

IONIAN UNIVERSITY

**Faculty of Information Science
and Informatics**

Department of Informatics



Master Thesis

Research Directions in Informatics

**Study of SDN Controller Locations in Network
Environments**

Asterios Papamichail – ΠΜΕ201910

10 September 2021

Supervisor

Michael Stefanidakis, *Associate Professor,*
Ionian University

Advisory Committee Members

Michael Stefanidakis, *Associate Professor,*
Ionian University

Konstantinos Oikonomou, *Professor,*
Ionian University

Theodore Andronikos, *Associate Professor,*
Ionian University

To those that pursue their dreams.

To the people that push their students to overcome their self.

Abstract

The emergence of the public and private cloud services resulted in on demand access to infrastructure, applications and other resources. For instance the the big data applications requires parallel processing of massive data on thousands of servers. These servers must be able to communicate with each other, to produce the final result. This communication requires a huge amount of network bandwidth. This amount of data raised many challenges to the existing TCP/IP network architecture. Software Defined Networking (SDN) has been proposed as a new network architecture to tackle these challenges.

The SDNs are capable of managing the network resources in a centralized manner. They allow dynamic and efficient network configuration to increase the network's performance. This increase in performance is accomplished by disassociating the forwarding process from the routing process. The network's function is decoupled into two distinct planes: the control plane and the data plane. The control plane is in charge of managing the network's resources efficiently, while the data plane is responsible for forwarding the packets from source to destination.

A common use case of SDNs is to facilitate backbone networks. Backbone networks consist of large number of devices in the network and relatively long distances between nodes. The long distances of between nodes raises a constraint that the controller of a SDN could not be placed in a bad location, i.e. the controller has minimum distances between a handful of switches and long distances between the remaining devices. Instead, the controller must be placed at a location that its distance from the devices of the network is minimum. This problem is the 1-median problem.

There are numerous ways to tackle this problem the most common is to formulate the problem in an integer linear problem and then solve it using heuristics. This study presents an approach that incorporates elements of algebraic graph theory.

The proposed approach uses the spectral characteristics of the distance matrix, in which rows and columns are indexed from the nodes of the graph and is populated with the shortest distance between each pair of nodes. More specifically, the proposed method uses the eigenvector of the distance matrix that corresponds to its largest eigenvalue, this eigenvector is also named principal eigenvector of the distance matrix. The selection of median is done by finding the element of the principal eigenvector that is closer to zero.

The evaluation of the proposed method is performed over various cases and number of nodes in random tree graphs. Its aim is to select the optimal solution of the median problem applied to SDNs. A program in Python programming language is developed in order to facilitate this evaluation process. This program is responsible to create the random tree graphs, produce the distance matrix and then to compare the acquired solution to the actual median.

The proposed method is firstly evaluated on random tree graphs with each edge weight equal to one. The proposed method is not able to find the optimal solution in some cases of these graphs. The proposed method is also evaluated on random tree graphs that its edges have random weights. The results show that the proposed method can find the median of a graph in most cases even in large-scale topologies.

Περίληψη

Η εμφάνιση των δημόσιων και ιδιωτικών υπηρεσιών ζλουδ είχε ως αποτέλεσμα την κατ' άπαιτηση πρόσβαση σε υποδομές, εφαρμογές και άλλους πόρους. Για παράδειγμα, οι εφαρμογές μεγάλων δεδομένων απαιτούν παράλληλη επεξεργασία μαζικών δεδομένων σε χιλιάδες διακομιστές. Αυτοί οι διακομιστές πρέπει να είναι σε θέση να επικοινωνούν μεταξύ τους, προκειμένου να παράξουν το τελικό αποτέλεσμα. Αυτή η επικοινωνία απαιτεί τεράστιο βανδωιδτη δικτύου. Αυτός ο όγκος δεδομένων έθεσε πολλές προκλήσεις στην υπάρχουσα αρχιτεκτονική δικτύου TCP/IP. Το Software Defined Networking (SDN) έχει προταθεί ως νέα αρχιτεκτονική δικτύου για να λύσει αυτού του είδους τις προκλήσεις.

Τα SDN είναι ικανά να διαχειρίζονται τους πόρους του δικτύου με κεντρικό τρόπο. Επιτρέπουν τη δυναμική και αποτελεσματική διαμόρφωση δικτύου για να αυξήσουν την απόδοσή του. Αυτή η αύξηση της απόδοσης επιτυγχάνεται με τον διαχωρισμό της διαδικασίας προώθησης δεδομένων από τη διαδικασία δρομολόγησης των πακέτων. Η λειτουργία του δικτύου αποδομήται σε δύο διαφορετικά επίπεδα: το επίπεδο ελέγχου και το επίπεδο δεδομένων. Το επίπεδο ελέγχου είναι υπεύθυνο για τη αποτελεσματική διαχείριση των πόρων του δικτύου, ενώ το επίπεδο δεδομένων είναι υπεύθυνο για την προώθηση των πακέτων από την πηγή στον προορισμό.

Μια συνηθισμένη περίπτωση χρήσης των SDN είναι η διαχείριση των backbone δικτύων. Αυτού του είδους δίκτυα αποτελούνται από μεγάλο αριθμό συσκευών στο δίκτυο και σχετικά μεγάλες αποστάσεις μεταξύ κόμβων. Οι μεγάλες αποστάσεις μεταξύ των κόμβων δημιουργούν έναν περιορισμό ότι ο controller ενός SDN δεν θα μπορούσε να τοποθετηθεί σε κακή θέση, δηλαδή σε σημείο όπου ο controller έχει ελάχιστες αποστάσεις από μερικούς κόμβους του δικτύου και μεγάλες αποστάσεις μεταξύ των υπόλοιπων κόμβων. Αντ' αυτού, ο controller πρέπει να τοποθετηθεί σε μια θέση που η απόσταση του από τις υπόλοιπες συσκευές του δικτύου είναι ελάχιστη. Αυτό το πρόβλημα λέγεται 1-median.

Υπάρχουν πολλοί τρόποι για την αντιμετώπιση του συγκεκριμένου προβλήματος, η πιο κοινή είναι η διατύπωσή του σε Integer Linear Problem και στη συνέχεια να επιλυθεί με την χρήση ευρετικών μεθόδων. Αυτή η μελέτη παρουσιάζει μια προσέγγιση που ενσωματώνει στοιχεία της αλγεβρικής θεωρίας γράφων. Η προτεινόμενη προσέγγιση χρησιμοποιεί τα φασματικά χαρακτηριστικά του distance matrix, στην οποία οι γραμμές και οι στήλες ευρετηριάζονται από τους κόμβους του γράφου και συμπληρώνεται με τη μικρότερη απόσταση μεταξύ κάθε ζεύγους κόμβων. Πιο συγκεκριμένα, η προτεινόμενη μέθοδος χρησιμοποιεί το ιδιοδιανύσμα του distance matrix που αντιστοιχεί στη μεγαλύτερη ιδιοτιμή του. Η επιλογή του median κόμβου γίνεται με την εύρεση του στοιχείου του κύριου ιδιοδιανύσματος που είναι πιο κοντά στο μηδέν.

Η αξιολόγηση της προτεινόμενης μεθόδου πραγματοποιείται σε τυχαίους γράφους δέντρων και διάφορους αριθμούς κόμβων. Στόχος της είναι να επιλέξει τη βέλτιστη λύση του median προβλήματος εφαρμοσμένο στα SDNs. Ένα πρόγραμμα στη γλώσσα προγραμματισμού Python αναπτύσσεται προκειμένου να διευκολυνθεί τη διαδικασία αξιολόγησης. Αυτό το πρόγραμμα είναι υπεύθυνο για τη δημιουργία των τυχαίων γράφων δέντρων, την παραγωγή του distance matrix και στη συνέχεια τη σύγκριση της λύσης της προτεινόμενης μεθόδου με το πραγματικό median του δικτύου.

Η προτεινόμενη μέθοδος αξιολογείται σε τυχαία γραφήματα δέντρων όπου το βάρος κάθε σύνδεσης είναι ίσο με ένα. Η προτεινόμενη μέθοδος δεν είναι σε θέση να βρει τη βέλτιστη λύση σε ορισμένες περιπτώσεις αυτών των γράφων. Η προτεινόμενη μέθοδος αξιολογείται επίσης σε τυχαία γραφήματα δέντρων που οι συνδέσεις της έχουν τυχαία βάρη. Τα αποτελέσματα δείχνουν ότι η προτεινόμενη μέθοδος μπορεί να βρει τον median του γράφου στις περισσότερες περιπτώσεις δικτύων ακόμη και σε τοπολογίες μεγάλης κλίμακας.

Acknowledgments

First and foremost, I would like to express my gratitude towards my supervisor, Associate Professor Michael Stefanidakis, for providing guidance, without which I would not be able to complete this thesis. I would like to thank Professor Konstantinos Oikonomou and Associate Professor Theodore Andronikos for their valuable comments over the course of completing this thesis. I would like to thank particularly Prof. Oikonomou for providing opportunities and experiences that allowed me to grow.

