.4) further reduces to as shown below:

$$L(x) = \lambda_2 v_2 v_2^T + .... + \lambda_n v_n v_n^T. \tag{2.5}$$

Adding $\lambda_2 e_0 e_0^T$ on both sides, it gives the following:

$$L(x) + \lambda_2 e_0 e_0^T = \lambda_2 e_0 e_0^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_n v_n v_n^T \tag{2.6a}$$

$$= \lambda_2 e_0 e_0^T + \sum_{i=2}^{n} \lambda_i v_i v_i^T. \tag{2.6b}$$

Finally, the equation (2.6) reduces to an inequality as:

$$L(x) + \lambda_2 e_0 e_0^T \succeq \lambda_2 \underbrace{\sum_{i=1}^{n} v_i v_i^T}_{I_n}, \tag{2.7a}$$

$$L(x) \succeq \lambda_2 (I_n - (e_0 e_0^T)). \tag{2.7b}$$

Now, the non-linear binary program in equation (2.3) can be equivalently expressed as an MISDP formulation which we will refer as $\mathcal{F}_1$ is as follows:

$$\gamma^* = \quad \max \ \gamma, \tag{2.8a}$$

$$s.t., \quad \sum\nolimits_{i \le j, \{i,j\} \in E} x_{ij} L_{ij} \succeq \gamma(I_n - (e_0 e_0^T)), \tag{2.8b}$$

$$\sum\nolimits_{i \le j, \{i,j\} \in E} x_{ij} \le q, \tag{2.8c}$$

$$x_{ij} \in \{0,1\}, \quad \forall \, \{i,j\} \in E. \tag{2.8d}$$

To show that $\gamma^* = \lambda_2(L(x^*))$, it is enough to prove that $\gamma^* \le \lambda_2(L(x^*))$ and $\gamma^* \ge \lambda_2(L(x^*))$, which can be proved using Rayleigh's inequality [12].

The simplest case of $\mathcal{F}_1$ corresponds to $q$ being $n-1$, where the feasible solutions are minimally constructed spanning trees. Therefore, the corresponding problem is to find a spanning tree that has the maximum algebraic connectivity:

$$\gamma^* = \quad \max \ \gamma, \tag{2.9a}$$

$$s.t., \quad \sum\nolimits_{i \le j, \{i,j\} \in E} x_{ij} L_{ij} \succeq \gamma(I_n - (e_0 e_0^T)), \tag{2.9b}$$

$$\sum\nolimits_{i \le j, \{i,j\} \in E} x_{ij} \le n - 1, \tag{2.9c}$$

$$x_{ij} \in \{0,1\}, \quad \forall \, \{i,j\} \in E. \tag{2.9d}$$

## 2.2 Relaxation of the MISDP

The feasible set of an MISDP formulation $\mathcal{F}_1$ is approximated by relaxing the few constraints to result in relaxed formulations. Here, the relaxed MISDP formulation is attained by relaxing the integer constraint or the semi-definite constraint, which can be solved using standard SDP or MILP solvers.

(i) **Binary relaxation:** The feasible set of $\mathcal{F}_1$ is expanded by replacing the integer variable with the continuous variable, i.e., $x_{ij} \in \{0, 1\}$ with $0 \leq x_{ij} \leq 1, \quad \forall\, i \leq j, \{i, j\} \in E$, which results in an SDP.

In the case of this relaxation, since it allows for fractional values of $x_{ij}$, there exist feasible solutions that violate connectivity constraint and correspond to weakly connected graphs. An example of a weakly connected graph is shown below in Figure (2.1) for a random weight matrix (Appendix A). From Figure (2.1), it is clear that for node eight, the sum of edge weights is less than unity, which violates the "cut" constraint for connectivity.

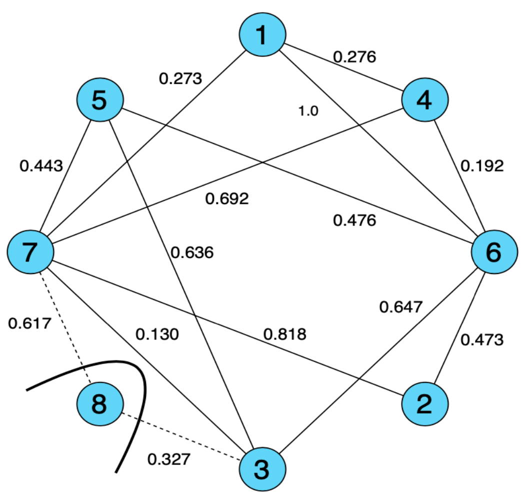



Figure 2.1: Weakly connected graph.

For a graph to be strongly connected, if $S$ is a strict subset of $V$, then there must be at least one edge between the set of nodes in $S$ and $V - S$. Adding these connectivity constraints will cut off the feasible solutions consisting of unconnected and weakly connected networks. Mathematically, these constraints can be expressed as:

$$\sum_{\{i,j\}\in\delta(S)} x_{ij} \geq 1, \quad \forall\, S \subset V, \tag{2.10}$$

where $\delta(S)$ (cutset of $S$) represents the subset of edges connecting nodes in $S$ with nodes in $V - S$.

However, these constraints are exponential in $|V|$. Magnanti and Wong's [38] flow formulation can be used to obtain an equivalently strong lifted formulation with a polynomial number of constraints.

Using the multi-commodity flow where $s$ is the source vertex and $f_{ij}^k$ be the $k^{th}$ commodity flowing from $i$ to $j$, the binary relaxed MISDP formulation with connectivity constraints is expressed as:

$$\begin{aligned}
\gamma_{sdp}^u = \quad & \max\ \gamma, && \text{(2.11a)}\\
s.t., \quad & \textstyle\sum_{i\leq j,\{i,j\}\in E} x_{ij} L_{ij} \succeq \gamma(I_n - (e_0 e_0^T)), && \text{(2.11b)}\\
& \textstyle\sum_{j\in V\setminus\{s\}} (f_{ij}^k - f_{ji}^k) = 1, \quad \forall\, k \in V and\ i = s, && \text{(2.11c)}\\
& \textstyle\sum_{j\in V} (f_{ij}^k - f_{ji}^k) = 0, \quad \forall\, \{i,k\} \in V and\ i \neq k, && \text{(2.11d)}\\
& \textstyle\sum_{j\in V} (f_{ij}^k - f_{ji}^k) = -1, \quad \forall\, \{i,k\} \in V and\ i = k, && \text{(2.11e)}\\
& f_{ij}^k + f_{ji}^k \leq x_{ij},\ \forall\, \{i,j\} \in E, \quad \forall\, k \in V, && \text{(2.11f)}\\
& 0 \leq f_{ij}^k \leq 1, \quad \forall\, \{i,j\} \in V,\ \forall\, k \in V, && \text{(2.11g)}\\
& \textstyle\sum_{i\leq j,\{i,j\}\in E} x_{ij} \leq n - 1, && \text{(2.11h)}\\
& 0 \leq x_{ij} \leq 1, \quad \forall\, \{i,j\} \in E. && \text{(2.11i)}
\end{aligned}$$

(ii) **Semi-definite constraint relaxation:** Polyhedral outer-approximation of the feasible set is achieved by replacing the semi-definite constraint by a set of linear inequalities, which results in an MILP as shown:

$$v_k \cdot (L(x) - \gamma(I_n - (e_0 e_0^T))v_k \geq 0, \quad \forall\, k = 1, 2, ...N.$$

Choosing the vectors to relax from the Fiedler vector set ($V_f$), a set consisting of Fiedler vectors of all $n^{n-2}$ feasible spanning trees, one can outer approximate the feasible set by Fiedler vectors. The Fiedler vector relaxation of the MISDP formulation is as shown:

$$\gamma_f^* = \quad \max\ \gamma, \tag{2.12a}$$

$$s.t., \quad v \cdot \left( \sum_{i \leq j, \{i,j\} \in E} x_{ij} L_{ij} \right) v \geq \gamma, \quad \forall\, v \in V_f, \tag{2.12b}$$

$$\sum_{i \leq j, \{i,j\} \in E} x_{ij} \leq n - 1, \tag{2.12c}$$

$$x_{ij} \in \{0, 1\}, \quad \forall\, \{i, j\} \in E. \tag{2.12d}$$

In this formulation, even for problems of moderate sizes ($n \geq 8$), it would be impractical to enumerate all the Fiedler vectors of feasible solutions to solve it to optimality. Hence, solving these relaxed formulations by outer approximating the feasible set with fewer Fiedler vectors from the set $V_f$ and maintaining the connectivity, one can readily obtain the upper bounds for the original MISDP problem.

### 2.2.1 Quality of the relaxed solutions

Table (2.1) summarizes the results of solving the binary relaxed MISDP formulation with connectivity constraints in equation (2.11) using the corresponding adjacency matrices in Appendix A. From these results, one can observe that upper bounds attained by solving the binary relaxed MISDP have a large gap from the optimal solutions. Moreover, the gap grows significantly with the size of the problem. It is concluded that the solutions provided by this binary relaxed formulation are very weak, and the gap is in order of magnitude higher than the optimal $\gamma^*$. Optimal solutions mentioned in Table (2.1) are computed by $n \times n$ eigenvector cuts method, which is discussed in subsection (2.3.1).

In the case of Fiedler vector relaxation, we observed that the quality of a solution depends on

| Nodes | $n = 8$ | | $n = 10$ | |
|---|---|---|---|---|
| Instance | $\gamma^*$ Optimal | $\gamma^u_{sdp}$ gap (%) | $\gamma^*$ Optimal | $\gamma^u_{sdp}$ gap (%) |
| 1 | 22.8042 | **105.91** | 34.2371 | **216.34** |
| 2 | 24.3207 | **132.15** | 41.4488 | **170.24** |
| 3 | 26.4111 | **130.00** | 37.7309 | **188.82** |
| 4 | 28.6912 | **127.93** | 41.4618 | **146.74** |
| 5 | 22.5051 | **118.82** | 34.3193 | **193.24** |
| 6 | 25.2167 | **130.66** | 39.9727 | **112.88** |
| 7 | 22.8752 | **136.94** | 36.1651 | **213.73** |
| 8 | 28.4397 | **113.15** | 42.3291 | **168.55** |
| 9 | 26.7965 | **125.67** | 39.4034 | **170.00** |
| 10 | 27.4913 | **106.41** | 34.9161 | **204.16** |

Table 2.1: Gaps between the optimal solutions and the upper bounds obtained by solving the binary relaxed MISDP formulation for the networks with eight and ten nodes.

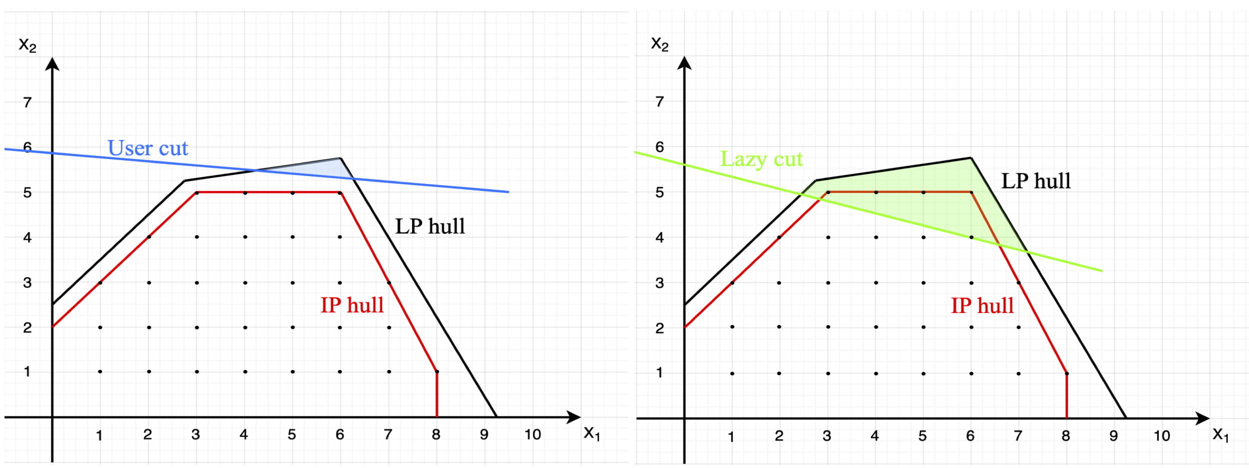
the topological structure of networks whose Fiedler vectors are chosen for outer-approximation. Hence, the quality of these relaxed solutions depends on choosing the Fiedler vectors based on two factors: (1) Fiedler vectors of the feasible solutions with higher $\lambda_2$, and (2) the number of Fiedler vectors. A systematic procedure of choosing Fiedler Vectors to relax the semi-definite constraint mentioned in [4] allows us to obtain better upper bounds.

In the next section, we focus on implementing different cutting plane techniques on the semi-definite constraint relaxed MISDP to obtain the optimal solution for the MISDP formulation $\mathcal{F}_1$. Relaxing the feasible set using Fiedler vectors we obtain upper bounds. Utilizing the cutting plane techniques, one can always tighten these upper bounds and eventually obtain optimal solutions. Therefore, after a brief introduction to the concepts of cutting plane techniques, we propose two different types of cutting planes to solve the problem of maximizing algebraic connectivity to optimality.

## 2.3 Cutting plane techniques

In optimization problems, cutting plane technique generally refers to an iterative refinement of the feasible set utilizing valid linear inequalities or "cutting planes." The cutting plane techniques can provide a monotonically decreasing sequence of upper bounds, which finally converges to the optimal algebraic connectivity value. If the optimal solution $(x^*, \gamma^*)$ for the relaxed problem is feasible for the original MISDP problem, it is also clearly optimal for the original MISDP problem; otherwise, one must refine the outer-approximation, via the introduction of the additional linear inequalities or "cuts." The outer-approximation is refined until the optimal solution is feasible for the original MISDP. This summarizes the algorithm's outline, which is discussed in detail in the latter part of this section. Implementation of the flow cuts for eliminating the weakly connected solutions is also discussed in this section.

These optimization problems are modeled using the JuMP package [33], where the problem is solved iteratively by adding cuts using a callback function after every iteration. Here, cuts can be modeled in two different ways, namely user cuts and lazy cuts.




(a) User cut (b) Lazy cut

Figure 2.2: User cut vs. Lazy cut.

The difference between user cuts and lazy cuts is shown in Figure (2.2); the feasible region of the linear program problem is the LP hull, and the IP hull is the smallest feasible set containing all feasible integer solutions. A user cut is a cut added by the user, where no integral solution is cut-off. In contrast to user cuts, lazy cuts are allowed to cut off integer-feasible solutions. From this, one can figure out that we are going to implement lazy cuts as we are cutting-off the integer solutions of the relaxed formulation, which are not feasible for the original MISDP.

These lazy cuts can be constructed in different ways using the properties of a PSD matrix to solve the relaxed MISDP formulation to optimality. Two types of cuts, namely eigenvector cuts, and semi-definite cuts, are discussed.

### 2.3.1 Eigenvector cuts

A symmetric positive semi-definite matrix has non-negative eigenvalues. Using this property, the $n \times n$ eigenvector cuts are constructed. Here, we outline a method to find the linear inequalities that cut-off solutions that are not feasible for the $\mathcal{F}_1$.

**Step 1:** The MILP in equation (2.13) resulting from the polyhedral outer-approximation of the MISDP is solved to optimality.

$$\gamma^* = \quad \max \ \gamma, \tag{2.13a}$$

$$s.t., \quad v_k \cdot (L(x) - \gamma(I_n - (e_0 e_0^T))v_k \geq 0, \quad \forall\, k = 1, 2, ...N, \tag{2.13b}$$

$$\sum\nolimits_{i \leq j, \{i,j\} \in E} x_{ij} \leq n - 1, \tag{2.13c}$$

$$\sum\nolimits_{\{i,j\} \in \delta(W)} x_{ij} \geq 1, \quad \forall\, W \subset V, \tag{2.13d}$$

$$x_{ij} \in \{0, 1\}, \quad \forall\, \{i, j\} \in E. \tag{2.13e}$$

The exponential number of cut-set constraints can be replaced by the multi-commodity flow constraints or the flow cuts, which is explained in subsection 2.3.3.

**Step 2:** Check the feasibility of the optimal solution $(x^*, \gamma^*)$ of the MILP for $\mathcal{F}_1$. If the semi-definite constraint is violated, one may readily use the eigenvector cut, i.e., if

$$\sum_{i \leq j, \{i,j\} \in E} x^*_{ij} L_{ij} - \gamma^*(I_n - (e_0 e_0^T)) \not\succeq 0. \tag{2.14}$$

The matrix on the left-hand side of the above inequality is not PSD iff there exists at least one negative eigenvalue. Then, a valid inequality is generated by the eigenvector ($v_{k+1}$) of the corresponding negative eigenvalue. The polyhedral outer-approximation is refined by augmenting an additional constraint that must be satisfied by any feasible solution to the $\mathcal{F}_1$ as:

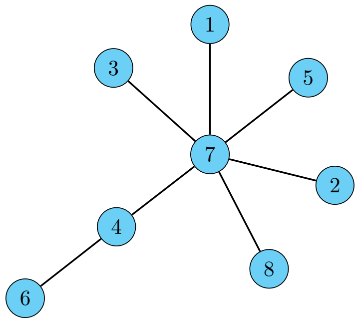
$$v_{k+1} \cdot (L(x) - \gamma(I_n - (e_0 e_0^T))v_{k+1} \geq 0. \tag{2.15}$$

This additional constraint ensures that the solution that was optimal for the previous relaxed MISDP will not be feasible now for the augmented set of inequalities, and the feasible set of the augmented MILP is a refined outer-approximation.

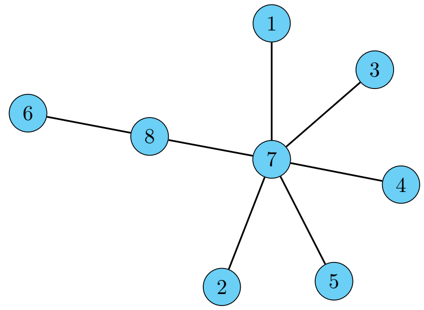
**Step 3:** Solve the augmented relaxed problem, i.e., solve the optimization problem over the feasible set of the refined approximation to get an updated optimal solution and go to Step 2.

