# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**PURPOSE-DRIVEN COMMUNITIES IN MULTIPLEX NETWORKS: THRESHOLDING USER-ENGAGED LAYER AGGREGATION**

by

Ryan E. Miller

June 2016

| | |
|---|---|
| Thesis Advisor: | Ralucca Gera |
| Second Readers: | Gerry Baumgartner |
| | Matthew Carlyle |

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

# REPORT DOCUMENTATION PAGE

*Form Approved OMB No. 0704–0188*

*Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.*

| 1. AGENCY USE ONLY *(Leave Blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | 06-17-2016 | Master's Thesis 07-05-2014 to 06-17-2016 |

**4. TITLE AND SUBTITLE**

PURPOSE-DRIVEN COMMUNITIES IN MULTIPLEX NETWORKS: THRESHOLDING USER-ENGAGED LAYER AGGREGATION

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**

Ryan E. Miller

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Naval Postgraduate School
Monterey, CA 93943

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Laboratory for Telecommunication Sciences

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(maximum 200 words)*

Discovering true and meaningful communities in dark networks is a non-trivial yet useful task. Because terrorists work hard to hide their relationships/network, analysts have an incomplete picture of their strategy; even worse, the degree of incompleteness is unknown. To better protect our nation, analysts would benefit from a tool that helps them identify meaningful terrorist communities. This thesis introduces a general-purpose algorithm for community detection in multiplex dark networks using the layers of the network based on edge attributes. The methodology includes community detection details from each layer, yet it is still flexible enough to be meaningful in a variety of networks based on the user's interest. The aim of this thesis is to build on current layer aggregation methodologies as well as preexisting community detection algorithms. We apply our algorithm to three multiplex terrorist networks: Noordin Top Network, Boko Haram and Fuerzas Armadas Revolucionarias de Colombia (FARC). We validate our algorithm by measuring adjusted conductance and cluster adequacy with respect to community quality. We demonstrate the utility of our community partitions by developing a community guided network shortest path interdiction model, which disrupts the information flow in the Noordin Top Network.

**14. SUBJECT TERMS**

community detection, network science, layer aggregation, dark networks, conductance, cluster adequacy, modularity, Louvain method, shortest path interdiction

**15. NUMBER OF PAGES** 155

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

# PURPOSE-DRIVEN COMMUNITIES IN MULTIPLEX NETWORKS: THRESHOLDING USER-ENGAGED LAYER AGGREGATION

Ryan E. Miller
Captain, United States Army
B.S., University of Virginia, 2008

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN APPLIED MATHEMATICS**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2016**

Approved by: Ralucca Gera
Thesis Advisor

Gerry Baumgartner
Second Reader

Matthew Carlyle
Second Reader

Craig Rasmussen
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Discovering true and meaningful communities in dark networks is a non-trivial yet useful task. Because terrorists work hard to hide their relationships/network, analysts have an incomplete picture of their strategy; even worse, the degree of incompleteness is unknown. To better protect our nation, analysts would benefit from a tool that helps them identify meaningful terrorist communities. This thesis introduces a general-purpose algorithm for community detection in multiplex dark networks using the layers of the network based on edge attributes. The methodology includes community detection details from each layer, yet it is still flexible enough to be meaningful in a variety of networks based on the user's interest. The aim of this thesis is to build on current layer aggregation methodologies as well as preexisting community detection algorithms. We apply our algorithm to three multiplex terrorist networks: Noordin Top Network, Boko Haram and Fuerzas Armadas Revolucionarias de Colombia (FARC). We validate our algorithm by measuring adjusted conductance and cluster adequacy with respect to community quality. We demonstrate the utility of our community partitions by developing a community guided network shortest path interdiction model, which disrupts the information flow in the Noordin Top Network.

THIS PAGE INTENTIONALLY LEFT BLANK

# Table of Contents

**1 Introduction** **1**
1.1 Problem Description . . . . . 2
1.2 Thesis Contribution . . . . . 3
1.3 Organization . . . . . 5

**2 Background** **7**
2.1 Network Science Overview . . . . . 7
2.2 General Community Detection . . . . . 15
2.3 Multilayer Community Detection . . . . . 25
2.4 Dark Networks . . . . . 28

**3 Data and Methodology** **35**
3.1 Data Description . . . . . 35
3.2 Methodology Overview . . . . . 48

**4 Results and Analysis** **63**
4.1 Experiment Design . . . . . 63
4.2 Noordin Results and Analysis . . . . . 66
4.3 Boko Haram Results and Analysis . . . . . 90
4.4 FARC Results and Analysis . . . . . 93
4.5 General Observations . . . . . 95