I am also thankful to have my good friends and colleagues: Konstantinos, Alexandros, Korina, Andreana, Spiros and Giannis, with whom I spent countless hours together. Thank you all for letting me contribute to our wonderful group, without their company this journey would not be as enjoyable as it is.

Last but not least, I would like to sincerely thank my parents, Aggelos and Filio, and my brother Giannis, for always believing and supporting me in any way. They were always there for me in times of hardship and I will be forever grateful for that. I would not be able to get here, without their care and nonstop efforts. I can only hope to make them proud and reciprocate in the future.

Contents

A	Introduction	1
A.1	The 5G concerns	2
A.2	Software Defined Networks	2
A.3	Facility Location and Algebraic Graph Theory	4
A.4	Past Related work	5
A.5	Contribution	6
A.6	Thesis Structure	7
B	Facility Location and Algebraic Graph Theory	9
B.1	Facility Location	9
B.1.1	Capacitated Facility Location	10
B.1.2	Uncapacitated Facility Location	11
B.2	Algebraic Graph Theory	11
B.2.1	Graph Theory	11
B.2.2	Matrix Algebra	12
B.2.3	Adjacency Matrix of a Graph	16
C	Distance Matrix and the Median Problem	19
C.1	Median Problem	19
C.2	Distance Matrix	21
C.2.1	Spectral Properties	21
C.3	Tree Topologies	21
C.3.1	Cases of a tree topology	22

<i>Contents</i>	vii
D Proposed Method and Simulation Results	25
D.1 The Proposed Method	26
D.2 Experimentation Environment	26
D.3 Main limitations of the method	26
D.3.1 The case of equal costs	27
D.3.2 The case of different costs	28
D.3.3 Similar topologies	29
D.4 Experimental results	30
E Summary and Conclusions	33
I Glossary	43
II Notations	45
III Abbreviations	46
Index	47

List of Figures

A.1	An example of a SDN network. The solid lines present the connections between the switches (data plane) and the dashed line presents the connection to the controller (control plane).	3
B.1	Examples of a directed graph and an undirected graph.	13
C.1	A weighted graph of 8 nodes.	20
C.2	Examples of tree topologies. A light red node presents the median of the graph. (a) A path topology of odd number of nodes. (b) A path topology of even number of nodes. (c) A simple case of a tree topology.	23
D.1	The nodes with the light blue color represent a path that its length is equal to the diameter of the network. The light red node represents the median found by the proposed method. The green node corresponds to the actual median.	28
D.2	The nodes with the light blue color represent a path that its length is equal to the diameter of the network. The light red node represents the median found by the proposed method. The green node corresponds to the actual median.	30
D.3	The mean number of hops between the node selected using the proposed method and the median, evaluated on six different sizes of random trees. For each size of network, 1000 different cases are evaluated.	32

List of Tables

<i>C.1</i>	<i>Characteristics of the graph presented in Figure C.1.</i>	20
<i>D.1</i>	<i>The spectral characteristics of the graph presented in Figure D.1 . .</i>	29
<i>D.2</i>	<i>The spectral characteristics of the graph presented in Figure</i>	31

Chapter A

Introduction

COMPUTER networks are built from a large number of devices like routers, switches and a wide variety of middleboxes. Network operators are responsible to configure these devices in order to respond to the network's events as well as applications' standards. Thus, they have to manually install those high level policies to the network devices while converting them to low level policies. Additionally, they have to take under consideration the dynamic nature of complex networks. Usually, they have to install those policies with limited access to tools.

A key technology that contributes to the dynamic nature of the internet is the Internet of Things (IoT). In the IoT realm, many every day objects will be on the network in one form or another. Sensor technologies will rise to meet this new challenge in which information and communication systems are integrated in the every day environment resulting into, the generation of enormous amount of data that have to be stored, processed and presented in a seamless, efficient and easily interpretable form.

In order to overcome this issue the idea of programmable networks has been proposed to aid the network evolution as its infrastructure has become critical to the society. In particular, Software Defined Networking (SDN) [12] is a new paradigm where the forwarding process is decoupled from the routing process (control process). SDN aims to simplify the network management by allowing software developers to rely on network resources in the same manner as they do on storage and computing resources. In this paradigm, the network intelligence is logically centralized in software-based controllers (control plane) and the other devices are responsible only

for packet forwarding (data plane). This can be programmed via an open interface, e.g. ForCES [1], OpenFlow [2], etc.

A.1 The 5G concerns

The evolution of humankind in technological advancements over the years resulted in an exponential increase of devices that have internet access. Also, these devices penetrate the everyday life of families in order to accommodate for everyday jobs, like housekeeping, etc. These devices have access to the internet to enable remote control. This technology is called the Internet of Things.

IoT has many applications [3, 4, 5, 6] in both the corporate and domestic fields. In the corporate field, an IoT application enables the remote control and monitoring of automation, industrial manufacturing, logistics, and transportation of goods. In the domestic field, IoT aids to ease everyday life such as assisted living, smart homes, offices, and a notable contribution of IoT is of course in healthcare applications [7, 8, 9, 10, 11].

IoT combined with the upcoming technology of fifth-generation networks (5G) is used in many applications consisting of large-scale networks. The large-scale networks produce an enormous amount of data on an hourly basis or even every minute. The collected data must somehow be transferred and stored in databases. The transportation of data is a very demanding process on the network resources. To accommodate for the amount of data transferred over the internet, the number of network devices has increased dramatically, resulting in higher demand of manual configuration. To minimize the manual configuration of the networking devices such as routers and switches, the new paradigm of SDN is proposed.

A.2 Software Defined Networks

Software Defined Networks (SDN)s are capable of managing the network in a centralized manner while allowing dynamic and efficient network configuration [13]. Frequent use of SDNs is to facilitate backbone networks [14, 15]. Specifically, SDNs allow dynamic and efficient network configuration in order to increase the network's performance and monitoring. This is accomplished by disassociating the forwarding

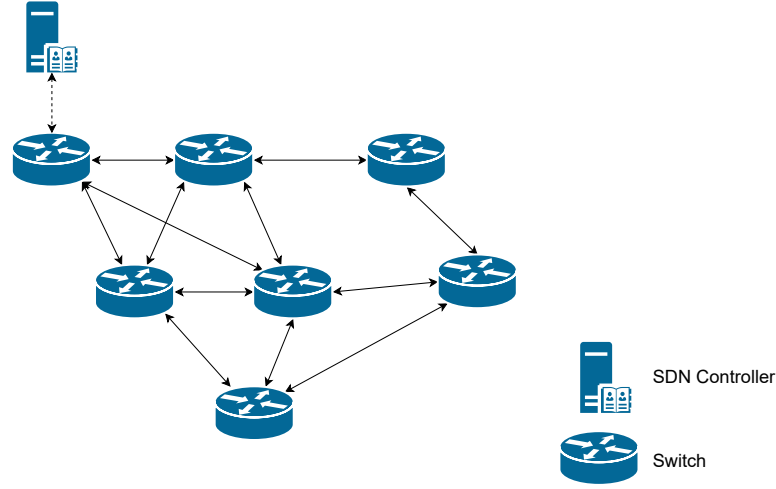


Figure A.1: An example of a SDN network. The solid lines present the connections between the switches (data plane) and the dashed line presents the connection to the controller (control plane).

process from the routing process. Specifically, SDNs decouple the function of the network into two distinct planes, the control plane and the data plane [12, 16]. The control plane is responsible for managing the resources of the network by routing the packets to minimize the utilization of the network traffic. The data plane is in charge of transferring the packets from source to destination. In Figure A.1, an SDN network is presented. It is distinguished the control plane consisting of the controller and the data plane comprised of the switches. The solid lines present the connections between the switches of network and the dashed line presents the connection to the controller.

The involved devices of the control plane are the switches and the controller of the domain. The domain is a group of switches assigned to a controller. When traffic is generated at a host of a particular switch, then this switch creates a **PACKET_IN** message to the controller. Also, **PACKET_IN** messages are created when a switch accepts a message that does not know how to handle it. This accommodates for the case where the packet comes from another domain, i.e. a neighboring switch that is assigned to a different controller. When a controller accepts a **PACKET_IN** message, has to respond to the switch with a **PACKET_OUT** or a **FLOW_MOD** message. This message instructs the switch on how to handle the incoming packet. In order to improve the network's performance, the time taken to receive the response must

be minimized. Specifically, the time for the controller to receive the `PACKET_IN` message and transmit the `packet_out` or the `FLOW_MOD` message to the switch. Thus, the minimization must be performed between the switches and the controller.

To minimize this waiting period, the controller must be placed in such a way that its distance from the switches of the domain must be the lowest. Extensive research has been conducted on this problem, which is similar to the facility location problem. For the facility location problem, numerous studies try to find a suboptimal solution. The facility location problem is an np-hard problem [17] that has troubled many researchers on how and where to place the controller with minimum cost.

A.3 Facility Location and Algebraic Graph Theory

Facility location problem has been studied for over a century [18], but the researchers are interested into it because, there are numerous problems that can be formulated as a facility location problem. Also, they developed new approximation algorithms in order to solve the facility location problem. The facility location problems can be divided into two categories, such as capacitated and uncapacitated facility location problems, which can be solved by formulating the problem into an Integer Linear Program [19, 20]. Facility location problem has many applications in supply chain [21], in healthcare applications [22], IoT [23], etc.