This procedure is iterated until we obtain an optimal solution $(x^*, \gamma^*)$, which satisfies the semi-definite constraint; once the semi-definite constraint is satisfied, the optimal solution for the relaxed problem $(x^*, \gamma^*)$ will also be optimal for the $\mathcal{F}_1$. This algorithm is guaranteed to terminate in a finite number of iterations since the number of feasible solutions for this problem is finite ($n^{n-2}$ for a problem with $n$ nodes). Optimal networks for the instances with eight nodes obtained using $n \times n$ eigenvector cuts are shown in Figure (2.3).

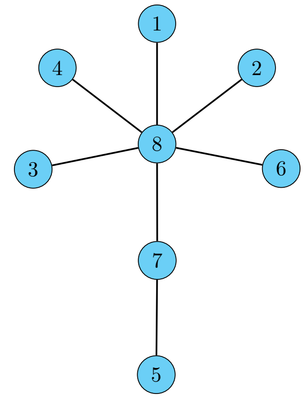
Table (2.2) summarizes the results of solving the relaxed MISDP formulation using the $n \times n$ eigenvector cuts (we will refer as $n \times n$ eigenvector cuts method) for the instances with eight and ten nodes in Appendix A. One can observe the exponential rise in the computational time with the problem size for the convergence of the $n \times n$ eigenvector cuts algorithm to optimality. Naturally, it will be a challenge to compute the optimal solutions for larger instances.



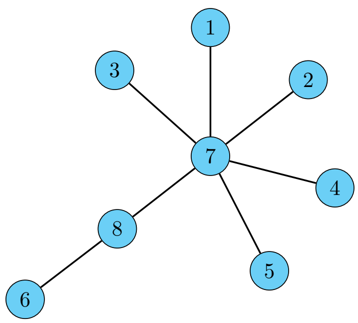
(a) $\gamma^* = 22.8042$



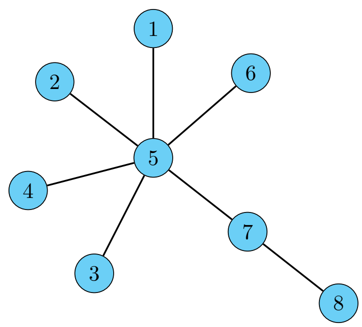
(b) $\gamma^* = 24.3207$



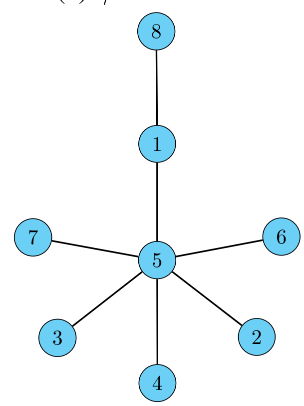
(c) $\gamma^* = 26.4111$



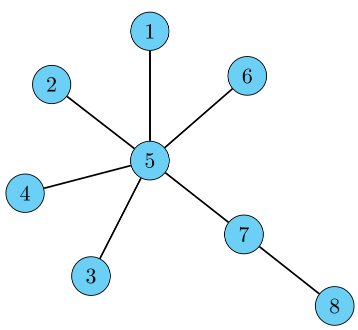
(d) $\gamma^* = 28.6912$



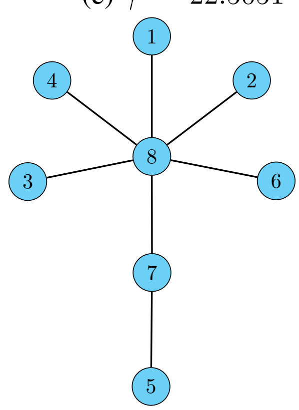
(e) $\gamma^* = 22.5051$



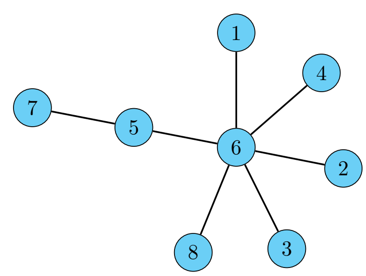
(f) $\gamma^* = 25.2167$



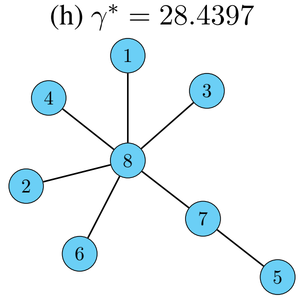
(g) $\gamma^* = 22.8752$



(h) $\gamma^* = 28.4397$



(i) $\gamma^* = 26.7965$



(j) $\gamma^* = 27.4913$

Figure 2.3: Optimal networks and maximum algebraic connectivity for the instances with eight nodes.

| Nodes | $n = 8$ | | | $n = 10$ | | |
|---|---|---|---|---|---|---|
| Instance | $\gamma^*$ Optimal | Time (sec) | Eigenvector cuts (# added) | $\gamma^*$ Optimal | Time (sec) | Eigenvector cuts (# added) |
| 1 | 22.8042 | 1.1 | 216 | 34.2371 | 613.1 | 2129 |
| 2 | 24.3207 | 3.4 | 306 | 41.4488 | 487.7 | 2049 |
| 3 | 26.4111 | 1.5 | 206 | 37.7309 | 673.6 | 2151 |
| 4 | 28.6912 | 1.1 | 207 | 41.4618 | 106.6 | 1034 |
| 5 | 22.5051 | 1.7 | 408 | 34.3193 | 283.2 | 2372 |
| 6 | 25.2167 | 3.8 | 521 | 39.9727 | 62.4 | 699 |
| 7 | 22.8752 | 1.9 | 390 | 36.1651 | 2532.7 | 2886 |
| 8 | 28.4397 | 1.1 | 201 | 42.3291 | 193.5 | 1237 |
| 9 | 26.7965 | 1.4 | 243 | 39.4034 | 155.0 | 1511 |
| 10 | 27.4913 | 1.0 | 133 | 34.9161 | 609.6 | 2160 |

Table 2.2: Maximum algebraic connectivity obtained using $n \times n$ eigenvector cuts for the networks with eight and ten nodes.

### 2.3.2 Semi-definite cuts

Semi-definite cuts are similar to the $n \times n$ eigenvector cuts. However, instead of eigenvector, we generate a suitable vector of unit length, $v \in R^n$, to yield a cutting plane. For a PSD matrix, all the diagonal elements are positive after the upper triangularization of the matrix [24]. Using this property, a process named super diagonalization is implemented to check whether the matrix is PSD or not after solving the MILP. In this process, once we obtain an optimal solution, proceeding in the order $i = 1, 2, \ldots, n$, we continue to zero out the elements in the $i^{th}$ column under the current $i^{th}$ diagonal element by performing elementary row operations using the $i^{th}$ row, so long as the diagonal elements encountered remain positive. Here, $L(x) - \gamma(I_n - (e_0 e_0^T))$ is represented by $L$, which is supposed to be a symmetric and PSD as it represents a connected graph. $L^i$ notation stands for the matrix $L$ after performing $i-1$ elementary row operations. $L^i[i:n, i:n]$ represents the sub-matrix from $L^i$ by selecting rows and columns from $i$ to $n$.

Starting with $L^1 = L^*$ (optimal solution) for $i = 1$, at the $i^{th}$ stage in this process, $i \in \{1, \ldots, n-1\}$, suppose that we have encountered all positive diagonal elements thus far, for the matrix $L^i \in R^{n\times n}$. Now for $i+1^{th}$ step, $L^{i+1}$ is computed as:

$$L^{i+1}[i+1:n, i+1:n] = L^i[i+1:n, i+1:n] - \frac{L^i[i+1:n, i] \cdot L^i[i+1:n, i]}{L^i_{ii}}, \quad (2.16a)$$

$$L^{i+1}[1:i, 1:i] = L^i[1:i, 1:i]. \quad (2.16b)$$

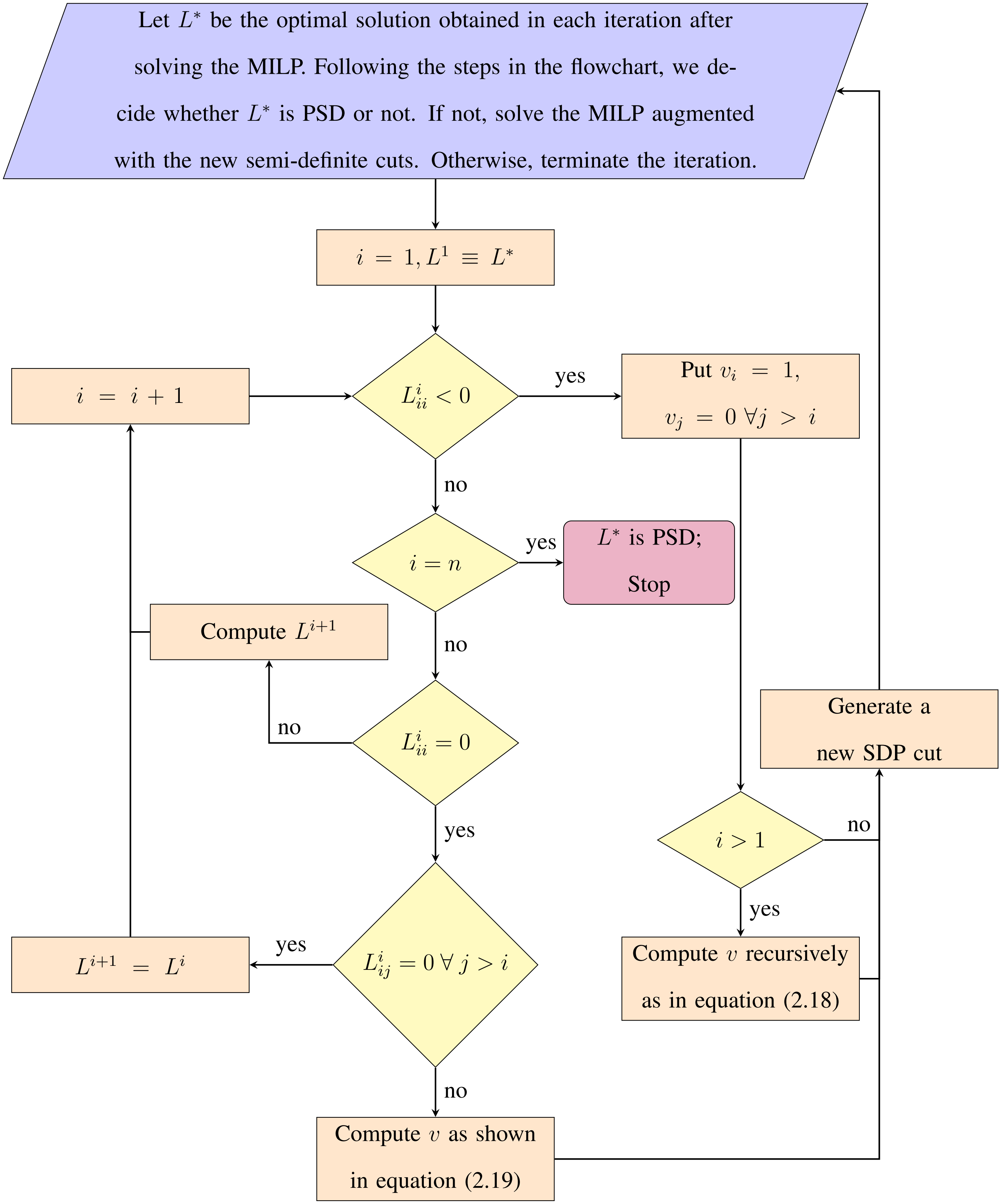
Once we encounter a negative diagonal element after zeroing out the elements in the $i^{th}$ column under the current $i^{th}$ diagonal element, we can conclude that the matrix is not PSD. For the cases where a diagonal element ($L^i_{ii}$) is zero, $L^{i+1} = L^i$ and proceed to next row if the whole row are zeros; otherwise if some element is non-zero in the $i^{th}$ row then the matrix $L$ is not PSD. This process is continued to the last diagonal element unless we found out the matrix $L$ is PSD or not.

If the matrix $L$ is not PSD, a semi-definite cut is generated to eliminate this optimal solution using a vector $v \in R^n$ as shown:

$$v \cdot (L(x) - \gamma(I_n - (e_0 e_0^T)))v \geq 0. \quad (2.17)$$

Once the semi-definite cut is generated after verifying the matrix $L$ is PSD or not, the MILP is augmented with the above new constraint. This augmented MILP is solved to obtain an updated optimal solution, and the same steps are followed to check whether it satisfies the semi-definite constraint. These iterations terminate when we obtain an optimal solution feasible for the $\mathcal{F}_1$. If it is feasible, then it is the optimal solution to maximizing algebraic connectivity problem. The whole process of super diagonalization, generating vector $v$ for the semi-definite cut is shown in the Figure (2.4) [24].

Figure 2.4: Flow-chart for the semi-definite cut generation.




Adapted from Sherali and Fraticelli, 2002 [24].

The vector $v$ is computed differently for different cases. Say that at $i^{th}$ step, we found out that diagonal element is negative, then the vector $v$ is computed as:

$$v_i = 1, \tag{2.18a}$$

$$v_j = 0, \quad \forall\, j > i, \tag{2.18b}$$

$$v_r = \begin{cases} \frac{-(v[r+1:n]\cdot L[r+1:n,r])}{L_{rr}}, & \text{if } L_{rr} \neq 0, \quad \forall\, r \in \{1, 2, \ldots, i-1\}, \\ 0, & \text{if } L_{rr} = 0, \quad \forall\, r \in \{1, 2, \ldots, i-1\}. \end{cases} \tag{2.18c}$$

For the cases where a diagonal element ($L^i_{ii}$) is zero, and some element ($L^i_{ij}$) in the row is non-zero, then vector $v$ is computed in a different way. To show the equations in a compact manner, lets represent $L^i_{ij}$ as $\theta$, $L^i_{jj}$ as $\phi$, and $\frac{\phi+\sqrt{\phi^2+4\theta^2}}{2}$ as $\lambda$. Using these terms, the vector $v$ is computed as shown:

$$v_i = \sqrt{\frac{1}{1+(\frac{\lambda}{\theta})^2}}, \tag{2.19a}$$

$$v_j = v_i \frac{\lambda}{\theta}, \tag{2.19b}$$

$$v_l = 0, \quad \forall\, l > i, \tag{2.19c}$$

$$v_r = \begin{cases} \frac{-(v[r+1:n]\cdot L[r+1:n,r])}{L_{rr}}, & \text{if } L_{rr} \neq 0, \quad \forall\, r \in \{1, 2, \ldots, i-1\}, \\ 0, & \text{if } L_{rr} = 0, \quad \forall\, r \in \{1, 2, \ldots, i-1\}, \end{cases}. \tag{2.19d}$$

Comparison of run times of solving the relaxed MISDPs using the corresponding adjacency matrices in Appendix A with $n \times n$ eigenvector cuts and semi-definite cuts are shown in Table (2.3). From those results, one can infer that eigenvector cuts are more efficient than semi-definite cuts. The computational time and the number of cuts added are less, implying that eigenvector cuts

| Instance | $\gamma^*$ Optimal | Time (sec) | Eigenvector cuts (# added) | Time (sec) | Semi-definite cuts (# added ) |
|---|---|---|---|---|---|
| 1 | 22.8042 | 1.1 | 216 | 5.8 | 1162 |
| 2 | 24.3207 | 3.4 | 306 | 6.1 | 1035 |
| 3 | 26.4111 | 1.5 | 206 | 5.9 | 1086 |
| 4 | 28.6912 | 1.1 | 207 | 3.5 | 1277 |
| 5 | 22.5051 | 1.7 | 408 | 5.9 | 1611 |
| 6 | 25.2167 | 3.8 | 521 | 15.1 | 2919 |
| 7 | 22.8752 | 1.9 | 390 | 10.3 | 1875 |
| 8 | 28.4397 | 1.1 | 201 | 5.1 | 1691 |
| 9 | 26.7965 | 1.4 | 243 | 8.5 | 1788 |
| 10 | 27.4913 | 1.0 | 133 | 3.0 | 938 |

Table 2.3: Eigenvector cuts vs. Semi-definite cuts for the graphs with eight nodes.

can tighten the outer-approximation more and converge to the optimal solution faster.

### 2.3.3 Cuts for connectedness

One can obtain the connected graphs using the multi-commodity flow [38], but it is a computationally arduous task as we have to deal with the polynomial number of constraints. An efficient way is to implement the flow cuts using the Ford-Fulkerson theorem [39, 40], once the relaxed MISDP is solved to optimality.

Using the concept of cutting plane method, one can generate the flow cuts. After solving a relaxed formulation, we check for the connectivity of the optimal solution ($x^*$). As explained in the introduction, a group of masses connected by springs attains only one rigid body mode if they are connected; this means that only the smallest eigenvalue of the Laplacian matrix is zero. Checking the eigenvalues of the Laplacian matrix of the weighted graph, we can determine whether the graph is connected or not.

If the graph is not connected, which is indicated by more than one zero eigenvalues, a flow cut is added between the unconnected components of the graph, as shown in equation (2.20).

$$\sum_{i \in C[1], j \in C[2]} x_{ij} \geq 1, \tag{2.20}$$

where $C[1], C[2]$ are the unconnected components of the graph.

## 2.4 Pajarito.jl for solving the problem of maximizing algebraic connectivity

Pajarito.jl [27, 37] is a mixed-integer convex programming (MICP) solver package written in Julia language [34]. MISDP and MISOCPs are two established sub-classes of the MICPs that Pajarito.jl can handle efficiently. The cutting plane algorithm implemented by Pajarito.jl itself is relatively straight-forward, while most of the computational burden is handled by the underlying MILP solver and the continuous convex conic solver. Pajarito.jl solves the MICP problems by constructing sequential polyhedral outer-approximations of the convex feasible set [27]. Pajarito.jl accesses state-of-the-art MILP solvers and continuous convex conic solvers through the MathOpt-Interface (MOI) [41].

Using Pajarito.jl, the optimal solutions to the MISDP problem of maximizing algebraic connectivity for the instances with eight and ten nodes are computed in better run times compared to the standard $n \times n$ eigenvector cuts method. However, as the problem size increases, the time for computing the optimal solution increases rapidly, in both Pajarito.jl and by using $n \times n$ eigenvector cuts method. Therefore, developing formulations with tighter relaxations for the MISDPs arising in the algebraic connectivity application are useful as they can provide tight bounds, which in turn determine the computational time required by the Branch-and-Bound (B&B) solvers. For large instances with unknown optimum, upper bounds also act as a proxy for determining the quality of the solutions obtained from heuristic methods.

In summary, this chapter has essentially dealt with the development of formulations and techniques to obtain optimal solutions and upper bounds; at times, the upper bounds were observed to be weak. However, it is clear that the time for computing optimal solutions grows rapidly with the size of the problem. Also, the bounds from the relaxed formulations are still not adequately tight to solve large scale problems effectively. In the next chapter, we utilize the various features of the PSD matrix and develop formulations to construct tighter upper bounds in reasonable run times.

# 3. MATRIX MINOR-BASED RELAXATIONS

In this section, the matrix minor-based relaxations are developed to obtain tighter upper bounds for the optimal algebraic connectivity in reasonable run times. The primary idea behind these relaxations is that a matrix is positive semi-definite if and only if all it's principal minors, which are sub-matrices obtained by selecting the same rows and columns, are non-negative. To compute the principal minors for the matrix in the semi-definite constraint, we formulate the MISDP formulation $\mathcal{F}_1$ in the lifted space of matrix variables. The latter part of the section deals with (1) Mixed-Integer Second Order Conic Program (MISOCP) formulation based on $2 \times 2$ principal minors, (2) upper bounds are further tightened by implementing eigenvector cuts on higher-order principal minors, (3) development of a variant formulation of $\mathcal{F}_1$ based on structure of the optimal networks to reduce the size of feasible set which leads to a faster convergence, and (4) the comparison of convergence rates of upper bounding formulations with respect to Pajarito.jl and $n \times n$ eigenvector cuts method.

## 3.1 MISDP formulation in the lifted space of matrix variables

The problem of maximizing algebraic connectivity with some simplifications for the purpose of implementation is expressed as an MISDP formulation $\mathcal{F}_1$ i.e.,

$$\begin{aligned}
\gamma^* = \quad & \max \ \gamma, && (3.1a)\\
s.t., \quad & \sum\nolimits_{i \leq j, \{i,j\} \in E} x_{ij} L_{ij} \succeq \gamma(I_n - (e_0 e_0^T)), && (3.1b)\\
& \sum\nolimits_{i \leq j, \{i,j\} \in E} x_{ij} \leq n - 1, && (3.1c)\\
& x_{ij} \in \{0, 1\}, \quad \forall \, \{i,j\} \in E, && (3.1d)
\end{aligned}$$

where $L_{ij}$ is defined as $w_{ij}(e_i - e_j) \otimes (e_i - e_j)$.

By simplifying the matrix inequality mentioned above in equation (3.1b), one can obtain the following:

$$\underbrace{\begin{pmatrix} \sum\limits_{\{1,j\}\in E} w_{1j}x_{1j} - \left(\frac{n-1}{n}\right)\gamma & -w_{12}x_{12} + \frac{\gamma}{n} & \cdots & -w_{1n}x_{1n} + \frac{\gamma}{n} \\ -w_{12}x_{12} + \frac{\gamma}{n} & \sum\limits_{\{2,j\}\in E} w_{2j}x_{2j} - \left(\frac{n-1}{n}\right)\gamma & \cdots & -w_{2n}x_{2n} + \frac{\gamma}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -w_{1n}x_{1n} + \frac{\gamma}{n} & -w_{2n}x_{2n} + \frac{\gamma}{n} & \cdots & \sum\limits_{\{n,j\}\in E} w_{nj}x_{nj} - \left(\frac{n-1}{n}\right)\gamma \end{pmatrix}}_{W} \succeq 0.$$

Let the matrix in the above inequality be represented by $W$. Then, the MISDP formulation $\mathcal{F}_1$, including the cut-set constraints, can be represented in the lifted space of matrix variables of $W$ as:

$$\begin{aligned} \gamma^* = \quad & \max\ \gamma, && \text{(3.2a)} \\ s.t., \quad & W \succeq 0, && \text{(3.2b)} \\ & W_{ii} = \sum_{\{i,j\}\in E} w_{ij}x_{ij} - \left(\frac{n-1}{n}\right)\gamma, \quad \forall\, i = 1,2,...n, && \text{(3.2c)} \\ & W_{ij} = W_{ji} = -w_{ij}x_{ij} + \frac{\gamma}{n}, \quad \forall\, \{i,j\} \in E, && \text{(3.2d)} \\ & \sum\nolimits_{i\le j,\{i,j\}\in E} x_{ij} \le n-1, && \text{(3.2e)} \\ & \sum\nolimits_{\{i,j\}\in \delta(S)} x_{ij} \ge 1, \quad \forall\, S \subset V, && \text{(3.2f)} \\ & x_{ij} \in \{0,1\}, \quad \forall\, \{i,j\} \in E. && \text{(3.2g)} \end{aligned}$$

The formulation in equation (3.2) will be referred as $\mathcal{F}_1{}'$. Using this representation, it is easier to construct the MISDP relaxations as it is easy to compute the principal minors compared to the previous formulation $\mathcal{F}_1$.

### 3.2 Minor-based relaxations

The basic definitions and propositions necessary for characterizing positive semi-definite and positive definite matrices [42] are provided below.

**Definition :** Given a matrix $A \in R^{n \times n}$, a minor of $A$ is a sub-matrix obtained by selecting only some rows $J_1 \subseteq [1, ..., n]$ and some columns $J_2 \subseteq [1, ..., n]$ of $A$. A principal minor $[A]_J$ is a minor (sub-matrix) obtained by selecting the same rows and columns of $A$, i.e., $J = J_1 = J_2$. Principal minor $[A]_J$ is a leading principal minor if $J = [1, ..., k]$ for any $1 \leq k \leq n$. Further, if any principal minor $[A]_J$ is said to be non-negative, then $det([A]_J) \geq 0$.

**Proposition :** Let $A \in R^{n \times n}$ be a symmetric matrix. $A$ is positive semi-definite (PSD) if and only if all principal minors are non-negative.

Sylvester's criterion [43] : $A \in R^{n \times n}$ is positive definite "if and only if" all leading principal minors are strictly positive, i.e., $det([A]_J) > 0, \quad \forall\, J = [1, ..., k]$ such that $1 \leq k \leq n$.

Using the above definition and proposition, we can formulate a relaxation using $2 \times 2$ principal minors and obtain tighter upper bounds to the maximum algebraic connectivity problem compared to binary or semi-definite relaxations. Further, the upper bounds are tightened by implementing eigenvector cuts on the $3 \times 3$ and $4 \times 4$ principal minors of $W$ matrix.

#### 3.2.1 MISOCP relaxation-based on $2 \times 2$ principal minors

Given the constraint in the MISDP formulation $\mathcal{F}_1{}'$ that $W$ is an PSD matrix, a relaxation is constructed based on the above preposition using only $2 \times 2$ principal minors, i.e., $[W]_J \quad \forall\, J \subseteq [1, ..., n]$, $|J| = 2$. A $2 \times 2$ principal minor is non-negative when it's determinant is non-negative which is given by:

$$W_{ij}^2 \leq W_{ii} W_{jj}, \quad \forall\, \{i, j\} \in E. \tag{3.3}$$

The MISDP is relaxed by replacing the semi-definite constraint in equation (3.2) by the inequality in equation (3.3), which is equivalently represented as an MISOCP formulation as follows:

$$\gamma_2^u = \quad \max\ \gamma, \tag{3.4a}$$

$$s.t., \quad W_{ii} = \sum_{\{i,j\}\in E} w_{ij}x_{ij} - \left(\frac{n-1}{n}\right)\gamma, \quad \forall\, i = 1,2,...n, \tag{3.4b}$$

$$W_{ij} = W_{ji} = -w_{ij}x_{ij} + \frac{\gamma}{n}, \quad \forall\,\{i,j\}\in E, \tag{3.4c}$$

$$W_{ij}^2 \leq W_{ii}W_{jj}, \quad \forall\,\{i,j\}\in E, \tag{3.4d}$$

$$\textstyle\sum_{i\leq j,\{i,j\}\in E} x_{ij} \leq n-1, \tag{3.4e}$$

$$\textstyle\sum_{\{i,j\}\in\delta(S)} x_{ij} \geq 1, \quad \forall\, S\subset V, \tag{3.4f}$$

$$x_{ij}\in\{0,1\}, \quad \forall\,\{i,j\}\in E. \tag{3.4g}$$

The $\gamma^*$ in equation (3.2) is upper bounded by $\gamma_2^u$. Gaps between the optimum and the upper bounds for the instances with eight and ten nodes, obtained by solving the MISOCP relaxation, using the corresponding adjacency matrices in Appendix A are shown in Table (3.1). Comparing the gaps from Tables (2.1) and (3.1), it can be stated that upper bounding formulation based on $2\times 2$ principal minors gives tighter upper bounds compared to the binary relaxation.

### 3.2.2 Relaxation based on outer-approximation

We can also formulate the relaxation based on the outer-approximation of $2\times 2$ principal minors and attain the same upper bounds as $\gamma_2^u$. In this formulation, cuts are implemented to refine the outer-approximation and attain upper bounds. As it involves cuts, this formulation assumes an iterative procedure where the following steps are followed in each iteration:

| Nodes | $n=8$ | | $n=10$ | |
|---|---|---|---|---|
| Instance | $\gamma^*$ Optimal | $\gamma_2^u$ gap (%) | $\gamma^*$ Optimal | $\gamma_2^u$ gap (%) |
| 1 | 22.8042 | **59.11** | 34.2371 | **103.01** |
| 2 | 24.3207 | **38.53** | 41.4488 | **83.87** |
| 3 | 26.4111 | **68.79** | 37.7309 | **70.67** |
| 4 | 28.6912 | **54.03** | 41.4618 | **54.41** |
| 5 | 22.5051 | **64.59** | 34.3193 | **109.56** |
| 6 | 25.2167 | **55.76** | 39.9727 | **46.03** |
| 7 | 22.8752 | **58.35** | 36.1651 | **85.69** |
| 8 | 28.4397 | **49.45** | 42.3291 | **66.84** |
| 9 | 26.7965 | **43.22** | 39.4034 | **73.23** |
| 10 | 27.4913 | **38.33** | 34.9161 | **70.51** |

Table 3.1: Gaps between the optimal solutions and the upper bounds obtained by solving the MISOCP relaxation based on $2 \times 2$ principal minors for the instances with eight and ten nodes.

**Step 1:** The optimization problem in equation (3.5) is solved for an optimal solution ($W^*$).

$$\gamma^u = \quad \max \ \gamma, \tag{3.5a}$$

$$s.t., \quad W_{ii} = \sum_{\{i,j\}\in E} w_{ij}x_{ij} - \left(\frac{n-1}{n}\right)\gamma, \quad \forall\, i = 1, 2, ...n, \tag{3.5b}$$

$$W_{ij} = W_{ji} = -w_{ij}x_{ij} + \frac{\gamma}{n}, \quad \forall\, \{i,j\} \in E, \tag{3.5c}$$

$$\sum\nolimits_{i\leq j,\{i,j\}\in E} x_{ij} \leq n-1, \tag{3.5d}$$

$$\sum\nolimits_{\{i,j\}\in\delta(S)} x_{ij} \geq 1, \quad \forall\, S \subset V, \tag{3.5e}$$

$$x_{ij} \in \{0,1\}, \quad \forall\, \{i,j\} \in E. \tag{3.5f}$$

**Step 2:** All $2 \times 2$ principal minors of $W^*$ are computed and checked for non-negativity using its determinant. If all minors are non-negative, then the optimal solution is the upper bound of $\gamma^*$, and iterations stop at this step. If at least one negative $2 \times 2$ principal minor exists, an inequality or a cut is added, which is derived using the lemma mentioned below.

**Lemma :** Let $f(W_{ij}, W_{ii}) = \frac{(W_{ij})^2}{W_{ii}}$. Then $(W_{ij})^2 \leq W_{ii}W_{jj}$ is satisfied iff the following infinite set of linear inequalities hold [44]:

$$f(\widehat{W}_{ij}, \widehat{W}_{ii}) + \frac{df(\widehat{W}_{ij}, \widehat{W}_{ii})}{dW_{ij}}(W_{ij} - \widehat{W}_{ij}) + \frac{df(\widehat{W}_{ij}, \widehat{W}_{ii})}{dW_{ii}}(W_{ii} - \widehat{W}_{ii}) \leq W_{jj}. \tag{3.6}$$

**Step 3:** If the $2 \times 2$ principal minor obtained by choosing $\{i, j\}$ rows and columns has negative determinant, then a valid cut is generated using the lemma. In the above lemma, the determinant inequality is replaced by a infinite set of linear inequalities. As it is not possible to add infinite inequalities, we add an inequality using the $W^*$ in equation (3.6) as a cut to eliminate this optimal solution as shown:

$$\frac{(W_{ij}^*)^2}{W_{ii}^*} + 2\frac{W_{ij}^*}{W_{ii}^*}(W_{ij} - W_{ij}^*) - \left(\frac{W_{ij}^*}{W_{ii}^*}\right)^2 (W_{ii} - W_{ii}^*) \leq W_{jj}, \tag{3.7}$$

which on simplifying we obtain the following inequality:

$$\frac{W_{ij}^*}{(W_{ii}^*)^2}(2\, W_{ii}^*\, W_{ij} - W_{ij}^*\, W_{ii}) \leq W_{jj}. \tag{3.8}$$

The MILP in equation (3.5) is augmented with the linear inequality in equation (3.8).

**Step 4:** Solve the augmented relaxed problem to get an updated optimal solution and go to Step 2.

This iterative procedure is terminated when there is no negative $2\times2$ principal minor exists in an optimal solution. This solution will be considered as the upper bound ($\gamma_2^u$) of the maximum algebraic connectivity($\gamma^*$).

### 3.2.3 Tightening of upper bounds

The upper bounds are further tightened by computing the principal minors of higher-order and implementing eigenvector cuts to ensure these minors are non-negative. As the size of principal minors computed increases, the gap between the upper bounds and the optimal solutions decreases. Principal minors of a size larger than two are checked for non-negativity by computing their eigenvalues. For the negative minors, cuts are added similar to $n \times n$ eigenvector cuts, which are explained in the following steps:

**Step 1:** The MILP in equation (3.5) is solved to optimality, and all principal minors of $W^*$ of a specific size are computed and verified for positive semi-definiteness by computing their eigenvalues.

**Step 2:** If any principal minor ($W^*_{pm}$) has at least one negative eigenvalue, then a cut is generated using its corresponding eigenvector ($v$) of the principal minor. The cut added is similar to $n \times n$ eigenvector cut which is shown below:

$$v \cdot (W_{pm})v \geq 0. \tag{3.9}$$

Now, the MILP in equation (3.5) is augmented with the new constraint.

**Step 3:** Solve the augmented relaxed problem to get an updated optimal solution and repeat from 2.

This procedure is continued until we obtain an optimal solution with all principal minors of the $W^*$ of a specific size are non-negative. Table (3.2) summarizes the results of the upper bounds obtained by computing $3 \times 3$ & $4 \times 4$ principal minors for the networks with eight and ten nodes. One can observe that $\gamma^* \leq \gamma_4^u \leq \gamma_3^u \leq \gamma_2^u$ from the Tables (3.1) and (3.2) for any instance in Appendix A.

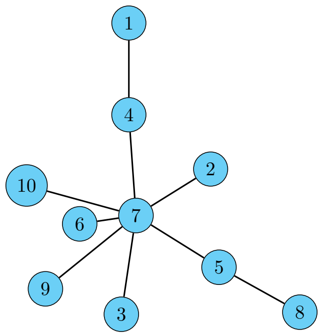
| Nodes | $n = 8$ | | | $n = 10$ | | |
|---|---|---|---|---|---|---|
| Instance | $\gamma^*$ Optimal | $\gamma_3^u$ gap (%) | $\gamma_4^u$ gap (%) | $\gamma^*$ Optimal | $\gamma_3^u$ gap (%) | $\gamma_4^u$ gap (%) |
| 1 | 22.8042 | **15.63** | **0.01** | 34.2371 | **48.97** | **18.87** |
| 2 | 24.3207 | **18.06** | **0.02** | 41.4488 | **36.60** | **6.76** |
| 3 | 26.4111 | **39.52** | **0.37** | 37.7309 | **39.25** | **6.45** |
| 4 | 28.6912 | **16.90** | **0.21** | 41.4618 | **15.31** | **0.20** |
| 5 | 22.5051 | **0.50** | **0.14** | 34.3193 | **43.83** | **13.63** |
| 6 | 25.2167 | **8.06** | **0.87** | 39.9727 | **12.34** | **3.49** |
| 7 | 22.8752 | **22.38** | **0.36** | 36.1651 | **45.59** | **19.22** |
| 8 | 28.4397 | **7.84** | **0.30** | 42.3291 | **29.73** | **0.90** |
| 9 | 26.7965 | **20.60** | **0.14** | 39.4034 | **23.96** | **7.06** |
| 10 | 27.4913 | **22.55** | **3.90** | 34.9161 | **35.72** | **28.10** |

Table 3.2: Gaps between the optimal solutions and the upper bounds obtained by solving the relaxation based on $3 \times 3$ and $4 \times 4$ principal minors for the networks with eight and ten nodes.

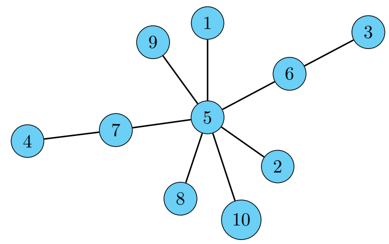
### 3.3 Degree-constrained formulation for maximizing algebraic connectivity

Since the MISDP formulation $\mathcal{F}_1{}'$ is not necessarily tractable for larger problem sizes, it presents an opportunity to study a variant of the formulation. The motivation behind the degree-constrained MISDP formulation (variant of $\mathcal{F}_1{}'$) is that the optimal solutions to the MISDP are clustered spanning trees. From the Figures (2.3) and (3.1), one can observe that there exists a node in every optimal network of $\mathcal{F}_1{}'$ whose connectivity is higher than the rest of all nodes (we will refer to this node as central node). Taking advantage of this feature, additional degree constraints are added to the MISDP formulation $\mathcal{F}_1{}'$, which leads to the degree-constrained MISDP formulation in equation (3.10). This formulation aims to search for the optimal solution in a smaller feasible set. This results in good feasible solutions for the MISDP formulation with better computational efficiency as the new feasible set is a subset of the original feasible set.