Even though the approximations to the facility location problem are very close to the optimal solution, their time complexity is very high. This means that the approximations of the problems take too long to compute. For this reason, the researchers found different methods to solve the facility location problems, such as simulated annealing [24, 25] and metaheuristic approaches [26, 27].

As a network can be formulated as a graph, someone would be interested in solving the facility location problem using algebraic graph theory [28, 29]. This would be possible if the topology of the network is known. For the particular case of SDNs, the topology of the network is known, as the controller has complete topology information. By finding an algebraic solution to the facility location problem, the researchers could understand in more depth the problem and would find easier as well as, computationally faster methods to solve it.

A.4 Past Related work

There is extensive research interest in the facility location problem, as this problem is studied for over a half century [30]. Tcha and Yoon [31] deal with a stochastic facility location problem, where the demands of customers are known random variables. This paper uses a step function to distribute the customers' demands, to such a degree that the problem is approximated by a linear programming problem and then solved by a heuristic procedure.

Galvão and Raggi [32] present a binary formulation of the uncapacitated facility location problem. To solve the uncapacitated facility location problem they developed a 3-stage solution. The proposed solution consists of hierarchical structure of algorithms stacked on top of each other, activated one at a time, when the previous one is not able to find the optimal solution. At the top is a primal-dual algorithm, then, follows a subgradient optimization to solve a lagrangean dual and finally, a branch-and-bound algorithm. This solution solves simpler problems in its early stages, while the "harder" problems may need the final stages of the algorithm in order to be solved.

Erlenkotter [33] tackles the uncapacitated facility location problem by formulating it in a linear programming dual. He solves it through simple ascent and adjustment methods that produces a monotone improvement of the dual objective. These methods often provide an optimal solution directly, if not, they provide bounds for a branch and bound procedure that completes the search for the optimum. This method is tested in 48 different large scale problems and in 46 of them a linear programming problem gives an optimal integer solution.

Goldengorin et al. [34] propose branch and peg algorithms which are based on the branch and bound ones. The proposed algorithms make use of some rules to reduce the size of Simple Plant Location Problem (SPLP), or uncapacitated facility location problem, instances. These rules are constructed with the help of the pseudo-Boolean representation of the SPLP. Additionally, these rules are not only used from the preprocessing step, as well as a tool to solve or reduce the size of a subproblem. This representation is also used to compute efficient branching functions. The results show that the proposed branch and peg algorithms outperform the branch and bound ones.

Heller et al. [35] are the first researchers that motivated the controller placement problem. They are focused on the number of controllers that are needed for a known topology and where they should be placed. They showed that one controller is sufficient to meet existing time constraints.

Sallahi and St-Hilaire [36] proposed a mathematical model for the controller placement problem in SDNs. This mathematical model takes as an input a set of switches, then determines the optimal number of controllers, their location, and type. The model tries to minimize the cost of the network while considering different constraints. The simulations show that the models work in small-scale networks. Contrariwise, in large-scale networks, the solver runs out of memory and takes too much time to solve the problem.

A.5 Contribution

The identification of alternative approaches can lead to efficient solutions, even if they are not the optimal ones. To this end, an approach incorporating elements of algebraic graph theory is proposed in an attempt to better understand how these unintuitive elements may contribute to finding a solution to the k-median problem. In particular, the 1-median problem is tackled, as it constitutes the stepping stone to the general solution.

In this study, the 1-median problem is tackled by using the algebraic graph theory. The 1-median of a graph is approximated with the spectral characteristics of a relatively unstudied matrix of the algebraic graph theory. More specifically, this study makes use of the distance matrix, in which rows and columns are indexed from the nodes of the graph and are populated with the shortest distance between each pair of nodes. This study proposes the approximation of the 1-median problem by finding the element of the distance matrix's principal eigenvector that is closest to zero.

The proposed method is evaluated in various cases and number of nodes in random tree graphs. For the evaluation process, a program in Python programming language is developed. This program is responsible to create random tree graphs, produce its distance matrix and then compare the solution that the method proposes to the median. The proposed method is evaluated on random tree graphs with each edge weight equal to one and in cases that the weights of the edges are random.

A.6 Thesis Structure

The rest of this study is structured as follows. Chapter B presents the definition of facility location problem and introduces the algebraic graph theory. First, it introduces the facility location problem and its types. Then, it introduces basic concepts of Algebraic Graph Theory. More specifically, it discusses some concepts of graph theory, like directed, undirected and weighted graphs. Furthermore, it presents basic concepts of matrix algebra like matrix multiplication, determinant of a matrix, eigenvalues, and eigenvectors of matrices. Finally, it discusses the adjacency matrix of a graph and its spectral properties.

Chapter C defines the median problem. Then, it introduces the creation of the distance matrix and its spectral properties. Finally, this chapter examines the special topology of trees.

In Chapter D, the experimental results are presented. This chapter presents the proposed method for finding the median of a graph. Then, it discusses the limitations of this method on simple graphs. Next, this chapter details the experimentation environment. Finally, it presents the experimental results of the proposed method.

Finally, Chapter E concludes this study. Also, the Chapter E gives directions for future work.

Chapter B

Elements of Facility Location Problems and Algebraic Graph Theory

THIS chapter presents the facility location problem while presenting the capacitated and uncapacitated types of facility location problems. Also, it introduces some basic concepts of Algebraic Graph Theory. More specifically, it introduces some types of graphs like, directed and undirected. Additionally, it presents the concept of weighted graphs. Furthermore, it refers to the basic concepts of matrix algebra like matrix multiplication, determinant of a matrix, eigenvalues, and eigenvectors of matrices. Finally, it presents the adjacency matrix of a graph as well as some spectral properties of the adjacency matrix.

B.1 Facility Location

The facility location problem is one of the most well studied problem in operations research [37, 38], computer science [39, 40] and in computational geometry [41, 42] as well. This problem examines the methods on how to place the facilities to minimize transportation costs while considering other factors like, locations that are not appropriate to facilitate hazardous materials because they are located near residential areas.

A facility location problem is also named a Weber problem in the branch of

geometry. A facility is to be placed at a location that minimizes the weighted sum of distances from a given set of points/sites. More complex formulations of the problem consider the placement of multiple facilities, constraints on the available locations of the facilities, and more criteria that affect the placement of the facility.

Often facility location problems are solved as integer programs. Integer Programming is a well-known method to formulate real-world problems, including many np-hard problems. Most of the techniques to solve these integer programs are based on linear programming relaxations. These techniques are followed because solving linear programs is much easier computationally than solving integer programs. The integer programs are np-hard, while the linear ones can be solved in polynomial time [43].

Facility location problems are usually presented as follows: suppose there exist n possible facilities and m customers. The goal is to find (1) which of the n facilities to open and (2) which opened facilities to use for the supply of the customers in order to satisfy their demand at minimum cost. Let f_i denote the opening cost of a facility i , for $i = 1, \dots, n$. Let c_{ij} denote the cost to transfer a good from a facility i to a customer j , for $i = 1, \dots, n$ and $j = 1, \dots, m$. Now suppose that the facilities have a maximum amount of goods that can produce in a unit of time. So, each facility i has a maximum production p_i for a unit of time. More precisely, p_i denotes the capacity of the facility i . The remainder of this section follows [44].

B.1.1 Capacitated Facility Location

Let's introduce a binary variable x_i for $i = 1, \dots, n$, where $x_i = 1$ if the facility i is opened, and $x_i = 0$ otherwise. Additionally, the introduction of a new variable is essential. The fraction of the demand d_j filled by a facility i is represented by the variable y_{ij} for $i = 1, \dots, n$ and $j = 1, \dots, m$. The capacitated facility location problem is given by

$$\min \sum_{i=1}^n \sum_{j=1}^m c_{ij} d_j y_{ij} + \sum_{i=1}^n f_i x_i$$

such that

$$\begin{aligned} \sum_{i=1}^n y_{ij} &= 1 && \text{for all } j = 1, \dots, m \\ \sum_{j=1}^m d_j y_{ij} &\leq p_i x_i && \text{for all } i = 1, \dots, n \\ y_{ij} &\geq 0 && \text{for all } i = 1, \dots, n \text{ and } j = 1, \dots, m \\ x_i &\in \{0, 1\} && \text{for all } i = 1, \dots, n \end{aligned}$$

Note that the constraints ensure that when the facility i is not opened ($x_i = 0$), no demand of any customer can be filled to the facility i .