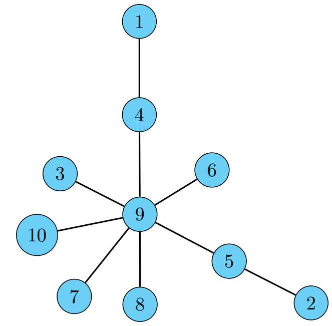
The problem is formulated to find a spanning tree with maximum algebraic connectivity such that there exists only one central node in the tree which has a degree of at least $(n - k)$, where $k$ $(\geq 1)$ is a positive. Let $d$ be a binary vector to determine the central node. Putting these words



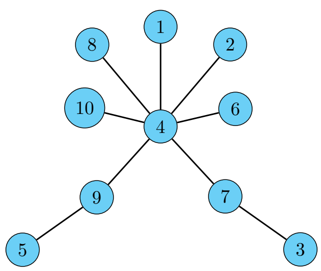
(a) $\gamma^* = 34.2371$



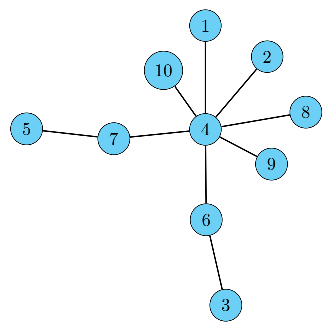
(b) $\gamma^* = 41.4488$



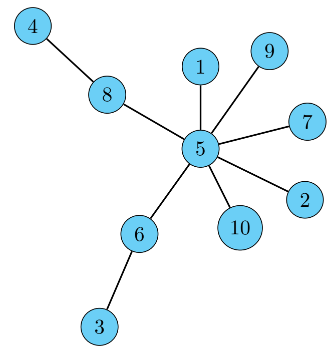
(c) $\gamma^* = 37.7309$



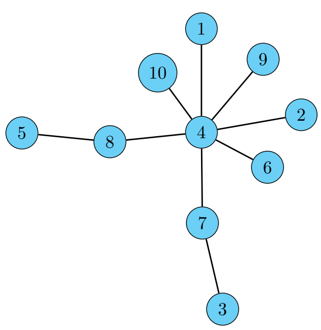
(d) $\gamma^* = 41.4618$



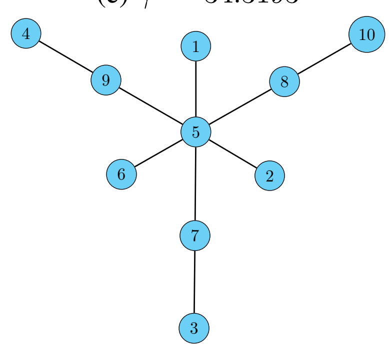
(e) $\gamma^* = 34.3193$



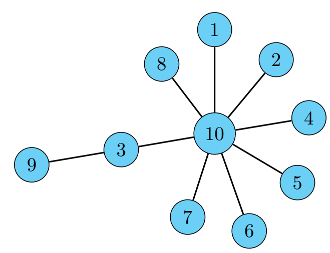
(f) $\gamma^* = 39.9727$



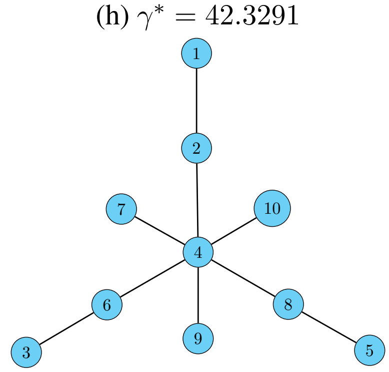
(g) $\gamma^* = 36.1651$



(h) $\gamma^* = 42.3291$



(i) $\gamma^* = 39.4034$



(j) $\gamma^* = 34.9161$

Figure 3.1: Optimal networks and maximum algebraic connectivity for the graphs with ten nodes.

into equations, the degree-constrained MISDP formulation is expressed as:

$$
\begin{aligned}
\gamma^*_{deg} = \quad & \max\ \gamma, && (3.10a)\\
s.t., \quad & W \succeq 0, && (3.10b)\\
& W_{ii} = \sum_{\{i,j\}\in E} w_{ij}x_{ij} - \left(\frac{n-1}{n}\right)\gamma, \quad \forall\, i = 1, 2, \ldots, n, && (3.10c)\\
& W_{ij} = W_{ji} = -w_{ij}x_{ij} + \frac{\gamma}{n}, \quad \forall\, \{i,j\} \in E, && (3.10d)\\
& \textstyle\sum_{i\leq j, \{i,j\}\in E} x_{ij} \leq n-1, && (3.10e)\\
& \textstyle\sum_{\{i,j\}\in \delta(S)} x_{ij} \geq 1, \quad \forall\, S \subset V, && (3.10f)\\
& \textstyle\sum_{j=1}^{n} x_{ij} \geq d_i(n-k-1)) + 1, \quad \forall\, i = 1, 2, \ldots, n, && (3.10g)\\
& \textstyle\sum_{i=1}^{n} d_i = 1, && (3.10h)\\
& x_{ij} \in \{0, 1\}, \quad \forall\, \{i,j\} \in E, && (3.10i)\\
& d_i \in \{0, 1\}, \quad \forall\, i = 1, 2, \ldots, n. && (3.10j)
\end{aligned}
$$

As a result of adding degree constraints to the relaxed MISDP formulation and solving it using $n \times n$ eigenvector cuts we attain solutions in smaller computational times. The run times for solving the relaxed MISDP with and without degree constraints are compared in Table (3.3) for all instances in Appendix A with ten nodes. Here, $T_1$ and $T_2$ represents the run times of the $n \times n$ eigenvector cuts methods with and without degree constraints respectively. In the case of instances with twelve nodes, a lot of computation power and time is required to obtain optimal networks using $n \times n$ eigenvector cuts method. However, enforcing degree constraints, good feasible networks for the instances with twelve nodes shown in Figure (3.2) are obtained for $k$ equal to five, in reasonable computational times.

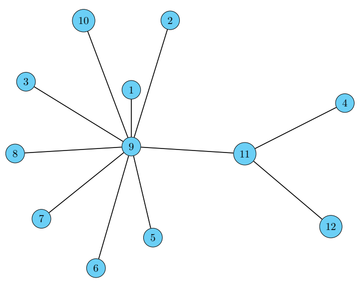
| Instance | $\gamma^* \ (= \gamma^*_{deg})$ Optimal | $T_1 \ (k = 4)$ (sec) | $T_2$ (sec) |
|---|---|---|---|
| 1 | 34.2371 | **16.3** | 613.1 |
| 2 | 41.4488 | **13.0** | 487.7 |
| 3 | 37.7309 | **13.0** | 673.6 |
| 4 | 41.4618 | **5.4** | 106.7 |
| 5 | 34.3193 | **5.0** | 283.2 |
| 6 | 39.9727 | **4.9** | 62.4 |
| 7 | 36.1651 | **22.5** | 1395.0 |
| 8 | 42.3291 | **7.0** | 193.5 |
| 9 | 39.4034 | **12.2** | 155.0 |
| 10 | 34.9161 | **27.1** | 609.6 |

Table 3.3: Comparison of run times of solving the relaxed MISDP with and without degree constraints for the instances with ten nodes.

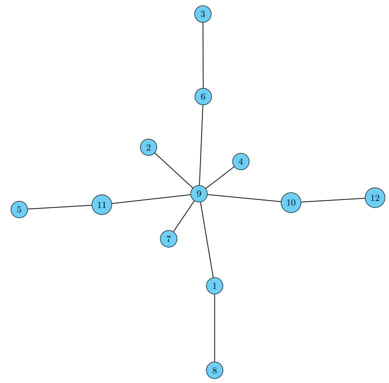
### 3.3.1 Quality of solutions and run times for various degree bounding parameters, $k$

The degree-constrained MISDP formulation in equation (3.10) finds a spanning tree with maximum algebraic connectivity such that there exists only one central node in the tree which has a degree of at least $(n - k)$, where $k \ (\geq 1)$ is a degree bounding parameter. By bounding the degree on the central node, we are constraining the feasible set of $\mathcal{F}_1{}'$. Therefore, the quality of the solution and the computational time depend on the value of $k$ chosen, as the feasible set changes with the $k$. The comparison of quality of the solutions and run times for different values of $k$ is shown in Figure (3.3) for all instances with ten nodes in Appendix A.

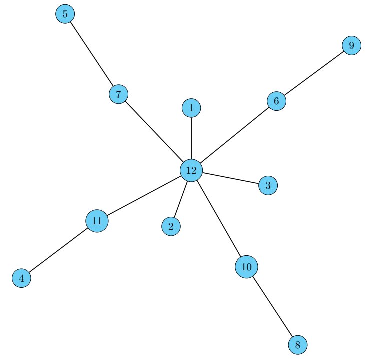
From the Figure (3.3), it can be observed that the quality of the solution for all instances of ten nodes increases with the value of $k$ until the optimal solutions of $\mathcal{F}_1{}'$ are attained. Also, the run times grows rapidly with the value of $k$, as the feasible set size increases. Same trend is observed for larger instances. .



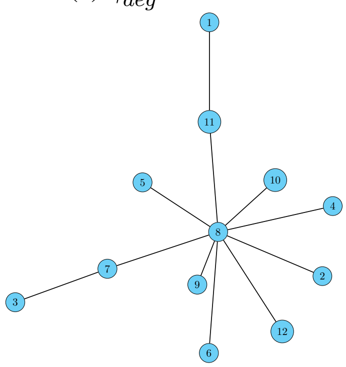
(a) $\gamma^*_{deg} = 54.0522$



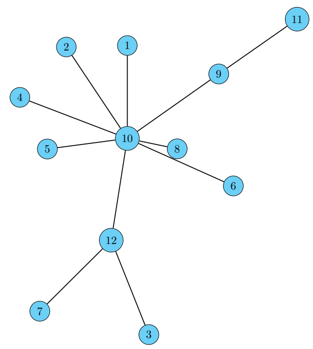
(b) $\gamma^*_{deg} = 53.2107$



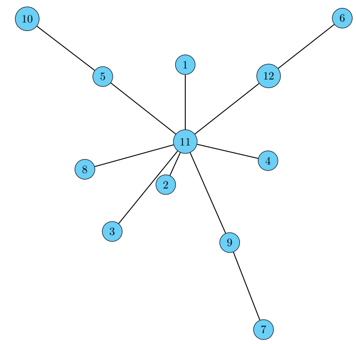
(c) $\gamma^*_{deg} = 47.2228$



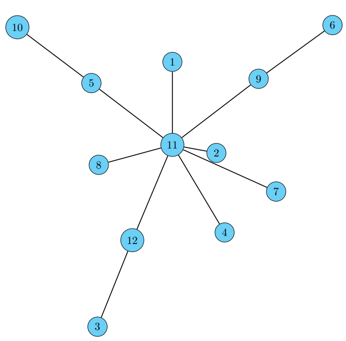
(d) $\gamma^*_{deg} = 43.9330$



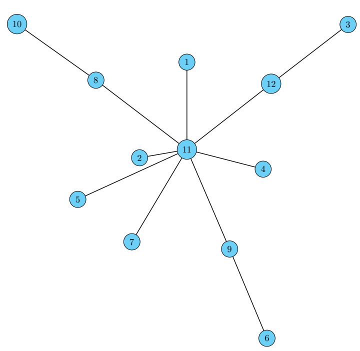
(e) $\gamma^*_{deg} = 51.1286$



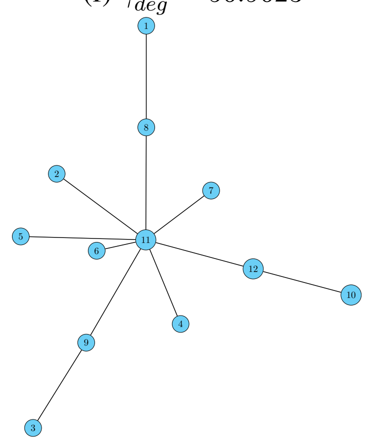
(f) $\gamma^*_{deg} = 56.9623$



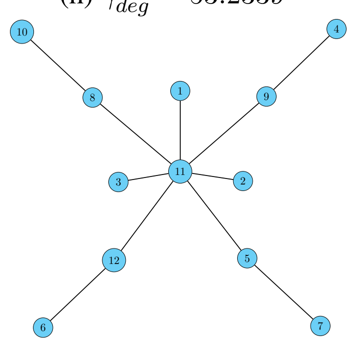
(g) $\gamma^*_{deg} = 57.2901$



(h) $\gamma^*_{deg} = 53.2339$



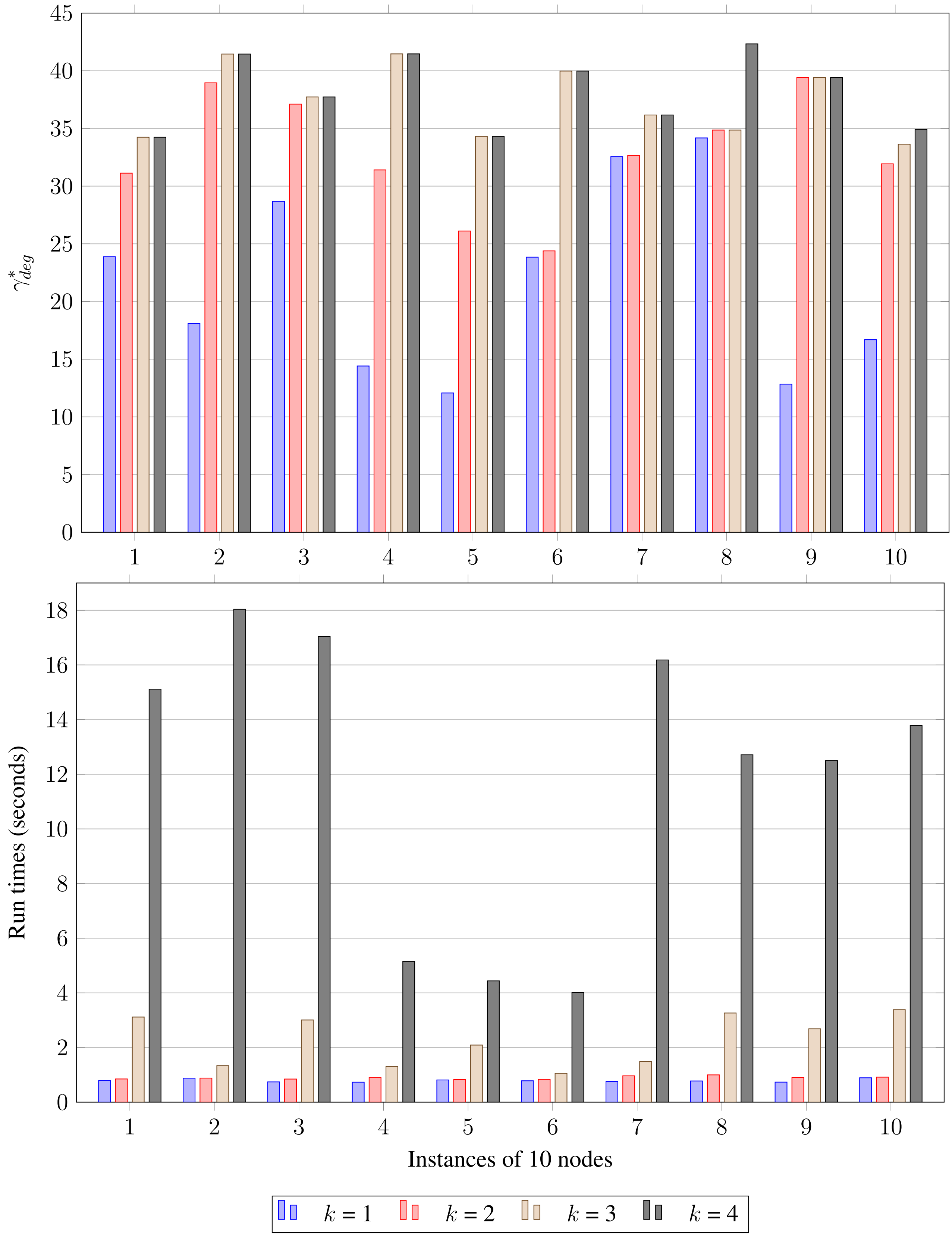
(i) $\gamma^*_{deg} = 53.5629$



(j) $\gamma^*_{deg} = 50.6987$

Figure 3.2: Feasible networks for the instances with twelve nodes obtained using degree-constrained formulation with $k = 5$.

Figure 3.3: Comparison of quality of the solutions and run times of solving the degree-constrained MISDP formulation with different values of $k$ for the instances with ten nodes.



### 3.4 Minor-based relaxations with degree constraints

The concept of the degree constraints can be extended to minor-based relaxed formulations. Adding these additional constraints to the minor-based relaxed formulation, we attain tighter upper bounds with a lesser computational time.

The formulation of the MISOCP relaxation along with the degree constraints is:

$$\gamma_{2_{deg}}^u = \quad \max\ \gamma, \tag{3.11a}$$

$$s.t., \quad W_{ii} = \sum_{\{i,j\}\in E} w_{ij}x_{ij} - \left(\frac{n-1}{n}\right)\gamma, \quad \forall\, i = 1,2,\dots,n, \tag{3.11b}$$

$$W_{ij} = W_{ji} = -w_{ij}x_{ij} + \frac{\gamma}{n}, \quad \forall\, \{i,j\} \in E, \tag{3.11c}$$

$$W_{ij}^2 \le W_{ii}W_{jj}, \quad \forall\, \{i,j\} \in E, \tag{3.11d}$$

$$\sum\nolimits_{i\le j,\{i,j\}\in E} x_{ij} \le n-1, \tag{3.11e}$$

$$\sum\nolimits_{\{i,j\}\in\delta(S)} x_{ij} \ge 1 \quad \forall\, S \subset V, \tag{3.11f}$$

$$\sum\nolimits_{j=1}^{n} x_{ij}, \ge d_i(n-k-1)) + 1, \quad \forall\, i = 1,2,\dots,n, \tag{3.11g}$$

$$\sum\nolimits_{i=1}^{n} d_i = 1, \tag{3.11h}$$

$$x_{ij} \in \{0,1\}, \quad \forall\, \{i,j\} \in E, \tag{3.11i}$$

$$d_i \in \{0,1\}, \quad \forall\, i = 1,2,\dots,n. \tag{3.11j}$$

Gaps between the optimal solutions and the upper bounds obtained from solving the MISOCP relaxation with and without degree constraints are compared in Table (3.4a). One can infer that $\gamma_{2_{deg}}^u \le \gamma_2^u$ by comparing those gaps. Similarly adding the degree constraints to relaxation based on $3\times 3$ and $4\times 4$, one can observe the similar trend in upper bound gaps from optimal solutions i.e., $\gamma_{3_{deg}}^u \le \gamma_3^u$ and $\gamma_{4_{deg}}^u \le \gamma_4^u$. Gaps between optimum and upper bounds obtained by higher-order minor-based relaxed formulations with and without degree constraints are presented in Tables (3.4b - 3.4c).

| Nodes | $n = 8$ $(k = 2)$ | | $n = 10$ $(k = 4)$ | |
|---|---|---|---|---|
| Instance | $\gamma_2^u$ gap (%) | $\gamma_{2_{deg}}^u$ gap (%) | $\gamma_2^u$ gap (%) | $\gamma_{2_{deg}}^u$ gap (%) |
| 1 | 59.11 | **11.24** | 103.01 | **70.51** |
| 2 | 38.53 | **12.92** | 83.87 | **70.36** |
| 3 | 68.79 | **16.74** | 70.67 | **62.58** |
| 4 | 54.03 | **11.20** | 54.41 | **36.59** |
| 5 | 64.59 | **3.07** | 109.56 | **45.76** |
| 6 | 55.76 | **13.42** | 46.03 | **21.67** |
| 7 | 58.35 | **14.63** | 85.69 | **53.95** |
| 8 | 49.45 | **1.93** | 66.84 | **57.67** |
| 9 | 43.22 | **1.61** | 73.23 | **36.51** |
| 10 | 38.33 | **1.86** | 70.51 | **63.61** |

(a) MISOCP relaxation based on $2 \times 2$ principal minors.

| Nodes | $n = 8$ $(k = 2)$ | | $n = 10$ $(k = 4)$ | |
|---|---|---|---|---|
| Instance | $\gamma_3^u$ gap (%) | $\gamma_{3_{deg}}^u$ gap (%) | $\gamma_3^u$ gap (%) | $\gamma_{3_{deg}}^u$ gap (%) |
| 1 | 15.63 | **0.33** | 48.97 | **37.80** |
| 2 | 18.06 | **0.54** | 36.60 | **23.79** |
| 3 | 39.52 | **2.15** | 39.25 | **17.17** |
| 4 | 16.90 | **0.76** | 15.31 | **15.31** |
| 5 | 0.50 | **0.50** | 43.83 | **25.54** |
| 6 | 8.06 | **2.10** | 12.34 | **8.85** |
| 7 | 22.38 | **1.50** | 45.59 | **38.86** |
| 8 | 7.84 | **0.12** | 29.73 | **6.38** |
| 9 | 20.60 | **0.85** | 23.96 | **11.37** |
| 10 | 22.55 | **0.86** | 35.72 | **31.34** |

(b) Relaxation based on $3 \times 3$ principal minors.

It can be concluded that adding the degree constraints to matrix minor-based relaxations, we generate tighter upper bounds in less computational times for the maximum algebraic connectivity problem of any size.

| Nodes | $n = 8$ $(k = 2)$ | | $n = 10$ $(k = 4)$ | |
|---|---|---|---|---|
| Instance | $\gamma_4^u$ gap (%) | $\gamma_{4_{deg}}^u$ gap (%) | $\gamma_4^u$ gap (%) | $\gamma_{4_{deg}}^u$ gap (%) |
| 1 | 0.01 | **0.01** | 18.87 | **1.58** |
| 2 | 0.02 | **0.02** | 6.76 | **0.15** |
| 3 | 0.37 | **0.37** | 6.45 | **0.27** |
| 4 | 0.21 | **0.21** | 0.20 | **0.20** |
| 5 | 0.14 | **0.14** | 13.63 | **1.67** |
| 6 | 0.87 | **0.87** | 3.49 | **1.95** |
| 7 | 0.36 | **0.36** | 19.22 | **0.07** |
| 8 | 0.30 | **0.03** | 0.90 | **0.90** |
| 9 | 0.14 | **0.14** | 7.06 | **0.38** |
| 10 | 3.90 | **0.13** | 28.10 | **10.06** |

(c) Relaxation based on $4 \times 4$ principal minors.

Table 3.4: Comparison of gaps between the optimal solutions and the upper bounds obtained by solving the minor-based relaxed formulations with and without degree constraints for the networks with eight and ten nodes.

## 3.5 Comparison of convergence rates

In this section, the convergence rates of solving the MISDPs by Pajarito.jl [37], $n \times n$ eigenvector cuts method and minor-based relaxation methods are compared. For these simulations, the time limit chosen is equal to the run time taken by the minor-based relaxation method to converge or 3600 seconds, whichever is less. The solutions attained by Pajarito.jl ($\gamma_p$) and $n \times n$ eigenvector cuts method ($\gamma_n$) are compared with the upper bounds obtained by the minor-based relaxation methods ($\gamma_2^u, \gamma_3^u, \gamma_4^u$). Tables (3.5) summarizes the gaps of the solutions from the optimal solutions, where $T_2, T_3, T_4$ are the time taken by the upper bounding formulation based on $2 \times 2$, $3 \times 3$ and $4 \times 4$ principal minors to converge, respectively.

| Instance $(n = 10)$ | $\gamma^*$ Optimal | $T_2$ (sec) | $\gamma_2^u$ gap (%) | $\gamma_p$ gap (%) | $\gamma_n$ gap (%) |
|---|---|---|---|---|---|
| 1 | 34.23 | 3.8 | 103.01 | 112.49 | 151.91 |
| 2 | 41.44 | 3.1 | 83.87 | 97.33 | 108.08 |
| 3 | 37.73 | 11.2 | 70.67 | 87.87 | 108.93 |
| 4 | 41.46 | 5.8 | 54.41 | 68.25 | 78.09 |
| 5 | 34.31 | 2.1 | 109.56 | 103.85 | 119.50 |
| 6 | 39.97 | 5.0 | 46.03 | 48.64 | 69.92 |
| 7 | 36.16 | 10.5 | 85.69 | 112.68 | 128.81 |
| 8 | 42.32 | 7.8 | 66.84 | 77.53 | 83.36 |
| 9 | 39.40 | 2.9 | 73.23 | 76.55 | 95.20 |
| 10 | 34.91 | 12.2 | 70.51 | 80.98 | 98.00 |

(a) Comparison of MISOCP formulation with respect to Pajarito.jl and $n \times n$ eigenvector cuts for instances with ten nodes.

| Instance $(n = 12)$ | $\gamma^*_{deg}$ $(k = 5)$ | $T_2$ (sec) | $\gamma_2^u$ gap (%) | $\gamma_p$ gap (%) | $\gamma_n$ gap (%) |
|---|---|---|---|---|---|
| 1 | 54.05 | 15.4 | 111.33 | 135.71 | 142.19 |
| 2 | 53.21 | 181.8 | 87.61 | 106.66 | 118.08 |
| 3 | 47.22 | 154.8 | 102.34 | 133.79 | 135.32 |
| 4 | 43.93 | 110.6 | 122.34 | 137.89 | 150.46 |
| 5 | 51.12 | 320.1 | 74.29 | 85.79 | 95.63 |
| 6 | 56.96 | 199.8 | 94.03 | 104.37 | 122.18 |
| 7 | 57.29 | 152.2 | 59.91 | 67.23 | 86.15 |
| 8 | 53.23 | 121.3 | 118.75 | 148.22 | 173.14 |
| 9 | 53.56 | 123.7 | 82.05 | 110.70 | 121.74 |
| 10 | 50.69 | 30.8 | 98.19 | 114.95 | 129.38 |

(b) Comparison of MISOCP formulation with respect to Pajarito.jl and $n \times n$ eigenvector cuts for instances with twelve nodes.

One can observe that for most instances with 10 nodes and 12 nodes problem, $\gamma_2^u \leq \gamma_p \leq \gamma_n$, $\gamma_3^u \leq \gamma_p \leq \gamma_n$ from Tables (3.5a - 3.5d). It can be implied that using MISOCP formulation and upper bounding formulation with $3 \times 3$ principal minors, sub-optimal solutions are computed faster compared to solving the MISDP with Pajarito.jl solver or $n \times n$ eigenvector cuts method.

| Instance ($n = 10$) | $\gamma^*$ Optimal | $T_3$ (sec) | $\gamma_3^u$ gap (%) | $\gamma_p$ gap (%) | $\gamma_n$ gap (%) |
|---|---|---|---|---|---|
| 1 | 34.23 | 26.8 | 48.97 | 65.97 | 115.44 |
| 2 | 41.44 | 23.3 | 36.60 | 54.12 | 79.49 |
| 3 | 37.73 | 28.8 | 39.25 | 62.95 | 90.65 |
| 4 | 41.46 | 32.3 | 15.31 | 25.27 | 38.43 |
| 5 | 34.31 | 13.5 | 43.83 | 52.42 | 86.82 |
| 6 | 39.97 | 24.5 | 12.34 | 0.00 | 34.67 |
| 7 | 36.16 | 37.4 | 45.59 | 87.17 | 105.95 |
| 8 | 42.32 | 23.5 | 29.73 | 45.41 | 62.86 |
| 9 | 39.40 | 39.5 | 23.96 | 19.75 | 49.77 |
| 10 | 34.91 | 37.1 | 35.72 | 63.85 | 79.73 |

(c) Comparison of upper bounding formulation with $3 \times 3$ principal minors with respect to Pajarito.jl and $n \times n$ eigenvector cuts for instances with ten nodes.

| Instance ($n = 12$) | $\gamma^*_{deg}$ $k = 5$ | $T_3$ (sec) | $\gamma_3^u$ gap (%) | $\gamma_p$ gap (%) | $\gamma_n$ gap (%) |
|---|---|---|---|---|---|
| 1 | 54.05 | 470.4 | 47.80 | 77.60 | 84.17 |
| 2 | 53.21 | 272.1 | 68.12 | 100.81 | 104.53 |
| 3 | 47.22 | 900.7 | 64.42 | 110.02 | 105.64 |
| 4 | 43.93 | 1246.4 | 64.83 | 112.92 | 106.87 |
| 5 | 51.12 | 1473.9 | 37.75 | 66.89 | 77.25 |
| 6 | 56.96 | 1333.4 | 48.05 | 78.80 | 88.36 |
| 7 | 57.29 | 674.2 | 29.54 | 48.82 | 60.00 |
| 8 | 53.23 | 1095.0 | 64.48 | 116.33 | 114.56 |
| 9 | 53.56 | 979.5 | 43.89 | 79.92 | 83.67 |
| 10 | 50.69 | 401.3 | 51.36 | 84.19 | 88.22 |

(d) Comparison of upper bounding formulation with $3 \times 3$ principal minors with respect to Pajarito.jl and $n \times n$ eigenvector cuts for instances with twelve nodes.

Comparing the upper bounding formulation with $4 \times 4$ principal minors with respect to Pajarito.jl and $n \times n$ eigenvector cuts for instances with ten nodes in Table (3.5e), Pajarito.jl seems to converge faster. However, as the problem size increases, all methods seem to converge at the same rate for most instances in Appendix A.

| Instance ($n = 10$) | $\gamma^*$ Optimal | $T_4$ (sec) | $\gamma_4^u$ gap (%) | $\gamma_p$ gap (%) | $\gamma_n$ gap (%) |
|---|---|---|---|---|---|
| 1 | 34.23 | 322.6 | 18.87 | 0.00 | 52.08 |
| 2 | 41.44 | 741.2 | 6.76 | 0.00 | 0.00 |
| 3 | 37.73 | 829.8 | 6.45 | 0.00 | 0.00 |
| 4 | 41.46 | 261.7 | 0.00 | 0.00 | 0.00 |
| 5 | 34.31 | 262.2 | 13.63 | 0.00 | 13.46 |
| 6 | 39.97 | 111.6 | 3.49 | 0.00 | 0.00 |
| 7 | 36.16 | 811.4 | 19.22 | 0.00 | 43.92 |
| 8 | 42.32 | 437.7 | 0.90 | 0.00 | 0.00 |
| 9 | 39.40 | 167.7 | 7.06 | 0.00 | 0.00 |
| 10 | 34.91 | 169.6 | 28.10 | 32.72 | 50.12 |

(e) Comparison of upper bounding formulation with $4 \times 4$ principal minors with respect to Pajarito.jl and $n \times n$ eigenvector cuts for instances with ten nodes.

| Instance ($n = 12$) | $\gamma_{deg}^*$ ($k = 5$) | Time limit (sec) | $\gamma_4^u$ gap (%) | $\gamma_p$ gap (%) | $\gamma_n$ gap (%) |
|---|---|---|---|---|---|
| 1 | 54.05 | 3600.0 | 58.75 | 47.88 | 52.66 |
| 2 | 53.21 | 3600.0 | 52.17 | 63.25 | 64.49 |
| 3 | 47.22 | 3600.0 | 80.11 | 88.01 | 87.71 |
| 4 | 43.93 | 3600.0 | 95.79 | 91.79 | 93.31 |
| 5 | 51.12 | 3600.0 | 61.41 | 60.85 | 66.56 |
| 6 | 56.96 | 3600.0 | 70.93 | 65.06 | 76.48 |
| 7 | 57.29 | 3600.0 | 40.19 | 28.35 | 38.99 |
| 8 | 53.23 | 3600.0 | 94.34 | 87.88 | 96.44 |
| 9 | 53.56 | 3600.0 | 66.38 | 61.56 | 65.12 |
| 10 | 50.69 | 3600.0 | 56.02 | 53.11 | 56.02 |

(f) Comparison of upper bounding formulation with $4 \times 4$ principal minors with respect to Pajarito.jl and $n \times n$ eigenvector cuts for instances with ten nodes.

Table 3.5: Comparison of convergence rates of solving the MISDPs by Pajarito.jl, $n \times n$ eigenvector cuts method and minor-based relaxation methods for instances with ten and twelve nodes.

# 4. SUMMARY AND CONCLUSIONS

## 4.1 Summary

In this thesis, we aimed at developing relaxations to compute tight upper bounds for a simplified version of an open problem in system realization theory; this problem has many applications in disparate fields of engineering. The underlying problem in the context of mechanical systems we considered was as follows: Given a collection of masses and a set of linear springs with a specified cost and stiffness, the problem was to determine an optimal connection of masses and springs so that the resulting structure was as stiff as possible. We showed that the structure is stiff when the second non-zero natural frequency of the interconnection is maximized under certain assumptions.

The network synthesis problem for maximizing algebraic connectivity (or the first non-zero eigenvalue of the weighted Laplacian matrix of a graph), an NP-hard problem, is formulated as an MISDP. Being a non-trivial problem, it is crucial to develop a systematic procedure to solve for optimum or to obtain good upper bounds. At present, the tools for producing feasible solutions within reasonable computational time and estimate the quality of the solutions they produce are lacking. To address this void in the literature, we developed relaxed formulations to produce upper bounds for the maximum algebraic connectivity problem.

We posed the problem of maximizing algebraic connectivity as an MISDP and utilized cutting plane techniques to solve. The basic idea of this method is to find a polyhedral outer-approximation of the feasible set of the MISDP problem and solve the optimization problem over the outer-approximation. If the optimal solution for the relaxed MISDP is feasible for the original MISDP problem, it is also clearly optimal for the original MISDP problem. Otherwise, we refined the outer-approximation via introducing the new linear inequalities or cuts until the optimal solution of the outer-approximation is feasible for the MISDP. Therefore, the proposed cutting plane method finds an optimal solution to the MISDP. However, the time for computing optimal solutions increases rapidly with the problem size.

Relaxing the feasible set by outer approximating the semi-definite constraint in the MISDP formulation with the $2 \times 2$ principal minors using Sylvester's criterion leads to an upper bound on the maximum algebraic connectivity. Based on this idea, we proposed MISOCP relaxation to produce upper bounds on the maximum algebraic connectivity. Further, these bounds are tightened by implementing eigen vector cuts on higher-order principal minors. We also proposed a formulation utilizing the fact that the optimal solutions to the MISDP are clustered spanning trees. Utilizing the characteristic feature of optimal networks, degree constraints are modeled and added to the MISDP formulation, which leads to good feasible solutions in less computational times. Later, these constraints are added to minor-based relaxations to produce better upper bounds with better computational efficiency.

### 4.2 Conclusions

We formulated various relaxations and cutting plane techniques for the problem of maximizing algebraic connectivity. We concluded that the eigenvector cuts are much more effective than the semi-definite cuts from the run times to compute optimal solutions and the number of cuts added. Comparing the binary relaxation and the MISOCP relaxation based on $2 \times 2$ principal minors, we observed that upper bounds obtained from MISOCP relaxation are much tighter and computable in a reasonable time. These upper bounds are further tightened using higher-order principal minors implying $\gamma^* \leq \gamma_4^u \leq \gamma_3^u \leq \gamma_2^u$. In degree-constrained formulation, the quality of the solution increases with the value of $k$ until the optimal solution of ${\mathcal{F}_1}^{'}$ is attained. Also, the run times grows rapidly with the value of $k$ as the feasible set size increases.

Later, the convergence rates of solving the MISDPs with Pajarito.jl (MISDP solver), $n \times n$ eigenvector cuts method, and minor-based relaxation methods are compared. We concluded that for most instances with ten and twelve nodes, $\gamma_2^u \leq \gamma_p \leq \gamma_n$, $\gamma_3^u \leq \gamma_p \leq \gamma_n$. It is implied that using MISOCP formulation and $3 \times 3$ principal minors upper bounding formulation, upper bounds are computed faster compared to solving MISDP with Pajarito solver or $n \times n$ eigenvector cuts method. However, in the case of comparison with $4 \times 4$ principal minors upper bounding formulation, all methods seem to converge at the same rate for most instances in Appendix A.

## REFERENCES


[1] D. Mosk-Aoyama, "Maximum algebraic connectivity augmentation is NP-hard," *Operations Research Letters*, vol. 36, no. 6, pp. 677–679, 2008.

[2] A. Jamakovic and S. Uhlig, "On the relationship between the algebraic connectivity and graph's robustness to node and link failures," in *2007 Next Generation Internet Networks*, pp. 96–102, IEEE, 2007.

[3] H. Nagarajan, S. Rathinam, S. Darbha, and K. Rajagopal, "Synthesizing robust communication networks for uavs," in *American Control Conference (ACC)*, pp. 3730–3735, IEEE, 2012.

[4] H. Nagarajan, S. Rathinam, and S. Darbha, "On maximizing algebraic connectivity of networks for various engineering applications," in *European Control Conference (ECC)*, pp. 1626–1632, IEEE, 2015.