B.1.2 Uncapacitated Facility Location

From the capacitated facility location problem someone could set the production (capacity) of a facility into $p_i = +\infty$ for all $i = 1, \dots, n$. In this case, it is always optimal to satisfy the whole demand d_j of the customer j to the nearest opened facility i . For this reason, we may replace the continuous variable y_{ij} from the capacitated facility location with a binary variable z_{ij} , where z_{ij} denotes if a customer j is supplied by the facility i . From above the uncapacitated facility location problem is given by

$$\min \sum_{i=1}^n \sum_{j=1}^m c_{ij} d_j z_{ij} + \sum_{i=1}^n f_i x_i$$

such that

$$\begin{aligned} \sum_{i=1}^n z_{ij} &= 1 && \text{for all } j = 1, \dots, m \\ \sum_{j=1}^m z_{ij} &\leq M x_i && \text{for all } i = 1, \dots, n \\ z_{ij} &\in \{0, 1\} && \text{for all } i = 1, \dots, n \text{ and } j = 1, \dots, m \\ x_i &\in \{0, 1\} && \text{for all } i = 1, \dots, n \end{aligned}$$

where M is a constant chosen to be suitably large.

B.2 Algebraic Graph Theory

B.2.1 Graph Theory

A network can be defined as a group of devices that are interconnected to communicate with each other [45]. A graph $G = (V, E)$ is formed by a set V of vertices and a set E of edges or links that connect pairs of vertices. The cardinality of V is denoted by n and the cardinality of E by m . If two vertices are connected, there is an edge that links them, then these vertices are *adjacent* and they called *neighbors*. An edge that connects two vertices $u, v \in V$ is denoted by $\{u \ v\}$. The degree of a vertex $v \in V$ of a graph $G = (V, E)$ is the number of edges connected to it.

Graphs can be distinguished in *undirected* and *directed*. In undirected graphs, the connections of vertices do not restrict the route between the vertices. An undirected edge that connects two vertices $u, v \in V$ is denoted by $\{u \ v\}$ or $\{v \ u\}$. Yet, in

the directed graphs, an edge has a direction that restricts the route from a node to another node. A directed edge that connects two vertices $u, v \in V$ is denoted by $(u \ v)$ which is not the same as $(v \ u)$. An undirected graph can be defined as a directed one, where each edge $\{u \ v\}$ of the undirected graph is replaced by two opposite directed edges, like $(u \ v)$ and $(v \ u)$. Figure B.1 presents an example of a directed and an undirected graph.

Often it is useful to associate numerical values (weights) with the edges of a graph, these graphs are called weighted graphs. These weights are represented by a function $w : E \rightarrow \mathbb{R}$ that assigns a weight to each edge. Depending on the problem, edge weights can describe various properties such as cost (e.g. travel time or distance), capacity, strength of interaction, or similarity. Also, there is another type of graph which is the unweighted one. An unweighted graph $G = (V, E)$ is equivalent to a weighted graph, where each edge $e \in E$ applies $w(e) = 1$. The diameter of a graph is the greatest shortest distance between any pair of vertices. In order to find the diameter of a graph, first, find the shortest path between each pair of vertices. Then the greatest length of any of these paths is the diameter of the graph. For example, the diameter of the graph presented in Figure B.1a is the path between nodes 1 and 6, the path is $1 \rightarrow 2 \rightarrow 4 \rightarrow 6$, with length 3.

B.2.2 Matrix Algebra

This section introduces some basic concepts from linear algebra.

Matrices

An $m \times n$ matrix consists of mn real numbers organized in m rows and n columns. The element a_{ij} of an array A is positioned at the i^{th} row and the j^{th} column. An $m \times 1$ matrix is called a column vector of order m , similarly a row vector is an $1 \times n$ matrix of order n . An $m \times n$ matrix is called a square matrix if the number of rows m is equal to the number of columns n . A square matrix is symmetric when $a_{ij} = a_{ji}$, for $i = 1, \dots, n$ and $j = 1, \dots, n$.

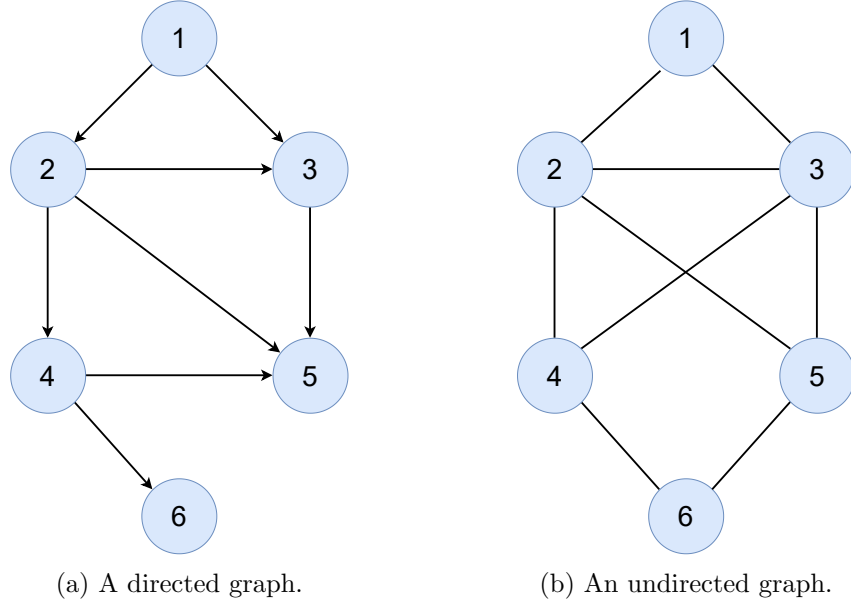


Figure B.1: Examples of a directed graph and an undirected graph.

Addition and Multiplication of Matrices

The addition of two matrices is defined only when both matrices have the same size. Let A and B two matrices with the same size, the addition is defined as follows:

$$R_{ij} = a_{ij} + b_{ij}$$

There are two possible types of multiplication in linear algebra: 1) a matrix multiplied by a scalar and 2) a matrix multiplied by a matrix. The multiplication of matrices is defined only when the number of columns of the first matrix is equal to the number of rows of the second matrix. The scalar multiplication is done by multiplying all the elements of the matrix with the scalar. The matrix multiplication is performed by multiplying the rows of the first matrix with the columns of the second matrix. The multiplication of two matrices results in a matrix with the number of rows equal to the number of rows of the first matrix and the number of columns equal to the number of columns of the second matrix. Let A be an $m \times n$ matrix and B be an $n \times p$ matrix, the resulted matrix of AB will be an $m \times p$ matrix. Let

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

the result of multiplication AB is equal to

$$AB = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}.$$

The formula for the result of $C = AB$ is defined as follows:

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj},$$

where n denotes the number of columns of the first matrix.

Determinant

The determinant of a square matrix is a scalar value that is a function of the matrix's entries. It characterizes some properties of the matrix and the linear map represented by the matrix. Specifically, the determinant of a square matrix is nonzero when the matrix is invertible. Invertible is called an $n \times n$ matrix A when there exists a matrix B of the same size such that $AB = BA = I_n$ where I_n denotes the identity matrix of order n . The identity matrix is a square matrix with ones in the main diagonal (i.e. when $i = j$) and zeros elsewhere. The determinant of a matrix A is denoted as $\det(A)$ or $|A|$.

In the case of a 2×2 matrix the determinant can be defined as

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

Similarly, in the case of a 3×3 matrix the determinant is computed as

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} = \dots$$

Eigenvalues of a Matrix

Let A be an $n \times n$ matrix. The determinant $\det(A - \lambda I_n)$ is a polynomial in the variable λ of degree n and it is called the *characteristic polynomial* of A . The equation

$$\det(A - \lambda I_n) = 0$$

is called the characteristic equation of A . This equation has n complex roots. These roots are called the eigenvalues of A .

The eigenvalues of a matrix A might not be distinct. The number of occurrences of an eigenvalue as a root of the characteristic equation is called the algebraic multiplicity of the eigenvalue. If $\lambda_1, \dots, \lambda_n$ are the eigenvalues of A , then the trace of the matrix A , the sum of entries of the main diagonal, is equal to the sum of eigenvalues,

$$\text{trace } A = \lambda_1 + \dots + \lambda_n = \sum_{i=1}^n a_{ii}.$$

To illustrate by an example, let

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Taking the determinant of $(A - \lambda I)$, the characteristic polynomial of A is

$$|A - \lambda I| = \begin{vmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{vmatrix} = \lambda^2 - 4\lambda + 3.$$

Solving the characteristic equation, its roots are $\lambda = 1$ and $\lambda = 3$, which are the two eigenvalues of A . Also, the sum of eigenvalues of A is equal to the sum of the main diagonal of A , i.e. $\text{trace } A = a_{11} + a_{22} = 2 + 2 = \lambda_1 + \lambda_2$. The eigenvalues are usually ordered as follows: $\lambda_n \leq \lambda_{n-1} \leq \dots \leq \lambda_1$

Eigenvectors of a Matrix

The eigenvectors correspond to each eigenvalue λ_i of a matrix A . They can be found by solving for the components of $\mathbf{v}$ in the equation

$$(A - \lambda I)\mathbf{v} = 0.$$

By developing the equation we get

$$(A - \lambda I)\mathbf{v} = 0 \Rightarrow A\mathbf{v} - \lambda I\mathbf{v} = 0 \Rightarrow A\mathbf{v} = \lambda\mathbf{v}.$$

By solving the system for the components x_i of vector $\mathbf{v}$ for each root of the characteristic equation of matrix A , we get the eigenvectors of the matrix A . To continue the example, the eigenvectors can be found by solving the system

$$\begin{aligned} 2x_1 + x_2 &= \lambda x_1 \\ x_1 + 2x_2 &= \lambda x_2 \end{aligned}$$

for x_1 and x_2 , for each eigenvalue λ . So, the eigenvectors of the matrix A are any nonzero scalar multiples of

$$v_{\lambda=1} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, v_{\lambda=3} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The eigenvector that corresponds to the greater eigenvalue is also called the principal eigenvector of the matrix.

B.2.3 Adjacency Matrix of a Graph

The adjacency matrix $A(G)$ of an unweighted graph $G = (V, E)$ is produced by filling in a square $n \times n$ matrix of order n , where $n = |V|$. The rows and columns of the adjacency matrix are indexed by its nodes. Each entry a_{ij} of the adjacency matrix is equal to one if there exists a link between the nodes i and j , zero elsewhere. The main diagonal of an adjacency matrix is populated by zeros. In the case of directed graphs, the adjacency matrix is populated as follows, the entry a_{ij} is equal to one if there is an edge that starts from node i and ends at the node j and zero in all other cases.

For example, the adjacency matrix of the directed graph presented in Figure B.1a is

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Instead, the adjacency matrix of the undirected graph presented in Figure B.1b is

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Properties

As shown above, the adjacency matrix of an undirected graph is a square symmetric matrix of size n , where $n = |V|$. The eigenvalues of an adjacency matrix are real and the sum of its eigenvalues is equal to the adjacency's trace which is equal to zero. Also, the adjacency matrix has an orthogonal eigenvector basis.

The greatest eigenvalue λ_1 is bounded above by the maximum degree. Let $\mathbf{v}$ be the principal eigenvector and μ be the component in which $\mathbf{v}$ has maximum absolute value. Assume v_μ is positive. Then

$$\lambda_1 v_\mu = (A\mathbf{v})_\mu = \sum_{y=1}^n A_{\mu,y} v_y \leq \sum_{y=1}^n A_{\mu,y} v_\mu = v_\mu \deg(\mu)$$

The matrix product of n copies of A has an interesting interpretation: the element (i, j) of A^n gives the number of walks of length n from vertex i to vertex j . If n is the smallest positive integer, such that for some i, j , the element (i, j) of A^n is positive, then n is the distance between the vertices i and j .

Chapter C

Distance Matrix and the Median Problem

THIS chapter gives the general definition of the median problem. Then, it presents the creation of a distance matrix and some spectral properties of it. Finally, this chapter discusses the particular topology of trees. This topology is the main focus of this study.

C.1 Median Problem

The facility location problem can be defined as follows: suppose there exist n number of sites that can be either a facility or a customer. The goal is to find which one of the n sites to be the facility/median of the network. Let c_{ij} denote the cost to transfer a good from the site i to the site j , for $i = 1, \dots, n$ and $j = 1, \dots, n$. Each site has a demand d_i of goods, for $i = 1, \dots, n$. So the median problem is given by

$$\min_j \sum_{i=1}^n c_{ij} d_i. \quad (\text{C.1})$$

For example, consider the topology shown in Figure C.1. This graph is comprised of 8 nodes, each edge of the network has a random weight in $[0, 1]$. For each node of the graph a random demand is computed. Table C.1, captures the demand d_i for each node as well as the cost C_i . The cost for each node is given by

$$C_i = \sum_{j=1}^n c_{ij}.$$

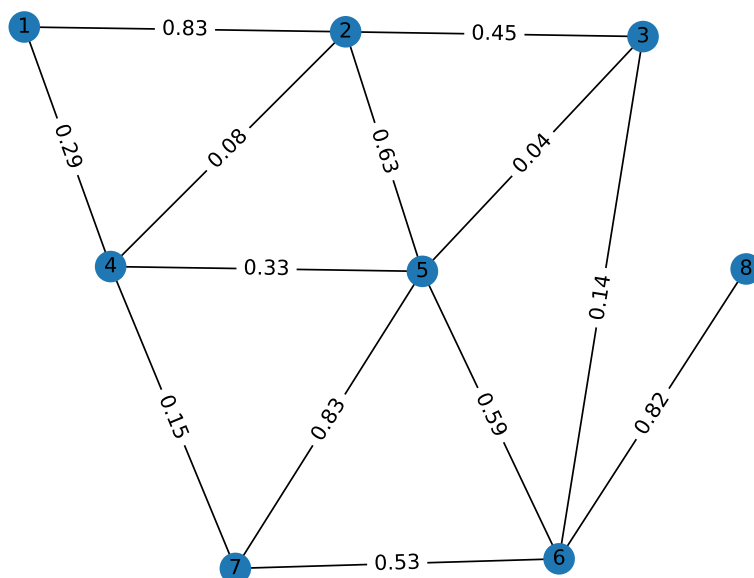


Figure C.1: A weighted graph of 8 nodes.

Hereafter, the selection of the optimal location for the facility to be placed is a deterministic process as the median node is the node with minimum cost. For this particular case, there are two nodes that have the minimum cost. Thus, either of nodes 3 or 4 could be the median of the network.

Table C.1: Characteristics of the graph presented in Figure C.1.

i	d_i	C_i
1	0.94	0.5
2	0.41	0.41
3	0.59	0.3
4	0.24	0.36
5	0.96	0.3
6	0.78	0.33
7	0.41	0.42
8	0.91	0.73

C.2 Distance Matrix

The distance matrix H of a graph $G = (V, E)$, shows the cost of the shortest distance between two nodes. The columns and rows of the distance matrix are indexed by the nodes of the graph. More specifically, the (i, j) entry of a distance matrix H shows the cost of the shortest path between the nodes i and j . Similar to the adjacency matrix, the main diagonal of the distance matrix is populated by zeros. The distance matrix of undirected graphs is symmetric, i.e. $h_{i,j} = h_{j,i}$ for $i = 1, \dots, n$ and $j = 1, \dots, n$. For example, the distance matrix H of the graph presented in Figure B.1b is

$$H = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 3 \\ 1 & 0 & 1 & 1 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 & 2 \\ 2 & 1 & 1 & 0 & 2 & 1 \\ 2 & 1 & 1 & 2 & 0 & 1 \\ 3 & 2 & 2 & 1 & 1 & 0 \end{bmatrix}.$$

C.2.1 Spectral Properties

The Distance Matrix is a square symmetric matrix and is populated by real numbers, thus its eigenvalues are real. The sum of its eigenvalues is equal to zero as the trace of the distance matrix is equal to zero. All values of the principal eigenvector of the distance matrix have the same sign.

C.3 Tree Topologies

An SDN controller knows the underlying topology of the network. This topology could be defined as a graph on n nodes. In order for the control and the data plane to be as much as possible efficient, the routing is done by selecting the shortest path for each pair of nodes. Thus, SDNs use only the minimum spanning tree of the network. If the minimum spanning tree of a network is known, someone could use it to select an optimal location to place the controller. Thus, in this thesis, an assumption is made that the tree topologies presented are the minimum spanning trees of networks.

A tree graph is an undirected graph in which any two vertices are connected by exactly one path. A tree graph G can satisfy any of the following conditions:

- G is a connected and acyclic graph (G does not contain loops).
- G is acyclic and a simple cycle can be formed by adding an edge to G .
- G is connected, but it would become disconnected if any edge is removed.

If a tree graph G has n vertices, then the above statements are equivalent to any of the following:

- G is connected and has $n - 1$ edges.
- G is connected and every subgraph of G includes at least one vertex with zero or one incident edges.
- G does not contain cycles.

C.3.1 Cases of a tree topology

All edges of the examples in this subsection have unit weight, i.e. $w_e = 1, \forall e \in E$, of a graph $G = (V, E)$. Also, the rates r_i of the vertices are all the same, i.e. $r_i = 1$ for each vertex i in V . Zelinka [46] studies the median problem in tree topologies, which corresponds to the graph's center of mass. The simplest form of a tree topology is the line topology as shown in Figures C.2a and C.2b. The line topology is a good example for someone to understand the mechanics of the median problem. There are two cases of line topologies: 1) when the number of vertices/nodes is odd and 2) when the number of vertices is even. The simplest case on the path topology is of course the first one, as the median of the graph has the same number of nodes to each side, in Figure C.2a top and bottom side.

The second case is shown in Figure C.2b. The network has an even number of nodes in a path topology. When there is an even number of nodes in a network, there is no one node that has an equal number of nodes on each side, as presented in the example the middle nodes have one node at one side and two on the other side. Thus, this topology has two medians. In the third case C.2c, in a simple tree topology, where the node with minimum distance to all other nodes is the median node.