[5] H. Nagarajan, S. Rathinam, S. Darbha, and K. Rajagopal, "Algorithms for synthesizing mechanical systems with maximal natural frequencies," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 5, pp. 2154–2162, 2012.

[6] A. Simonetto, T. Keviczky, and R. Babuška, "Constrained distributed algebraic connectivity maximization in robotic networks," *Automatica*, vol. 49, no. 5, pp. 1348–1357, 2013.

[7] S. Darbha, K. Rajagopal, *et al.*, "Information flow and its relation to the stability of the motion of vehicles in a rigid formation," in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 1853–1858, IEEE, 2005.

[8] R. K. Kincaid, N. Alexandrov, and M. J. Holroyd, "An investigation of synchrony in transport networks," *Complexity*, vol. 14, no. 4, pp. 34–43, 2009.

[9] P. Wei, G. Spiers, and D. Sun, "Algebraic connectivity maximization for air transportation networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 685–

698, 2013.

[10] P. Wei, L. Chen, and D. Sun, “Algebraic connectivity maximization of an air transportation network: The flight routes’ addition/deletion problem,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 61, pp. 13–27, 2014.

[11] L. R. Varshney, “Distributed inference networks with costly wires,” in *Proceedings of the 2010 American Control Conference*, pp. 1053–1058, IEEE, 2010.

[12] H. Nagarajan, *Synthesizing Robust Networks for Engineering Applications With Resource Constraints*. PhD thesis, Department of Mechanical Engineering, Texas A&M University at College Station, 2014.

[13] H. Nagarajan, P. Wei, S. Rathinam, and D. Sun, “Heuristics for synthesizing robust networks with a diameter constraint,” *Mathematical Problems in Engineering*, vol. 2014, 2014.

[14] H. Nagarajan, S. Rathinam, S. Darbha, and K. Rajagopal, “Algorithms for finding diameter-constrained graphs with maximum algebraic connectivity,” in *Dynamics of Information Systems: Mathematical Foundations*, pp. 121–135, Springer, 2012.

[15] C. Maas, “Transportation in graphs and the admittance spectrum,” *Discrete Applied Mathematics*, vol. 16, no. 1, pp. 31–49, 1987.

[16] Y. Wan, S. Roy, X. Wang, A. Saberi, T. Yang, M. Xue, and B. Malek, “On the structure of graph edge designs that optimize the algebraic connectivity,” in *2008 47th IEEE Conference on Decision and Control*, pp. 805–810, IEEE, 2008.

[17] A. Sydney, C. Scoglio, and D. Gruenbacher, “Optimizing algebraic connectivity by edge rewiring,” *Applied Mathematics and computation*, vol. 219, no. 10, pp. 5465–5479, 2013.

[18] S. Y. Shafi, M. Arcak, and L. El Ghaoui, “Graph weight allocation to meet laplacian spectral constraints,” *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1872–1877, 2011.

[19] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 99–103, IEEE, 2005.

[20] W. Malik, S. Rajagopal, S. Darbha, and K. Rajagopal, "Maximizing the algebraic connectivity of a graph subject to a constraint on the maximum number of edges," *Advances in Dynamics and Control: Theory, Methods and Applications, Cambridge Scientific Publishers Ltd, Cottenham, UK*, pp. 135–148, 2009.

[21] H. Nagarajan, P. Wei, S. Rathinam, and D. Sun, "Air transportation network robustness optimization under limited legs itinerary constraint," in *the 5th International Conference on Research in Air Transportation (ICRAT 2012), Berkeley, CA, USA*, Citeseer, 2012.

[22] A. Atamtürk and V. Narayanan, "Conic mixed-integer rounding cuts," *Mathematical programming*, vol. 122, no. 1, pp. 1–20, 2010.

[23] A. Atamtürk and V. Narayanan, "Lifting for conic mixed-integer programming," *Mathematical programming*, vol. 126, no. 2, pp. 351–363, 2011.

[24] H. D. Sherali and B. M. Fraticelli, "Enhancing RLT relaxations via a new class of semidefinite cuts," *Journal of Global Optimization*, vol. 22, no. 1-4, pp. 233–261, 2002.

[25] H. Nagarajan, S. Rathinam, and S. Darbha, "Synthesizing robust communication networks for unmanned aerial vehicles with resource constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 6, 2015.

[26] M. Lubin, *Mixed-integer convex optimization: outer approximation algorithms and modeling power.* PhD thesis, Massachusetts Institute of Technology, 2017.

[27] M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma, "Polyhedral approximation in mixed-integer convex optimization," *Mathematical Programming*, vol. 172, no. 1-2, pp. 139–168, 2018.

[28] A. Ghosh and S. Boyd, "Upper bounds on algebraic connectivity via convex optimization," *Linear algebra and its applications*, vol. 418, no. 2-3, pp. 693–707, 2006.

[29] B. Kocuk, S. S. Dey, and X. A. Sun, "Matrix minor reformulation and SOCP-based spatial branch-and-cut method for the AC optimal power flow problem," *Mathematical Programming Computation*, vol. 10, no. 4, pp. 557–596, 2018.

[30] H. Hijazi, C. Coffrin, and P. Van Hentenryck, "Polynomial sdp cuts for optimal power flow," in *2016 Power Systems Computation Conference (PSCC)*, pp. 1–7, IEEE, 2016.

[31] G. Blekherman, S. S. Dey, M. Molinaro, and S. Sun, "Sparse psd approximation of the psd cone," *arXiv preprint arXiv:2002.02988*, 2020.

[32] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[33] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[34] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.

[35] A. Mosek, "The mosek optimization software," *Online at http://www. mosek. com*, vol. 54, no. 2-1, p. 5, 2010.

[36] G. OPTIMIZATION, "Inc. gurobi optimizer reference manual, 2015," *URL: http://www. gurobi. com*, p. 29, 2014.

[37] C. Coey, M. Lubin, and J. P. Vielma, "Outer approximation with conic certificates for mixed-integer convex problems," *Mathematical Programming Computation*, pp. 1–45, 2020.

[38] T. Magnanti and L. Wolsey, "Optimal trees, handbooks in operational research and management science, vol. 7," 1995.

[39] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, 1981.

[40] V. Vazirani, "Approximation algorithms springer-verlag," *New York*, 2001.

[41] B. Legat, O. Dowson, J. D. Garcia, and M. Lubin, "Mathoptinterface: a data structure for mathematical optimization problems," *arXiv preprint arXiv:2002.03447*, 2020.

[42] J. E. Prussing, "The principal minor test for semi-definite matrices," *Journal of Guidance, Control, and Dynamics*, vol. 9, no. 1, pp. 121–122, 1986.

[43] G. T. Gilbert, "Positive definite matrices and sylvester's criterion," *The American Mathematical Monthly*, vol. 98, no. 1, pp. 44–46, 1991.

[44] H. Nagarajan, E. Yamangil, R. Bent, P. Van Hentenryck, and S. Backhaus, "Optimal resilient transmission grid design," in *2016 Power Systems Computation Conference (PSCC)*, pp. 1–7, IEEE, 2016.

APPENDIX A

All the computational results in this thesis are based on the weighted adjacency matrices shown below.

## A.1 n = 8

**Random weighted adjacency matrices for *eight* nodes problem**

$$w_1 = \begin{pmatrix} 0.0 & 4.561 & 19.02 & 37.537 & 82.393 & 18.295 & 50.073 & 5.511 \\ 4.561 & 0.0 & 50.358 & 2.819 & 5.916 & 34.933 & 43.855 & 44.377 \\ 19.02 & 50.358 & 0.0 & 16.268 & 11.806 & 2.159 & 45.568 & 77.271 \\ 37.537 & 2.819 & 16.268 & 0.0 & 28.642 & 45.083 & 62.932 & 24.352 \\ 82.393 & 5.916 & 11.806 & 28.642 & 0.0 & 2.59 & 23.84 & 13.704 \\ 18.295 & 34.933 & 2.159 & 45.083 & 2.59 & 0.0 & 4.041 & 35.791 \\ 50.073 & 43.855 & 45.568 & 62.932 & 23.84 & 4.041 & 0.0 & 55.83 \\ 5.511 & 44.377 & 77.271 & 24.352 & 13.704 & 35.791 & 55.83 & 0.0 \end{pmatrix}$$

$$w_2 = \begin{pmatrix} 0.0 & 7.991 & 19.023 & 40.147 & 46.093 & 9.834 & 48.182 & 39.823 \\ 7.991 & 0.0 & 82.412 & 17.293 & 26.714 & 31.59 & 36.865 & 22.808 \\ 19.023 & 82.412 & 0.0 & 34.046 & 22.715 & 18.902 & 50.309 & 14.671 \\ 40.147 & 17.293 & 34.046 & 0.0 & 25.462 & 10.701 & 51.117 & 34.138 \\ 46.093 & 26.714 & 22.715 & 25.462 & 0.0 & 38.596 & 53.231 & 16.664 \\ 9.834 & 31.59 & 18.902 & 10.701 & 38.596 & 0.0 & 13.779 & 58.921 \\ 48.182 & 36.865 & 50.309 & 51.117 & 53.231 & 13.779 & 0.0 & 53.351 \\ 39.823 & 22.808 & 14.671 & 34.138 & 16.664 & 58.921 & 53.351 & 0.0 \end{pmatrix}$$

$$w_3 = \begin{pmatrix} 0.0 & 5.449 & 13.087 & 39.46 & 14.189 & 26.056 & 30.279 & 41.788 \\ 5.449 & 0.0 & 23.49 & 18.772 & 24.992 & 43.876 & 14.074 & 66.58 \\ 13.087 & 23.49 & 0.0 & 13.379 & 44.093 & 11.845 & 45.53 & 65.366 \\ 39.46 & 18.772 & 13.379 & 0.0 & 28.403 & 54.327 & 68.801 & 30.908 \\ 14.189 & 24.992 & 44.093 & 28.403 & 0.0 & 31.147 & 62.558 & 8.237 \\ 26.056 & 43.876 & 11.845 & 54.327 & 31.147 & 0.0 & 21.427 & 78.777 \\ 30.279 & 14.074 & 45.53 & 68.801 & 62.558 & 21.427 & 0.0 & 61.276 \\ 41.788 & 66.58 & 65.366 & 30.908 & 8.237 & 78.777 & 61.276 & 0.0 \end{pmatrix}$$

$$w_4 = \begin{pmatrix} 0.0 & 3.166 & 10.819 & 69.61 & 7.771 & 35.867 & 47.759 & 11.385 \\ 3.166 & 0.0 & 23.452 & 26.608 & 13.743 & 63.817 & 56.875 & 12.734 \\ 10.819 & 23.452 & 0.0 & 16.165 & 30.174 & 46.717 & 41.704 & 66.899 \\ 69.61 & 26.608 & 16.165 & 0.0 & 5.841 & 57.495 & 67.21 & 14.102 \\ 7.771 & 13.743 & 30.174 & 5.841 & 0.0 & 63.502 & 61.732 & 23.618 \\ 35.867 & 63.817 & 46.717 & 57.495 & 63.502 & 0.0 & 11.427 & 38.997 \\ 47.759 & 56.875 & 41.704 & 67.21 & 61.732 & 11.427 & 0.0 & 98.913 \\ 11.385 & 12.734 & 66.899 & 14.102 & 23.618 & 38.997 & 98.913 & 0.0 \end{pmatrix}$$

$$w_5 = \begin{pmatrix} 0.0 & 2.544 & 18.566 & 23.983 & 44.333 & 11.513 & 47.634 & 8.196 \\ 2.544 & 0.0 & 17.548 & 20.902 & 29.848 & 56.828 & 16.094 & 45.784 \\ 18.566 & 17.548 & 0.0 & 20.03 & 21.883 & 21.306 & 19.583 & 13.961 \\ 23.983 & 20.902 & 20.03 & 0.0 & 33.448 & 50.94 & 7.763 & 22.462 \\ 44.333 & 29.848 & 21.883 & 33.448 & 0.0 & 60.604 & 57.279 & 7.599 \\ 11.513 & 56.828 & 21.306 & 50.94 & 60.604 & 0.0 & 19.492 & 7.163 \\ 47.634 & 16.094 & 19.583 & 7.763 & 57.279 & 19.492 & 0.0 & 98.613 \\ 8.196 & 45.784 & 13.961 & 22.462 & 7.599 & 7.163 & 98.613 & 0.0 \end{pmatrix}$$

$$w_6 = \begin{pmatrix} 0.0 & 3.368 & 5.354 & 64.684 & 66.925 & 28.203 & 41.094 & 53.284 \\ 3.368 & 0.0 & 34.119 & 8.39 & 27.285 & 35.904 & 11.076 & 51.05 \\ 5.354 & 34.119 & 0.0 & 33.155 & 33.273 & 28.636 & 34.563 & 59.182 \\ 64.684 & 8.39 & 33.155 & 0.0 & 28.884 & 20.305 & 43.513 & 15.11 \\ 66.925 & 27.285 & 33.273 & 28.884 & 0.0 & 62.458 & 34.925 & 3.265 \\ 28.203 & 35.904 & 28.636 & 20.305 & 62.458 & 0.0 & 4.674 & 27.095 \\ 41.094 & 11.076 & 34.563 & 43.513 & 34.925 & 4.674 & 0.0 & 45.437 \\ 53.284 & 51.05 & 59.182 & 15.11 & 3.265 & 27.095 & 45.437 & 0.0 \end{pmatrix}$$

$$w_7 = \begin{pmatrix} 0.0 & 5.721 & 8.828 & 22.02 & 55.966 & 5.384 & 34.178 & 43.546 \\ 5.721 & 0.0 & 17.823 & 18.462 & 31.074 & 26.09 & 18.068 & 28.879 \\ 8.828 & 17.823 & 0.0 & 23.527 & 25.014 & 48.801 & 40.533 & 53.078 \\ 22.02 & 18.462 & 23.527 & 0.0 & 37.835 & 38.275 & 4.024 & 19.766 \\ 55.966 & 31.074 & 25.014 & 37.835 & 0.0 & 50.395 & 50.884 & 11.786 \\ 5.384 & 26.09 & 48.801 & 38.275 & 50.395 & 0.0 & 12.491 & 35.477 \\ 34.178 & 18.068 & 40.533 & 4.024 & 50.884 & 12.491 & 0.0 & 71.75 \\ 43.546 & 28.879 & 53.078 & 19.766 & 11.786 & 35.477 & 71.75 & 0.0 \end{pmatrix}$$

$$w_8 = \begin{pmatrix} 0.0 & 1.537 & 12.505 & 45.077 & 68.271 & 6.608 & 20.672 & 37.893 \\ 1.537 & 0.0 & 76.166 & 11.996 & 10.903 & 25.45 & 57.973 & 36.482 \\ 12.505 & 76.166 & 0.0 & 37.794 & 22.848 & 20.843 & 15.406 & 39.688 \\ 45.077 & 11.996 & 37.794 & 0.0 & 37.311 & 29.056 & 36.097 & 27.623 \\ 68.271 & 10.903 & 22.848 & 37.311 & 0.0 & 63.989 & 59.293 & 4.22 \\ 6.608 & 25.45 & 20.843 & 29.056 & 63.989 & 0.0 & 12.757 & 33.223 \\ 20.672 & 57.973 & 15.406 & 36.097 & 59.293 & 12.757 & 0.0 & 105.431 \\ 37.893 & 36.482 & 39.688 & 27.623 & 4.22 & 33.223 & 105.431 & 0.0 \end{pmatrix}$$

$$w_9 = \begin{pmatrix} 0.0 & 7.473 & 13.871 & 74.945 & 59.785 & 28.499 & 36.559 & 41.392 \\ 7.473 & 0.0 & 63.104 & 1.118 & 18.255 & 56.46 & 30.67 & 28.415 \\ 13.871 & 63.104 & 0.0 & 21.09 & 12.332 & 26.304 & 31.328 & 38.784 \\ 74.945 & 1.118 & 21.09 & 0.0 & 34.87 & 35.743 & 13.807 & 6.835 \\ 59.785 & 18.255 & 12.332 & 34.87 & 0.0 & 74.24 & 78.291 & 8.182 \\ 28.499 & 56.46 & 26.304 & 35.743 & 74.24 & 0.0 & 13.607 & 60.731 \\ 36.559 & 30.67 & 31.328 & 13.807 & 78.291 & 13.607 & 0.0 & 100.509 \\ 41.392 & 28.415 & 38.784 & 6.835 & 8.182 & 60.731 & 100.509 & 0.0 \end{pmatrix}$$

$$w_{10} = \begin{pmatrix} 0.0 & 4.673 & 11.233 & 47.921 & 20.123 & 5.275 & 11.57 & 41.965 \\ 4.673 & 0.0 & 59.46 & 26.49 & 24.895 & 48.453 & 49.937 & 45.337 \\ 11.233 & 59.46 & 0.0 & 20.843 & 21.083 & 33.312 & 3.12 & 56.785 \\ 47.921 & 26.49 & 20.843 & 0.0 & 23.79 & 14.368 & 57.961 & 26.491 \\ 20.123 & 24.895 & 21.083 & 23.79 & 0.0 & 63.058 & 84.36 & 10.774 \\ 5.275 & 48.453 & 33.312 & 14.368 & 63.058 & 0.0 & 6.137 & 37.142 \\ 11.57 & 49.937 & 3.12 & 57.961 & 84.36 & 6.137 & 0.0 & 82.681 \\ 41.965 & 45.337 & 56.785 & 26.491 & 10.774 & 37.142 & 82.681 & 0.0 \end{pmatrix}$$