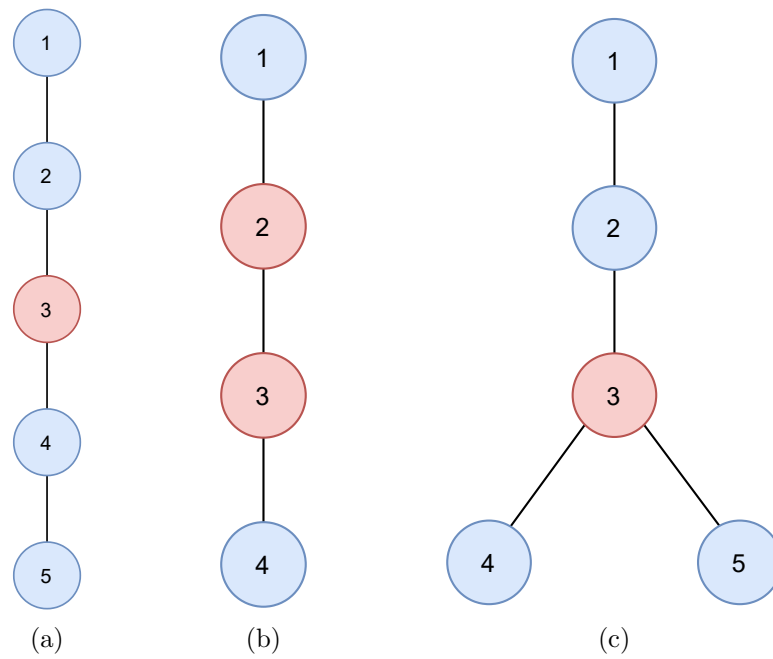


Figure C.2: Examples of tree topologies. A light red node presents the median of the graph. (a) A path topology of odd number of nodes. (b) A path topology of even number of nodes. (c) A simple case of a tree topology.

Chapter D

Proposed Method and Simulation Results

THIS chapter presents the experimental results. More specifically, it discusses the main limitations of the proposed method for finding the median of a simple tree graph. A simple tree graph is a tree graph where all edges have equal weights, i.e. $w_e = 1$ for each edge e as well as the demand of all nodes are equal. Then, the method is evaluated on tree graphs that have random weights on edges.

In order to find the optimal placement of an SDN controller, someone has to redefine the median problem to work in SDNs. Let redefine the median problem in the case of SDNs. The goal remains the same as the general median problem, i.e. find the optimal placement of the SDN controller. All the switches of the SDN network are possible candidates for being the controller. Each switch has a random *PACKET_IN* rate r_i in $[0, 1]$. The weight of each edge represents the time needed for a packet to be transmitted from one node to another. Thus, the median problem can be defined for the particular case of the SDN controller as

$$\min_j \sum_{i=1}^n c_{ij} r_i,$$

where c_{ij} denotes the shortest distance between the switches i and j .

D.1 The Proposed Method

The proposed method uses the spectral space of the distance matrix. This method selects the median from the principal eigenvector of the distance matrix. More specifically, this method selects the median by detecting the entry that is closest to zero of the principal eigenvector of the distance matrix. The index of the entry that is closest to zero corresponds to the median of the network.

D.2 Experimentation Environment

This section presents the tools that are used to take the experiments and the experimentation process. The networks are created using the NetworkX [47] library, which is a Python package that enables the creation, manipulation, and study of structure, functions, and dynamics of complex networks. Also, the NumPy [48] library is used for the manipulation of matrices. Numpy is a scientific computing library in Python that provides multidimensional array objects, an assortment of fast routines including mathematical, logical, basic linear algebra operations, etc.

First, the simulator creates a new topology or reads an existing one saved in the hard drive. Then the simulator assigns the weights on the edges and gives each node a rate. After the assignment of weights, the simulator computes the network's diameter. Then, the simulator exports the distance matrix from the produced network's topology and computes the eigenvalues and eigenvectors of the network. Next, the simulator selects a node using the proposed method and then computes the median as well.

D.3 Main limitations of the method

First, the proposed method is evaluated on all non-isomorphic permutations of trees for number of nodes smaller than or equal to 20. It is worth mentioning that the method is evaluated in topologies, where the weights of the edges are equal and the PACKET_IN rates as well. The evaluation of the method showed that the method is capable of finding the median of the network in all cases of small tree topologies, i.e. the number of nodes does not exceed 16 nodes. When the number of nodes is increased to 17 then, the method was not able to work properly in some cases. More

specifically, the number of networks that the method is not able to find the median node is nine networks. Then the method is tested on 18 nodes, the method can find the median at all cases of networks comprised of 18 nodes. In the case of 19 node networks, the method is not able to find the median in some cases of networks. The number of networks of 19 nodes, that the method is not able to find the median node, rises to 314. Similar to the 18 node networks, the method is able to find the median in all cases of networks that have 20 nodes. For networks that have more than 20 nodes, there is always a network that the method is not able to find the median.

From the experimental process, two cases are detected that the proposed method is not able to find the median. The first case is when the second smallest cost can be found two times in the table of costs. And the second case is when the smaller costs are different.

D.3.1 The case of equal costs

The Figure D.1 presents a tree network of 17 nodes. The diameter of this particular network is equal to 8 hops. A path whose length is equal to the diameter is presented with light blue nodes. The green node of the network is the median, while the light red node is the node that the proposed method selected.

Table D.1 is shorted on the values of the principal eigenvector $\mathcal{P}$ of the distance matrix. The median selected by the proposed method is node 1, which is on a path that its length is equal to the diameter of the network. The difference in mean cost between the median and the node that the proposed method selected is 0.0625, while the difference in the principal eigenvector is 0.00003. This small difference both in the cost as well as in the principal eigenvector shows that these nodes affect the network in a similar way.

Table D.1 also shows that the node 3 has the same cost as the node that the proposed method selected. Why the spectral method does not select the node with the biggest degree, as it has the same cost? Also, the proposed median seems to be influenced by the longest path of the network. It is easily observed that the proposed median “tried” to go near the middle node of the biggest path of the network.

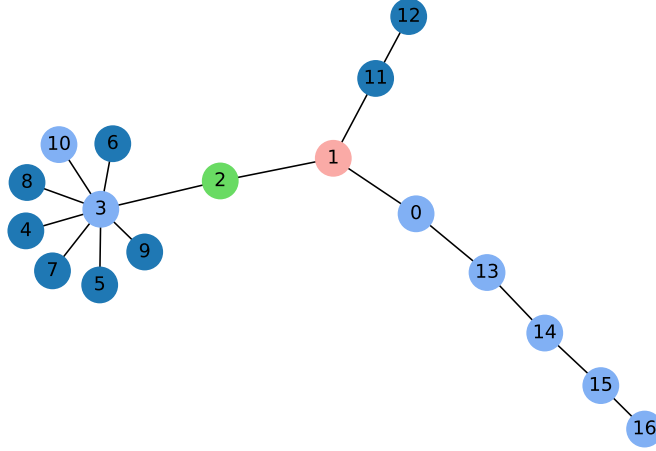


Figure D.1: The nodes with the light blue color represent a path that its length is equal to the diameter of the network. The light red node represents the median found by the proposed method. The green node corresponds to the actual median.

D.3.2 The case of different costs

The network presented in the Figure D.2 consists of 17 nodes. The diameter of this particular network is equal to 7 hops. A path whose length is equal to the network's diameter is presented with light blue nodes. The green node of the network is the median, while the light red node is the node that the proposed method selected as the median of the network.

The Table D.2 is shorted on the values of the principal eigenvector $\mathcal{P}$ of the distance matrix. The median selected by the proposed method is the node with $i = 0$, which is on a path that its length is equal to the diameter. The difference in mean cost between the actual median and the median selected with the proposed method is 0.0625, while the difference in the principal eigenvector is 0.00075. This small difference both in the cost as well as in the principal eigenvector shows that these nodes affect the network in a similar way.

Table D.2 also shows that the node with $i = 8$ has the biggest degree and is the actual median of the network. From the topology presented in Figure D.2, the node selected as the median $i = 0$ is part of the path that its length is equal to the diameter. Then, the question that remains from the case of the same costs of the neighbors is, Why the proposed method cannot find the median, but as median selects a neighboring node of the median that is closest to the middle node of a path

Table D.1: The spectral characteristics of the graph presented in Figure D.1

i	$\mathcal{P}_i$	$\overline{C}_i$	d_i
1	0.1747422	2.6250000	3
2	0.1747737	2.5625000	2
3	0.1804749	2.6250000	8
0	0.1967529	3.0625000	2
11	0.2240696	3.4375000	2
13	0.2251463	3.6250000	2
10	0.2384409	3.5625000	1
9	0.2384409	3.5625000	1
8	0.2384409	3.5625000	1
7	0.2384409	3.5625000	1
6	0.2384409	3.5625000	1
5	0.2384409	3.5625000	1
4	0.2384409	3.5625000	1
14	0.2608435	4.3125000	2
12	0.2806657	4.3750000	1
15	0.3050024	5.1250000	2
16	0.3590556	6.0625000	1

that its length is equal to the network's diameter?

D.3.3 Similar topologies

The topologies presented here have a common characteristic, that has to do with the degree of nodes. More specifically, the presented topologies have a node that has almost the half nodes of the network as neighbors. In these cases the method is not able to detect the median node. Now it is interesting to evaluate the method on similar topologies from above. These topologies could be created by changing the position of a leaf node (i.e. has one neighbor) to an arbitrary position in the network.