## A.2 n = 10

### Random weighted adjacency matrices for *ten* nodes problem

$$w_1 = \begin{pmatrix} 0.0 & 163.76 & 3.503 & 67.876 & 14.394 & 54.438 & 25.474 & 99.876 & 15.913 & 6.022 \\ 163.76 & 0.0 & 47.574 & 67.104 & 28.66 & 51.183 & 57.218 & 9.822 & 59.615 & 27.217 \\ 3.503 & 47.574 & 0.0 & 18.147 & 30.961 & 52.739 & 125.676 & 58.656 & 37.765 & 67.003 \\ 67.876 & 67.104 & 18.147 & 0.0 & 5.52 & 30.418 & 92.04 & 102.249 & 121.226 & 58.646 \\ 14.394 & 28.66 & 30.961 & 5.52 & 0.0 & 106.921 & 136.93 & 104.609 & 54.813 & 113.919 \\ 54.438 & 51.183 & 52.739 & 30.418 & 106.921 & 0.0 & 49.676 & 22.745 & 32.664 & 51.791 \\ 25.474 & 57.218 & 125.676 & 92.04 & 136.93 & 49.676 & 0.0 & 17.25 & 40.612 & 47.413 \\ 99.876 & 9.822 & 58.656 & 102.249 & 104.609 & 22.745 & 17.25 & 0.0 & 23.457 & 71.664 \\ 15.913 & 59.615 & 37.765 & 121.226 & 54.813 & 32.664 & 40.612 & 23.457 & 0.0 & 36.308 \\ 6.022 & 27.217 & 67.003 & 58.646 & 113.919 & 51.791 & 47.413 & 71.664 & 36.308 & 0.0 \end{pmatrix}$$

$$w_2 = \begin{pmatrix} 0.0 & 93.316 & 2.527 & 53.971 & 85.534 & 26.498 & 74.277 & 78.661 & 64.908 & 28.791 \\ 93.316 & 0.0 & 60.327 & 86.297 & 72.952 & 47.083 & 11.959 & 35.63 & 46.547 & 30.998 \\ 2.527 & 60.327 & 0.0 & 12.693 & 22.384 & 73.088 & 49.787 & 75.031 & 107.348 & 44.196 \\ 53.971 & 86.297 & 12.693 & 0.0 & 2.949 & 38.669 & 93.402 & 100.18 & 146.217 & 26.371 \\ 85.534 & 72.952 & 22.384 & 2.949 & 0.0 & 146.542 & 111.069 & 70.153 & 76.185 & 103.969 \\ 26.498 & 47.083 & 73.088 & 38.669 & 146.542 & 0.0 & 46.297 & 33.206 & 12.786 & 54.221 \\ 74.277 & 11.959 & 49.787 & 93.402 & 111.069 & 46.297 & 0.0 & 45.418 & 46.791 & 89.519 \\ 78.661 & 35.63 & 75.031 & 100.18 & 70.153 & 33.206 & 45.418 & 0.0 & 33.588 & 16.903 \\ 64.908 & 46.547 & 107.348 & 146.217 & 76.185 & 12.786 & 46.791 & 33.588 & 0.0 & 35.324 \\ 28.791 & 30.998 & 44.196 & 26.371 & 103.969 & 54.221 & 89.519 & 16.903 & 35.324 & 0.0 \end{pmatrix}$$

$$w_3 = \begin{pmatrix} 0.0 & 107.208 & 3.406 & 87.978 & 64.914 & 20.753 & 62.166 & 53.803 & 27.427 & 25.672 \\ 107.208 & 0.0 & 77.458 & 12.672 & 68.199 & 62.85 & 44.67 & 59.036 & 36.669 & 34.066 \\ 3.406 & 77.458 & 0.0 & 15.809 & 31.875 & 66.292 & 39.837 & 17.27 & 40.375 & 96.172 \\ 87.978 & 12.672 & 15.809 & 0.0 & 3.448 & 88.789 & 92.433 & 143.336 & 112.918 & 93.149 \\ 64.914 & 68.199 & 31.875 & 3.448 & 0.0 & 70.83 & 15.616 & 88.457 & 119.224 & 57.526 \\ 20.753 & 62.85 & 66.292 & 88.789 & 70.83 & 0.0 & 8.483 & 21.698 & 53.277 & 26.15 \\ 62.166 & 44.67 & 39.837 & 92.433 & 15.616 & 8.483 & 0.0 & 55.076 & 53.104 & 65.449 \\ 53.803 & 59.036 & 17.27 & 143.336 & 88.457 & 21.698 & 55.076 & 0.0 & 61.078 & 67.917 \\ 27.427 & 36.669 & 40.375 & 112.918 & 119.224 & 53.277 & 53.104 & 61.078 & 0.0 & 38.274 \\ 25.672 & 34.066 & 96.172 & 93.149 & 57.526 & 26.15 & 65.449 & 67.917 & 38.274 & 0.0 \end{pmatrix}$$

$$w_4 = \begin{pmatrix} 0.0 & 98.015 & 5.041 & 61.941 & 81.069 & 48.515 & 56.169 & 37.872 & 62.173 & 34.978 \\ 98.015 & 0.0 & 39.34 & 45.233 & 74.345 & 55.39 & 6.68 & 12.732 & 8.656 & 27.611 \\ 5.041 & 39.34 & 0.0 & 21.846 & 19.463 & 77.648 & 76.21 & 17.843 & 43.12 & 99.103 \\ 61.941 & 45.233 & 21.846 & 0.0 & 4.436 & 52.059 & 137.542 & 58.659 & 115.875 & 62.556 \\ 81.069 & 74.345 & 19.463 & 4.436 & 0.0 & 50.048 & 114.321 & 112.669 & 89.348 & 65.561 \\ 48.515 & 55.39 & 77.648 & 52.059 & 50.048 & 0.0 & 9.728 & 35.854 & 35.726 & 86.644 \\ 56.169 & 6.68 & 76.21 & 137.542 & 114.321 & 9.728 & 0.0 & 38.119 & 19.893 & 44.227 \\ 37.872 & 12.732 & 17.843 & 58.659 & 112.669 & 35.854 & 38.119 & 0.0 & 59.293 & 36.726 \\ 62.173 & 8.656 & 43.12 & 115.875 & 89.348 & 35.726 & 19.893 & 59.293 & 0.0 & 13.371 \\ 34.978 & 27.611 & 99.103 & 62.556 & 65.561 & 86.644 & 44.227 & 36.726 & 13.371 & 0.0 \end{pmatrix}$$

$$w_5 = \begin{pmatrix} 0.0 & 152.166 & 3.101 & 53.275 & 85.707 & 17.515 & 76.988 & 120.713 & 26.992 & 14.145 \\ 152.166 & 0.0 & 32.231 & 43.645 & 81.611 & 84.352 & 11.515 & 5.379 & 29.947 & 17.032 \\ 3.101 & 32.231 & 0.0 & 24.969 & 11.575 & 78.514 & 70.706 & 65.214 & 36.853 & 82.594 \\ 53.275 & 43.645 & 24.969 & 0.0 & 4.775 & 124.872 & 114.592 & 47.112 & 105.923 & 40.946 \\ 85.707 & 81.611 & 11.575 & 4.775 & 0.0 & 37.74 & 54.109 & 107.016 & 45.716 & 9.647 \\ 17.515 & 84.352 & 78.514 & 124.872 & 37.74 & 0.0 & 50.261 & 14.399 & 6.229 & 52.908 \\ 76.988 & 11.515 & 70.706 & 114.592 & 54.109 & 50.261 & 0.0 & 13.946 & 15.281 & 66.432 \\ 120.713 & 5.379 & 65.214 & 47.112 & 107.016 & 14.399 & 13.946 & 0.0 & 30.171 & 60.843 \\ 26.992 & 29.947 & 36.853 & 105.923 & 45.716 & 6.229 & 15.281 & 30.171 & 0.0 & 52.181 \\ 14.145 & 17.032 & 82.594 & 40.946 & 9.647 & 52.908 & 66.432 & 60.843 & 52.181 & 0.0 \end{pmatrix}$$

$$w_6 = \begin{pmatrix} 0.0 & 9.377 & 4.721 & 56.313 & 47.009 & 23.767 & 41.655 & 71.889 & 19.274 & 34.509 \\ 9.377 & 0.0 & 52.754 & 7.786 & 71.228 & 51.851 & 16.601 & 7.737 & 62.512 & 38.278 \\ 4.721 & 52.754 & 0.0 & 30.106 & 20.132 & 111.838 & 56.533 & 27.151 & 47.957 & 23.537 \\ 56.313 & 7.786 & 30.106 & 0.0 & 2.868 & 45.341 & 47.961 & 140.71 & 69.064 & 24.247 \\ 47.009 & 71.228 & 20.132 & 2.868 & 0.0 & 94.186 & 49.221 & 132.985 & 119.548 & 50.596 \\ 23.767 & 51.851 & 111.838 & 45.341 & 94.186 & 0.0 & 56.992 & 9.717 & 22.386 & 46.981 \\ 41.655 & 16.601 & 56.533 & 47.961 & 49.221 & 56.992 & 0.0 & 20.686 & 62.272 & 41.335 \\ 71.889 & 7.737 & 27.151 & 140.71 & 132.985 & 9.717 & 20.686 & 0.0 & 43.462 & 65.37 \\ 19.274 & 62.512 & 47.957 & 69.064 & 119.548 & 22.386 & 62.272 & 43.462 & 0.0 & 74.307 \\ 34.509 & 38.278 & 23.537 & 24.247 & 50.596 & 46.981 & 41.335 & 65.37 & 74.307 & 0.0 \end{pmatrix}$$

$$w_7 = \begin{pmatrix} 0.0 & 101.136 & 1.215 & 82.567 & 82.826 & 55.768 & 95.407 & 77.741 & 51.205 & 6.494 \\ 101.136 & 0.0 & 80.624 & 77.672 & 22.052 & 77.739 & 24.241 & 33.349 & 29.863 & 49.246 \\ 1.215 & 80.624 & 0.0 & 11.01 & 31.372 & 31.381 & 83.527 & 19.564 & 117.149 & 81.872 \\ 82.567 & 77.672 & 11.01 & 0.0 & 4.273 & 79.156 & 101.863 & 104.498 & 35.981 & 46.455 \\ 82.826 & 22.052 & 31.372 & 4.273 & 0.0 & 113.807 & 93.604 & 89.148 & 57.226 & 32.074 \\ 55.768 & 77.739 & 31.381 & 79.156 & 113.807 & 0.0 & 36.344 & 50.372 & 48.309 & 50.214 \\ 95.407 & 24.241 & 83.527 & 101.863 & 93.604 & 36.344 & 0.0 & 28.374 & 35.749 & 88.373 \\ 77.741 & 33.349 & 19.564 & 104.498 & 89.148 & 50.372 & 28.374 & 0.0 & 31.387 & 63.114 \\ 51.205 & 29.863 & 117.149 & 35.981 & 57.226 & 48.309 & 35.749 & 31.387 & 0.0 & 49.766 \\ 6.494 & 49.246 & 81.872 & 46.455 & 32.074 & 50.214 & 88.373 & 63.114 & 49.766 & 0.0 \end{pmatrix}$$

$$w_8 = \begin{pmatrix} 0.0 & 105.994 & 1.516 & 30.929 & 94.926 & 75.9 & 72.28 & 79.469 & 18.141 & 31.378 \\ 105.994 & 0.0 & 50.337 & 32.106 & 86.205 & 76.337 & 25.389 & 48.963 & 37.923 & 55.441 \\ 1.516 & 50.337 & 0.0 & 17.087 & 22.263 & 77.999 & 82.941 & 26.196 & 115.641 & 81.094 \\ 30.929 & 32.106 & 17.087 & 0.0 & 4.244 & 56.06 & 24.887 & 52.251 & 121.221 & 79.681 \\ 94.926 & 86.205 & 22.263 & 4.244 & 0.0 & 81.117 & 120.527 & 162.315 & 118.702 & 32.954 \\ 75.9 & 76.337 & 77.999 & 56.06 & 81.117 & 0.0 & 65.026 & 33.864 & 43.707 & 34.559 \\ 72.28 & 25.389 & 82.941 & 24.887 & 120.527 & 65.026 & 0.0 & 37.091 & 53.066 & 55.84 \\ 79.469 & 48.963 & 26.196 & 52.251 & 162.315 & 33.864 & 37.091 & 0.0 & 12.097 & 92.458 \\ 18.141 & 37.923 & 115.641 & 121.221 & 118.702 & 43.707 & 53.066 & 12.097 & 0.0 & 37.76 \\ 31.378 & 55.441 & 81.094 & 79.681 & 32.954 & 34.559 & 55.84 & 92.458 & 37.76 & 0.0 \end{pmatrix}$$

$$w_9 = \begin{pmatrix} 0.0 & 79.718 & 2.269 & 30.546 & 63.745 & 56.296 & 70.907 & 75.924 & 41.221 & 38.567 \\ 79.718 & 0.0 & 37.167 & 74.454 & 29.224 & 31.561 & 5.847 & 11.246 & 11.579 & 51.085 \\ 2.269 & 37.167 & 0.0 & 31.31 & 15.315 & 59.282 & 15.098 & 23.671 & 85.555 & 128.246 \\ 30.546 & 74.454 & 31.31 & 0.0 & 4.201 & 106.915 & 43.454 & 114.02 & 102.764 & 78.297 \\ 63.745 & 29.224 & 15.315 & 4.201 & 0.0 & 55.669 & 53.152 & 98.398 & 54.739 & 54.331 \\ 56.296 & 31.561 & 59.282 & 106.915 & 55.669 & 0.0 & 74.77 & 7.117 & 17.501 & 44.721 \\ 70.907 & 5.847 & 15.098 & 43.454 & 53.152 & 74.77 & 0.0 & 22.829 & 48.972 & 82.026 \\ 75.924 & 11.246 & 23.671 & 114.02 & 98.398 & 7.117 & 22.829 & 0.0 & 46.802 & 106.862 \\ 41.221 & 11.579 & 85.555 & 102.764 & 54.739 & 17.501 & 48.972 & 46.802 & 0.0 & 11.785 \\ 38.567 & 51.085 & 128.246 & 78.297 & 54.331 & 44.721 & 82.026 & 106.862 & 11.785 & 0.0 \end{pmatrix}$$

$$w_{10} = \begin{pmatrix} 0.0 & 139.623 & 3.504 & 9.438 & 46.775 & 74.135 & 66.013 & 69.794 & 51.525 & 35.588 \\ 139.623 & 0.0 & 62.188 & 89.264 & 58.413 & 42.108 & 3.835 & 12.505 & 16.795 & 51.974 \\ 3.504 & 62.188 & 0.0 & 23.907 & 46.883 & 76.479 & 60.688 & 44.685 & 91.614 & 66.43 \\ 9.438 & 89.264 & 23.907 & 0.0 & 2.651 & 95.249 & 71.894 & 151.338 & 60.165 & 76.407 \\ 46.775 & 58.413 & 46.883 & 2.651 & 0.0 & 59.667 & 43.035 & 53.699 & 36.473 & 44.557 \\ 74.135 & 42.108 & 76.479 & 95.249 & 59.667 & 0.0 & 33.549 & 33.213 & 15.545 & 38.764 \\ 66.013 & 3.835 & 60.688 & 71.894 & 43.035 & 33.549 & 0.0 & 17.11 & 21.631 & 62.847 \\ 69.794 & 12.505 & 44.685 & 151.338 & 53.699 & 33.213 & 17.11 & 0.0 & 24.707 & 87.962 \\ 51.525 & 16.795 & 91.614 & 60.165 & 36.473 & 15.545 & 21.631 & 24.707 & 0.0 & 12.805 \\ 35.588 & 51.974 & 66.43 & 76.407 & 44.557 & 38.764 & 62.847 & 87.962 & 12.805 & 0.0 \end{pmatrix}$$

## A.3 n = 12

### Random weighted adjacency matrices for *twelve* nodes problem

$$w_1 = \begin{pmatrix}
0.0 & 11.014 & 23.262 & 145.202 & 70.808 & 61.591 & 19.983 & 48.297 & 129.076 & 31.658 & 44.223 & 97.277 \\
11.014 & 0.0 & 187.812 & 25.706 & 47.94 & 161.133 & 93.665 & 94.864 & 59.407 & 140.463 & 74.94 & 55.939 \\
23.262 & 187.812 & 0.0 & 15.062 & 16.787 & 129.905 & 105.352 & 75.077 & 100.375 & 27.792 & 78.505 & 95.289 \\
145.202 & 25.706 & 15.062 & 0.0 & 13.448 & 75.88 & 27.749 & 82.322 & 28.711 & 45.277 & 151.469 & 102.898 \\
70.808 & 47.94 & 16.787 & 13.448 & 0.0 & 39.254 & 85.161 & 73.246 & 83.359 & 79.564 & 44.088 & 87.954 \\
61.591 & 161.133 & 129.905 & 75.88 & 39.254 & 0.0 & 99.26 & 23.808 & 144.449 & 59.048 & 36.143 & 92.374 \\
19.983 & 93.665 & 105.352 & 27.749 & 85.161 & 99.26 & 0.0 & 6.273 & 103.233 & 35.753 & 21.122 & 222.415 \\
48.297 & 94.864 & 75.077 & 82.322 & 73.246 & 23.808 & 6.273 & 0.0 & 67.702 & 101.18 & 3.474 & 33.456 \\
129.076 & 59.407 & 100.375 & 28.711 & 83.359 & 144.449 & 103.233 & 67.702 & 0.0 & 146.493 & 185.943 & 6.817 \\
31.658 & 140.463 & 27.792 & 45.277 & 79.564 & 59.048 & 35.753 & 101.18 & 146.493 & 0.0 & 37.312 & 6.794 \\
44.223 & 74.94 & 78.505 & 151.469 & 44.088 & 36.143 & 21.122 & 3.474 & 185.943 & 37.312 & 0.0 & 209.836 \\
97.277 & 55.939 & 95.289 & 102.898 & 87.954 & 92.374 & 222.415 & 33.456 & 6.817 & 6.794 & 209.836 & 0.0
\end{pmatrix}$$

$$w_2 = \begin{pmatrix}
0.0 & 7.757 & 13.476 & 139.507 & 58.49 & 60.214 & 59.227 & 118.921 & 158.696 & 92.288 & 44.94 & 31.67 \\
7.757 & 0.0 & 135.051 & 23.492 & 27.912 & 141.659 & 19.012 & 62.034 & 92.05 & 114.914 & 110.569 & 20.289 \\
13.476 & 135.051 & 0.0 & 44.339 & 25.855 & 182.658 & 74.865 & 56.346 & 40.625 & 90.83 & 83.891 & 32.133 \\
139.507 & 23.492 & 44.339 & 0.0 & 95.647 & 62.19 & 66.996 & 98.434 & 121.344 & 18.703 & 101.472 & 18.152 \\
58.49 & 27.912 & 25.855 & 95.647 & 0.0 & 91.903 & 43.995 & 49.833 & 27.875 & 87.214 & 169.894 & 44.854 \\
60.214 & 141.659 & 182.658 & 62.19 & 91.903 & 0.0 & 89.487 & 110.441 & 124.077 & 20.919 & 18.309 & 63.258 \\
59.227 & 19.012 & 74.865 & 66.996 & 43.995 & 89.487 & 0.0 & 78.248 & 154.573 & 38.422 & 24.034 & 106.527 \\
118.921 & 62.034 & 56.346 & 98.434 & 49.833 & 110.441 & 78.248 & 0.0 & 42.406 & 87.05 & 111.209 & 31.234 \\
158.696 & 92.05 & 40.625 & 121.344 & 27.875 & 124.077 & 154.573 & 42.406 & 0.0 & 181.389 & 200.691 & 5.373 \\
92.288 & 114.914 & 90.83 & 18.703 & 87.214 & 20.919 & 38.422 & 87.05 & 181.389 & 0.0 & 32.509 & 172.227 \\
44.94 & 110.569 & 83.891 & 101.472 & 169.894 & 18.309 & 24.034 & 111.209 & 200.691 & 32.509 & 0.0 & 146.974 \\
31.67 & 20.289 & 32.133 & 18.152 & 44.854 & 63.258 & 106.527 & 31.234 & 5.373 & 172.227 & 146.974 & 0.0
\end{pmatrix}$$

$$w_3 = \begin{pmatrix}
0.0 & 4.481 & 4.372 & 162.981 & 54.708 & 5.064 & 31.413 & 80.41 & 107.447 & 18.058 & 35.372 & 106.553 \\
4.481 & 0.0 & 138.467 & 30.147 & 50.896 & 146.114 & 132.638 & 79.716 & 82.208 & 30.738 & 111.101 & 95.943 \\
4.372 & 138.467 & 0.0 & 44.249 & 21.745 & 78.477 & 77.729 & 40.408 & 94.024 & 49.356 & 54.848 & 117.825 \\
162.981 & 30.147 & 44.249 & 0.0 & 76.88 & 61.893 & 81.709 & 59.782 & 54.572 & 44.711 & 113.184 & 27.38 \\
54.708 & 50.896 & 21.745 & 76.88 & 0.0 & 69.071 & 101.382 & 95.005 & 42.727 & 61.051 & 58.489 & 21.734 \\
5.064 & 146.114 & 78.477 & 61.893 & 69.071 & 0.0 & 37.564 & 72.55 & 98.545 & 77.372 & 38.281 & 141.395 \\
31.413 & 132.638 & 77.729 & 81.709 & 101.382 & 37.564 & 0.0 & 127.326 & 112.752 & 67.921 & 19.475 & 141.649 \\
80.41 & 79.716 & 40.408 & 59.782 & 95.005 & 72.55 & 127.326 & 0.0 & 82.461 & 86.795 & 63.494 & 17.32 \\
107.447 & 82.208 & 94.024 & 54.572 & 42.727 & 98.545 & 112.752 & 82.461 & 0.0 & 106.355 & 35.415 & 15.101 \\
18.058 & 30.738 & 49.356 & 44.711 & 61.051 & 77.372 & 67.921 & 86.795 & 106.355 & 0.0 & 18.232 & 148.464 \\
35.372 & 111.101 & 54.848 & 113.184 & 58.489 & 38.281 & 19.475 & 63.494 & 35.415 & 18.232 & 0.0 & 169.657 \\
106.553 & 95.943 & 117.825 & 27.38 & 21.734 & 141.395 & 141.649 & 17.32 & 15.101 & 148.464 & 169.657 & 0.0
\end{pmatrix}$$

$$w_4 = \begin{pmatrix}
0.0 & 13.713 & 8.995 & 28.892 & 19.183 & 59.007 & 13.38 & 10.721 & 126.631 & 48.432 & 118.943 & 62.629 \\
13.713 & 0.0 & 186.919 & 48.72 & 24.239 & 79.129 & 103.805 & 81.252 & 42.634 & 100.101 & 85.384 & 125.215 \\
8.995 & 186.919 & 0.0 & 42.576 & 59.559 & 42.376 & 127.416 & 18.409 & 79.541 & 107.651 & 87.948 & 162.709 \\
28.892 & 48.72 & 42.576 & 0.0 & 88.796 & 13.119 & 60.913 & 95.772 & 52.32 & 82.426 & 53.571 & 26.191 \\
19.183 & 24.239 & 59.559 & 88.796 & 0.0 & 34.812 & 55.85 & 95.983 & 68.219 & 62.333 & 116.199 & 76.146 \\
59.007 & 79.129 & 42.376 & 13.119 & 34.812 & 0.0 & 49.76 & 67.905 & 144.614 & 80.604 & 78.979 & 136.573 \\
13.38 & 103.805 & 127.416 & 60.913 & 55.85 & 49.76 & 0.0 & 104.984 & 99.603 & 40.116 & 53.716 & 75.441 \\
10.721 & 81.252 & 18.409 & 95.772 & 95.983 & 67.905 & 104.984 & 0.0 & 61.509 & 53.511 & 130.665 & 53.468 \\
126.631 & 42.634 & 79.541 & 52.32 & 68.219 & 144.614 & 99.603 & 61.509 & 0.0 & 16.798 & 121.392 & 25.248 \\
48.432 & 100.101 & 107.651 & 82.426 & 62.333 & 80.604 & 40.116 & 53.511 & 16.798 & 0.0 & 13.987 & 41.66 \\
118.943 & 85.384 & 87.948 & 53.571 & 116.199 & 78.979 & 53.716 & 130.665 & 121.392 & 13.987 & 0.0 & 177.042 \\
62.629 & 125.215 & 162.709 & 26.191 & 76.146 & 136.573 & 75.441 & 53.468 & 25.248 & 41.66 & 177.042 & 0.0
\end{pmatrix}$$

$$w_5 = \begin{pmatrix}
0.0 & 6.865 & 18.26 & 62.434 & 108.091 & 5.148 & 41.198 & 171.447 & 101.612 & 55.501 & 50.001 & 45.888 \\
6.865 & 0.0 & 80.638 & 12.45 & 36.249 & 102.127 & 134.207 & 80.616 & 82.336 & 60.481 & 21.063 & 56.081 \\
18.26 & 80.638 & 0.0 & 49.748 & 48.813 & 51.232 & 117.464 & 58.362 & 111.02 & 26.483 & 54.251 & 161.775 \\
62.434 & 12.45 & 49.748 & 0.0 & 92.756 & 31.956 & 57.416 & 83.623 & 75.458 & 72.153 & 134.216 & 50.353 \\
108.091 & 36.249 & 48.813 & 92.756 & 0.0 & 101.646 & 53.45 & 51.339 & 98.899 & 136.791 & 21.391 & 61.091 \\
5.148 & 102.127 & 51.232 & 31.956 & 101.646 & 0.0 & 43.411 & 18.226 & 60.807 & 77.151 & 77.574 & 153.168 \\
41.198 & 134.207 & 117.464 & 57.416 & 53.45 & 43.411 & 0.0 & 49.458 & 109.347 & 22.304 & 28.038 & 157.194 \\
171.447 & 80.616 & 58.362 & 83.623 & 51.339 & 18.226 & 49.458 & 0.0 & 25.502 & 103.166 & 57.223 & 6.213 \\
101.612 & 82.336 & 111.02 & 75.458 & 98.899 & 60.807 & 109.347 & 25.502 & 0.0 & 145.419 & 110.529 & 8.967 \\
55.501 & 60.481 & 26.483 & 72.153 & 136.791 & 77.151 & 22.304 & 103.166 & 145.419 & 0.0 & 23.963 & 203.399 \\
50.001 & 21.063 & 54.251 & 134.216 & 21.391 & 77.574 & 28.038 & 57.223 & 110.529 & 23.963 & 0.0 & 126.619 \\
45.888 & 56.081 & 161.775 & 50.353 & 61.091 & 153.168 & 157.194 & 6.213 & 8.967 & 203.399 & 126.619 & 0.0
\end{pmatrix}$$

$$w_6 = \begin{pmatrix}
0.0 & 13.502 & 14.214 & 132.37 & 91.14 & 18.601 & 5.116 & 80.91 & 116.529 & 22.986 & 108.889 & 90.092 \\
13.502 & 0.0 & 129.206 & 13.278 & 39.849 & 177.142 & 115.114 & 59.651 & 15.296 & 94.739 & 78.184 & 128.966 \\
14.214 & 129.206 & 0.0 & 33.473 & 56.445 & 91.037 & 140.609 & 91.965 & 92.085 & 62.828 & 96.342 & 53.737 \\
132.37 & 13.278 & 33.473 & 0.0 & 127.372 & 42.441 & 80.69 & 32.326 & 64.564 & 96.31 & 57.596 & 53.745 \\
91.14 & 39.849 & 56.445 & 127.372 & 0.0 & 64.492 & 100.325 & 91.058 & 35.344 & 110.853 & 169.814 & 24.615 \\
18.601 & 177.142 & 91.037 & 42.441 & 64.492 & 0.0 & 97.66 & 116.064 & 71.065 & 31.231 & 28.443 & 180.026 \\
5.116 & 115.114 & 140.609 & 80.69 & 100.325 & 97.66 & 0.0 & 112.396 & 161.069 & 44.557 & 39.961 & 72.08 \\
80.91 & 59.651 & 91.965 & 32.326 & 91.058 & 116.064 & 112.396 & 0.0 & 27.72 & 83.315 & 134.004 & 3.957 \\
116.529 & 15.296 & 92.085 & 64.564 & 35.344 & 71.065 & 161.069 & 27.72 & 0.0 & 186.543 & 213.781 & 13.974 \\
22.986 & 94.739 & 62.828 & 96.31 & 110.853 & 31.231 & 44.557 & 83.315 & 186.543 & 0.0 & 37.65 & 139.465 \\
108.889 & 78.184 & 96.342 & 57.596 & 169.814 & 28.443 & 39.961 & 134.004 & 213.781 & 37.65 & 0.0 & 184.076 \\
90.092 & 128.966 & 53.737 & 53.745 & 24.615 & 180.026 & 72.08 & 3.957 & 13.974 & 139.465 & 184.076 & 0.0
\end{pmatrix}$$

$$
w_7 = \begin{pmatrix}
0.0 & 3.299 & 25.76 & 160.737 & 110.404 & 46.461 & 20.633 & 21.078 & 38.924 & 86.499 & 80.313 & 53.444 \\
3.299 & 0.0 & 35.084 & 45.83 & 37.035 & 23.11 & 70.675 & 30.598 & 63.217 & 92.614 & 113.786 & 25.595 \\
25.76 & 35.084 & 0.0 & 46.733 & 16.427 & 45.288 & 58.248 & 62.133 & 70.81 & 79.087 & 53.695 & 149.741 \\
160.737 & 45.83 & 46.733 & 0.0 & 77.053 & 36.741 & 93.938 & 113.928 & 46.053 & 103.057 & 66.734 & 25.234 \\
110.404 & 37.035 & 16.427 & 77.053 & 0.0 & 94.503 & 37.72 & 103.935 & 18.226 & 135.773 & 155.182 & 76.221 \\
46.461 & 23.11 & 45.288 & 36.741 & 94.503 & 0.0 & 95.173 & 90.228 & 94.285 & 22.359 & 18.133 & 158.672 \\
20.633 & 70.675 & 58.248 & 93.938 & 37.72 & 95.173 & 0.0 & 66.668 & 35.183 & 71.215 & 63.114 & 82.636 \\
21.078 & 30.598 & 62.133 & 113.928 & 103.935 & 90.228 & 66.668 & 0.0 & 40.784 & 85.641 & 202.177 & 34.037 \\
38.924 & 63.217 & 70.81 & 46.053 & 18.226 & 94.285 & 35.183 & 40.784 & 0.0 & 109.866 & 206.458 & 25.421 \\
86.499 & 92.614 & 79.087 & 103.057 & 135.773 & 22.359 & 71.215 & 85.641 & 109.866 & 0.0 & 24.554 & 80.298 \\
80.313 & 113.786 & 53.695 & 66.734 & 155.182 & 18.133 & 63.114 & 202.177 & 206.458 & 24.554 & 0.0 & 154.591 \\
53.444 & 25.595 & 149.741 & 25.234 & 76.221 & 158.672 & 82.636 & 34.037 & 25.421 & 80.298 & 154.591 & 0.0
\end{pmatrix}
$$

$$
w_8 = \begin{pmatrix}
0.0 & 10.562 & 22.649 & 183.866 & 161.819 & 9.992 & 41.689 & 21.38 & 172.037 & 116.774 & 53.119 & 120.174 \\
10.562 & 0.0 & 212.503 & 8.306 & 58.447 & 83.755 & 104.872 & 84.318 & 87.424 & 96.942 & 93.75 & 43.126 \\
22.649 & 212.503 & 0.0 & 27.286 & 30.641 & 99.566 & 83.049 & 95.784 & 84.182 & 118.212 & 15.973 & 137.407 \\
183.866 & 8.306 & 27.286 & 0.0 & 79.298 & 28.227 & 103.382 & 124.775 & 43.534 & 35.015 & 118.911 & 83.198 \\
161.819 & 58.447 & 30.641 & 79.298 & 0.0 & 65.963 & 97.425 & 116.612 & 117.953 & 27.496 & 97.735 & 5.81 \\
9.992 & 83.755 & 99.566 & 28.227 & 65.963 & 0.0 & 70.395 & 92.062 & 118.486 & 85.423 & 33.203 & 121.705 \\
41.689 & 104.872 & 83.049 & 103.382 & 97.425 & 70.395 & 0.0 & 64.636 & 130.906 & 24.38 & 58.238 & 162.3 \\
21.38 & 84.318 & 95.784 & 124.775 & 116.612 & 92.062 & 64.636 & 0.0 & 17.045 & 139.618 & 140.14 & 28.612 \\
172.037 & 87.424 & 84.182 & 43.534 & 117.953 & 118.486 & 130.906 & 17.045 & 0.0 & 34.171 & 184.492 & 26.158 \\
116.774 & 96.942 & 118.212 & 35.015 & 27.496 & 85.423 & 24.38 & 139.618 & 34.171 & 0.0 & 33.691 & 68.506 \\
53.119 & 93.75 & 15.973 & 118.911 & 97.735 & 33.203 & 58.238 & 140.14 & 184.492 & 33.691 & 0.0 & 217.171 \\
120.174 & 43.126 & 137.407 & 83.198 & 5.81 & 121.705 & 162.3 & 28.612 & 26.158 & 68.506 & 217.171 & 0.0
\end{pmatrix}
$$

$$
w_9 = \begin{pmatrix}
0.0 & 12.508 & 10.665 & 114.215 & 97.657 & 18.428 & 62.385 & 118.298 & 73.626 & 41.375 & 49.357 & 136.139 \\
12.508 & 0.0 & 11.955 & 8.548 & 20.377 & 120.44 & 84.044 & 52.538 & 4.652 & 123.59 & 104.817 & 87.186 \\
10.665 & 11.955 & 0.0 & 29.02 & 13.832 & 8.144 & 117.998 & 77.834 & 110.498 & 106.686 & 25.649 & 76.944 \\
114.215 & 8.548 & 29.02 & 0.0 & 21.115 & 39.736 & 45.43 & 113.265 & 128.596 & 156.64 & 111.413 & 70.067 \\
97.657 & 20.377 & 13.832 & 21.115 & 0.0 & 84.935 & 81.583 & 127.149 & 76.491 & 87.578 & 81.489 & 56.081 \\
18.428 & 120.44 & 8.144 & 39.736 & 84.935 & 0.0 & 72.929 & 48.246 & 43.856 & 69.469 & 73.56 & 35.458 \\
62.385 & 84.044 & 117.998 & 45.43 & 81.583 & 72.929 & 0.0 & 92.343 & 53.498 & 20.555 & 53.548 & 161.536 \\
118.298 & 52.538 & 77.834 & 113.265 & 127.149 & 48.246 & 92.343 & 0.0 & 49.963 & 23.195 & 165.392 & 5.172 \\
73.626 & 4.652 & 110.498 & 128.596 & 76.491 & 43.856 & 53.498 & 49.963 & 0.0 & 102.598 & 220.403 & 33.515 \\
41.375 & 123.59 & 106.686 & 156.64 & 87.578 & 69.469 & 20.555 & 23.195 & 102.598 & 0.0 & 18.014 & 170.01 \\
49.357 & 104.817 & 25.649 & 111.413 & 81.489 & 73.56 & 53.548 & 165.392 & 220.403 & 18.014 & 0.0 & 131.855 \\
136.139 & 87.186 & 76.944 & 70.067 & 56.081 & 35.458 & 161.536 & 5.172 & 33.515 & 170.01 & 131.855 & 0.0
\end{pmatrix}
$$

$$
w_{10} = \begin{pmatrix}
0.0 & 9.663 & 19.529 & 49.797 & 152.406 & 5.725 & 38.503 & 90.324 & 107.378 & 87.201 & 79.937 & 41.879 \\
9.663 & 0.0 & 8.486 & 9.238 & 22.479 & 82.918 & 13.682 & 28.279 & 59.756 & 65.419 & 52.074 & 133.047 \\
19.529 & 8.486 & 0.0 & 34.512 & 33.886 & 119.437 & 95.187 & 84.328 & 48.61 & 36.861 & 96.694 & 61.215 \\
49.797 & 9.238 & 34.512 & 0.0 & 48.766 & 57.715 & 48.346 & 65.077 & 118.319 & 111.97 & 39.746 & 70.638 \\
152.406 & 22.479 & 33.886 & 48.766 & 0.0 & 59.061 & 120.758 & 46.87 & 40.937 & 86.183 & 170.071 & 51.353 \\
5.725 & 82.918 & 119.437 & 57.715 & 59.061 & 0.0 & 72.74 & 63.276 & 59.343 & 78.708 & 26.132 & 188.502 \\
38.503 & 13.682 & 95.187 & 48.346 & 120.758 & 72.74 & 0.0 & 14.086 & 146.877 & 63.459 & 20.118 & 52.806 \\
90.324 & 28.279 & 84.328 & 65.077 & 46.87 & 63.276 & 14.086 & 0.0 & 37.715 & 128.688 & 160.005 & 9.194 \\
107.378 & 59.756 & 48.61 & 118.319 & 40.937 & 59.343 & 146.877 & 37.715 & 0.0 & 167.468 & 135.021 & 21.325 \\
87.201 & 65.419 & 36.861 & 111.97 & 86.183 & 78.708 & 63.459 & 128.688 & 167.468 & 0.0 & 15.279 & 150.21 \\
79.937 & 52.074 & 96.694 & 39.746 & 170.071 & 26.132 & 20.118 & 160.005 & 135.021 & 15.279 & 0.0 & 157.726 \\
41.879 & 133.047 & 61.215 & 70.638 & 51.353 & 188.502 & 52.806 & 9.194 & 21.325 & 150.21 & 157.726 & 0.0
\end{pmatrix}
$$