Using the topology presented in Figure D.1, suppose that the node with index 12 changes position. More specifically, the node with index 12 becomes neighbor of the

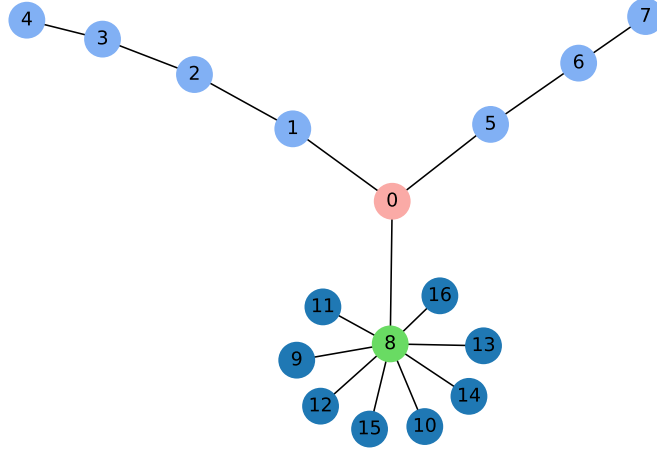


Figure D.2: The nodes with the light blue color represent a path that its length is equal to the diameter of the network. The light red node represents the median found by the proposed method. The green node corresponds to the actual median.

node that has index 0 and it is not neighbor with the node 11. It is worth mentioning that this change in the topology does not affect the diameter of the initial topology either the median of the network. The evaluation on this new topology showed that the proposed method is able to find the median of this network.

The same behavior occurs in the topology presented in Figure D.2. For example, let the node 4 of the network to change position. The node 4 is neighbor with the node that has index 0 and is not neighbor to the node 3. In this case, the diameter of the network changes and becomes 6. In contrast, the median of the network remains the same, i.e. the node with index 8. The proposed method is able to select the median node.

D.4 Experimental results

The evaluation showed that the proposed method has limitations on finding the median node in the cases of trees where the weights of edges are the same. Thus, the proposed method is evaluated on trees with random weights. More specifically, the proposed method is tested in six different sizes of networks starting from 20 nodes in 20 node increments. For each size of the network, the proposed method is tested in 1000 cases. Each tree network is created randomly and random weights,

Table D.2: The spectral characteristics of the graph presented in Figure

i	$\mathcal{P}_i$	$\overline{C}_i$	d_i
0	0.1612879	2.0625000	3
8	0.1620313	2.0000000	9
1	0.1967758	2.6250000	2
5	0.2076586	2.7500000	2
16	0.2298242	2.9375000	1
15	0.2298242	2.9375000	1
14	0.2298242	2.9375000	1
13	0.2298242	2.9375000	1
11	0.2298242	2.9375000	1
10	0.2298242	2.9375000	1
9	0.2298242	2.9375000	1
12	0.2298242	2.9375000	1
2	0.2397144	3.3125000	2
6	0.2618920	3.5625000	2
3	0.2917295	4.1250000	2
7	0.3260417	4.5000000	1
4	0.3547906	5.0625000	1

in the range $[0, 1]$, are assigned to the links that connect two nodes.

Figure D.3 depicts the mean number of hops between the node that the proposed method selected and the median of the network. More specifically, the proposed method is evaluated in six different sizes of networks from 20 nodes in 20 node increments. For each network size, 1000 different random tree topologies are created. After the creation of the network, each link between the nodes has a random weight assigned to it. The results show that the proposed method can work in large-scale topologies as the mean number of hops stay relatively stable throughout the increase of the network's size.

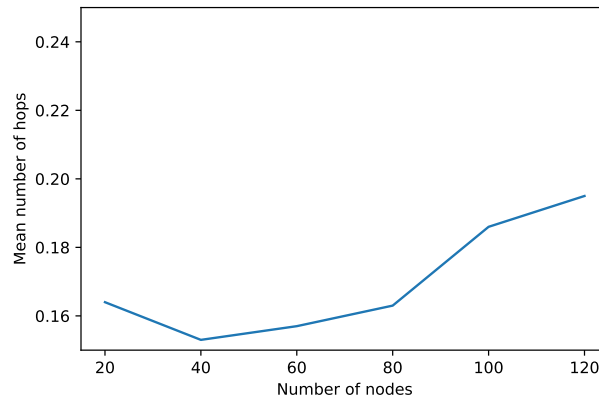


Figure D.3: The mean number of hops between the node selected using the proposed method and the median, evaluated on six different sizes of random trees. For each size of network, 1000 different cases are evaluated.

Chapter E

Summary and Conclusions

In this thesis, the facility location problem was examined, in order to be implemented in Software Defined Networks. More specifically, this study was focused in the 1-median problem applied to Software Defined Networks, using alternative approaches, that can lead the researchers to efficient solutions, even if they are not optimal, as the current approaches have enormous computational demands even in small-scale topologies.

To this end, an approach incorporating elements of algebraic graph theory was proposed. The proposed approach uses the spectral characteristics of the distance matrix to find an optimal solution to the 1-median problem. More specifically, the proposed method uses the eigenvector of the distance matrix that corresponds to its largest eigenvalue. The median is selected by finding the element of the eigenvector that is closest to zero.

The proposed approach was evaluated over various cases and number of nodes in random tree graphs. The evaluation focuses to find the optimal solution of the median problem applied to SDNs. To facilitate this evaluation process a program in Python programming language was developed. This program was responsible to create the random tree graphs, compute the distance matrix and then compare the median that the proposed method selects to the median of the network.

The proposed method was firstly evaluated on random tree graphs with each edge weight equal to one. In some cases of these graphs the proposed approach could not find the optimal solution. Then, the proposed method was evaluated on random tree graphs where the edges had random weights. The results were unexpected as

the proposed method could find the median of the graph in most cases even in large-scale topologies.

The proposed approach does not take under consideration the `PACKET_IN` rate of the switches. Also, the proposed approach is not able to be applied in a k-median problem. These limitations may be the stepping stone for future work. The future work may be directed on finding a method to incorporate the `PACKET_IN` rates to the proposed method. Additionally, an extensive examination on the cases of tree topologies that the proposed method was not able to find the median. Finally, the proposed approach would see a great improvement if it was extended to solve the k-median problem.

Bibliography

- [1] A. Doria, J. H. Salim, R. Haas, H. M. Khosravi, W. Wang, L. Dong, R. Gopal, and J. M. Halpern, “Forwarding and control element separation (forces) protocol specification.” *RFC*, vol. 5810, pp. 1–124, 2010.
- [2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.
- [3] D. Bandyopadhyay and J. Sen, “Internet of things: Applications and challenges in technology and standardization,” *Wireless personal communications*, vol. 58, no. 1, pp. 49–69, 2011.
- [4] N. L. Fantana, T. Riedel, J. Schlick, S. Ferber, J. Hupp, S. Miles, F. Michahelles, and S. Svensson, “Iot applications—value creation for industry,” *Internet of Things: Converging technologies for smart environments and integrated ecosystems*, p. 153, 2013.
- [5] R. Gómez-Chabla, K. Real-Avilés, C. Morán, P. Grijalva, and T. Recalde, “Iot applications in agriculture: A systematic literature review,” in *2nd International conference on ICTs in agronomy and environment*. Springer, 2019, pp. 68–76.
- [6] D. Kyriazis, T. Varvarigou, D. White, A. Rossi, and J. Cooper, “Sustainable smart city iot applications: Heat and electricity management & eco-conscious cruise control for public transportation,” in *2013 IEEE 14th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2013, pp. 1–5.

- [7] D. Wood, N. Apthorpe, and N. Feamster, “Cleartext data transmissions in consumer iot medical devices,” in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, 2017, pp. 7–12.
- [8] J. Kim, “Energy-efficient dynamic packet downloading for medical iot platforms,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1653–1659, 2015.
- [9] A. S. Yeole and D. R. Kalbande, “Use of internet of things (iot) in healthcare: A survey,” in *Proceedings of the ACM Symposium on Women in Research 2016*, 2016, pp. 71–76.
- [10] A. T. Thakar and S. Pandya, “Survey of iot enables healthcare devices,” in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2017, pp. 1087–1090.
- [11] D. A. Gandhi and M. Ghosal, “Intelligent healthcare using iot: a extensive survey,” in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. IEEE, 2018, pp. 800–802.
- [12] K. Kirkpatrick, “Software-defined networking,” *Communications of the ACM*, vol. 56, no. 9, pp. 16–19, 2013.
- [13] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, “A survey on software-defined networking,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2014.
- [14] M. Birk, G. Choudhury, B. Cortez, A. Goddard, N. Padi, A. Raghuram, K. Tse, S. Tse, A. Wallace, and K. Xi, “Evolving to an sdn-enabled isp backbone: key technologies and applications,” *IEEE Communications Magazine*, vol. 54, no. 10, pp. 129–135, 2016.
- [15] T. Wu, P. Zhou, B. Wang, A. Li, X. Tang, Z. Xu, K. Chen, and X. Ding, “Joint traffic control and multi-channel reassignment for core backbone network in sdn-iot: A multi-agent deep reinforcement learning approach,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 231–245, 2020.

- [16] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.
- [17] Y. Gurevich, L. Stockmeyer, and U. Vishkin, “Solving np-hard problems on graphs that are almost trees and an application to facility location problems,” *Journal of the ACM (JACM)*, vol. 31, no. 3, pp. 459–473, 1984.
- [18] R. Z. Farahani, M. SteadieSeifi, and N. Asgari, “Multiple criteria facility location problems: A survey,” *Applied mathematical modelling*, vol. 34, no. 7, pp. 1689–1709, 2010.
- [19] M. X. Goemans and M. Skutella, “Cooperative facility location games,” *Journal of Algorithms*, vol. 50, no. 2, pp. 194–214, 2004.
- [20] D.-w. Tcha and B.-i. Lee, “A branch-and-bound algorithm for the multi-level uncapacitated facility location problem,” *European Journal of Operational Research*, vol. 18, no. 1, pp. 35–43, 1984.
- [21] M. S. Daskin, L. V. Snyder, and R. T. Berger, “Facility location in supply chain design,” in *Logistics systems: Design and optimization*. Springer, 2005, pp. 39–65.
- [22] A. Ahmadi-Javid, P. Seyedi, and S. S. Syam, “A survey of healthcare facility location,” *Computers & Operations Research*, vol. 79, pp. 223–263, 2017.
- [23] İ. Gülbay, H. Acap, A. Ayözen, G. Yücel, E. Sönmez, Ç. Çelik, M. Demir, M. Paldrak, and O. Karabaşoğlu, “Location optimization of receivers for iot-based infrastructures,” in *The International Symposium for Production Research*. Springer, 2020, pp. 930–942.
- [24] A. T. Murray and R. L. Church, “Applying simulated annealing to location-planning models,” *Journal of Heuristics*, vol. 2, no. 1, pp. 31–53, 1996.
- [25] T. Levanova and A. Y. Gnusarev, “Simulated annealing for competitive p-median facility location problem,” in *Journal of Physics: Conference Series*, vol. 1050, no. 1. IOP Publishing, 2018, p. 012044.

- [26] S. Basu, M. Sharma, and P. S. Ghosh, “Metaheuristic applications on discrete facility location problems: a survey,” *Opsearch*, vol. 52, no. 3, pp. 530–561, 2015.
- [27] M. Marić, Z. Stanimirović, and N. Milenković, “Metaheuristic methods for solving the bilevel uncapacitated facility location problem with clients’ preferences,” *Electronic Notes in Discrete Mathematics*, vol. 39, pp. 43–50, 2012.
- [28] N. Krivulin, “Algebraic solution of minimax single-facility constrained location problems with chebyshev and rectilinear distances,” *Journal of Logical and Algebraic Methods in Programming*, vol. 115, p. 100578, 2020.
- [29] —, “An algebraic approach to multidimensional minimax location problems with chebyshev distance,” *arXiv preprint arXiv:1211.2425*, 2012.
- [30] K. Jakob and P. M. Pruzan, “The simple plant location problem: Survey and synthesis,” *European journal of operational research*, vol. 12, pp. 36–81, 1983.
- [31] D.-W. Tcha and M.-G. Yoon, “A dual-based heuristic for the simple facility location problem with stochastic demand,” *IIE transactions*, vol. 17, no. 4, pp. 364–369, 1985.
- [32] R. D. Galvão and L. A. Raggi, “A method for solving to optimality uncapacitated location problems,” *Annals of Operations Research*, vol. 18, no. 1, pp. 225–244, 1989.
- [33] D. Erlenkotter, “A dual-based procedure for uncapacitated facility location,” *Operations Research*, vol. 26, no. 6, pp. 992–1009, 1978.
- [34] B. Goldengorin, D. Ghosh, and G. Sierksma, “Branch and peg algorithms for the simple plant location problem,” *Computers & Operations Research*, vol. 30, no. 7, pp. 967–981, 2003.
- [35] B. Heller, R. Sherwood, and N. McKeown, “The controller placement problem,” *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 473–478, 2012.

- [36] A. Sallahi and M. St-Hilaire, “Optimal model for the controller placement problem in software defined networks,” *IEEE communications letters*, vol. 19, no. 1, pp. 30–33, 2014.
- [37] M. S. Daskin, L. V. Snyder, and R. T. Berger, “Facility location in supply chain design,” in *Logistics systems: Design and optimization*. Springer, 2005, pp. 39–65.
- [38] M. T. Melo, S. Nickel, and F. Saldanha-Da-Gama, “Facility location and supply chain management—a review,” *European journal of operational research*, vol. 196, no. 2, pp. 401–412, 2009.
- [39] A. Meyerson, “Online facility location,” in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, 2001, pp. 426–431.
- [40] M. Pal, T. Tardos, and T. Wexler, “Facility location with nonuniform hard capacities,” in *Proceedings 42nd IEEE symposium on foundations of computer science*. IEEE, 2001, pp. 329–338.
- [41] J.-M. Robert and G. Toussaint, “Computational geometry and facility location,” in *Proc. International Conference on Operations Research and Management Science*. Citeseer, 1990.
- [42] J. Cardinal and S. Langerman, “Min-max-min geometric facility location problems,” in *Proc. European Workshop on Computational Geometry (EWCG’06)*, 2006, pp. 149–152.
- [43] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 1984, pp. 302–311.
- [44] M. Conforti, G. Cornuéjols, G. Zambelli *et al.*, *Integer programming*. Springer, 2014, vol. 271.
- [45] R. B. Bapat, *Graphs and matrices*. Springer, 2010, vol. 27.
- [46] B. Zelinka, “Medians and peripherians of trees,” *Archivum Mathematicum*, vol. 4, no. 2, pp. 87–95, 1968.

- [47] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using networkx,” Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.
- [48] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>

Appendices

Appendix I

Glossary

Switch	A device that connects the devices of the network using packet switching to receive and forward data.
Control Plane	Manages the resources of the network to minimize the network utilization.
Data Plane	Forwards the packets from source to destination.
Controller	The responsible node of the network that manages the network resources.
Adjacency Matrix	A square matrix that represents a graph, its elements indicate whether a pair of nodes are adjacent or not.
Acyclic Graph	A graph that does not contain loops.
Characteristic Polynomial	The polynomial produced from the determinant $\det(A - \lambda I)$ on variable λ .
Characteristic Equation	The equation produced by setting the characteristic polynomial equal to zero.
Vector	A 2D matrix that the size of one dimension is equal to zero.
Degree	In graph theory, the number of neighbors of a node.
Diameter	Number of hops contained in the longest shortest path between nodes of a graph.
Directed Graphs	Graphs that its edges have a direction from one node to another one.
Distance Matrix	A matrix that its entries correspond to the distance between respective the nodes.

Eigenvalue	A solution of the charecteristic equation.
Eigenvector	An eigenvector v is produced by solving the equation: $Av = \lambda v$, where λ is an eigenvalue of the matrix A .
Identity Matrix	A square matrix that its etries are equal to zero appart from the main diagonal where there are equal to one.
Median Problem	The problem on finding the node that has the minimum cost between the network's nodes.
Minimum Spanning Tree	A minimum spanning tree is a subset of edges of a connected graph, that connects all vetrices without any cycles and with minimum possible total edge weight.
Principal Eigenvector	The eigenvector of a matrix that coresponds to the biggest eigenvalue.
Square Matrix	A two dimentional matrix that its dimetions sizes are equal.
Simetric Matrix	A matrix is called simetric when transposed matrix is the same with the original one.
Trace	The sum of the main diagoanal of a square matrix.

Appendix II

Notations

G	A graph
V	Set of network nodes
E	Set of network edges
A	The adjacency matrix
H	The distance matrix
d_i	The degree of node i
$\overline{C}_i$	The mean cost of the node i
λ_i	The i^{th} eigenvalue
v_i	An eigenvector that corresponds to λ_i eigenvalue
$\mathcal{P}_i$	The i^{th} entry of the principal eigenvector of the distance matrix H

Appendix III

Abbreviations

SDNs	Software Defined Networks
5G	Fifth-generation networks
IoT	Internet of Things

Index

- acyclic graph, 22
- adjacency matrix, 16
- algebraic graph theory, 4
- algebraic multiplicity, 15
- capacitated facility location problem, 10
- characteristic equation, 15
- characteristic polynomial, 14
- column vector, 12
- controller, 3, 25
- degree, 11
- determinant, 14
- diameter, 12
- directed graphs, 12
- distance matrix, 21, 26
- edges, 11
- eigenvalue, 15
- eigenvector, 15
- facility location problem, 4, 9, 19
- graph, 11
- identity matrix, 14
- invertible matrix, 14
- matrix addition, 13
- matrix multiplication, 13
- median problem, 19, 25
- minimum spanning tree, 21
- principal eigenvector, 16, 26
- row vector, 12
- Software Defined Networks, 2
- square matrix, 12
- symmetric matrix, 12
- trace, 15
- tree graph, 22
- tree topology, 21
- uncapacitated facility location problem, 11
- undirected graphs, 11
- unweighted graph, 12
- vector, 12
- vertices, 11
- weighted graphs, 12