**5 Modeling and Application** **99**
5.1 Model Formulation . . . . . 99
5.2 Noordin Formulation . . . . . 102
5.3 Community Properties and Attack Strategy . . . . . 105

**6 Future Work and Recommendations** **115**
6.1 Community Detection Algorithm Improvements . . . . . 115
6.2 Network Flow Model Enhancements. . . . . 119

6.3 Alternative Disruption Strategies . . . . . 122
6.4 Conclusions . . . . . 124

**List of References** **127**

**Initial Distribution List** **133**

# List of Figures


Figure 2.1 Human HIV-1 genetic interaction network. . . . . 9

Figure 2.2 HIGGS multiplex social interaction Twitter data. . . . . 10

Figure 2.3 Multiplex European Airport Network. . . . . 12

Figure 2.4 Block Models and Network Community Profiles: (a) Zachary Karate Club (b) Core-periphery structure example (c) Erdős-Rényi graph. (d) bipartite block model example. . . . . 18

Figure 2.5 Understanding modularity using different community partitions of a network: (a) Optimal $M = 0.41$ (b) Suboptimal $M = 0.22$ (c) One Community $M = 0$ (d) Negative $M = -0.12$. . . . . 21

Figure 2.6 Louvain method using fast greedy at Step 1 and collapsing communities at Step 2 for a total of two iterations. . . . . 23

Figure 3.1 An overview of the Noordin Network. . . . . 37

Figure 3.2 Noordin Network monoplex, $O$. . . . . 38

Figure 3.3 Noordin Network weighted degree distribution. . . . . 39

Figure 3.4 An overview of the Boko Haram Network. . . . . 41

Figure 3.5 Boko Haram Network monoplex, $O$. . . . . 42

Figure 3.6 Boko Haram Network weighted degree distribution. . . . . 43

Figure 3.7 An overview of the FARC Network. . . . . 45

Figure 3.8 Boko Haram Network monoplex, $O$. . . . . 46

Figure 3.9 FARC Network weighted degree distribution. . . . . 47

Figure 3.10 Algorithm overview (general case). . . . . 50

Figure 3.11 Algorithm overview (Noordin example) . . . . . 51

Figure 3.12 Step 1: Layer selection (Noordin example). . . . . 55

Figure 3.13 Step 2: Weighted category sorting (Noordin example). . . . . 58

Figure 3.14 Step 3: Community detection algorithm (Noordin example). . . . 58

Figure 3.15 Step 4: Community to clique conversion (Noordin example). . . 59

Figure 3.16 Step 5: Weighted graph, $W$ (Noordin example). . . . . 60

Figure 3.17 Step 6: KSCs and plot in $O$ (Noordin example). . . . . 61

Figure 4.1 Chapter 4 case study organization. . . . . 63

Figure 4.2 Noordin control case community output plot and size, and conductance plot. . . . . 66

Figure 4.3 Noordin community output plot for subcases 1.1-1.3 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. . . . . 68

Figure 4.4 Noordin community size and normalized conductance for subcases 1.1-1.3 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. . . . . 69

Figure 4.5 Noordin community output plot for subcases 1.4-1.6 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. . . . . 70

Figure 4.6 Noordin community size and normalized conductance for subcases 1.4-1.6 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. . . . . 71

Figure 4.7 Noordin community output plot for subcases 1.7-1.9 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. . . . . 72

Figure 4.8 Noordin community size and normalized conductance for subcases 1.7-1.9 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. . . . . 73

Figure 4.9 Noordin community output plot for subcases 2.1-2.3 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$. . . . . 75

Figure 4.10 Noordin community size and normalized conductance for subcases 2.1-2.3 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$. . . . . 76

Figure 4.11 Noordin community output plot for subcases 2.4-2.6 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$. . . . . 77

Figure 4.12 Noordin community size and normalized conductance for subcases 2.4-2.6 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$. . . . . 78

Figure 4.13 Noordin community output plot for subcases 2.7-2.9 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$. . . . . 79

Figure 4.14 Noordin community size and normalized conductance for subcases 2.7-2.9 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$. . . . . 80

Figure 4.15 Noordin community output plot for subcases 3.1-3.3 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$. . . . . 82

Figure 4.16 Noordin community size and normalized conductance for subcases 3.1-3.3 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$. . . . . 83

Figure 4.17 Noordin community output plot for subcases 3.4-3.6 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$. . . . . 84

Figure 4.18 Noordin community size and normalized conductance for subcases 3.4-3.6 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$. . . . . 85

Figure 4.19 Noordin community output plot for subcases 3.7-3.9 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$. . . . . 86

Figure 4.20 Noordin community size and normalized conductance for subcases 3.7-3.9 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$. . . . . 87

Figure 4.21 Average community size, average normalized conductance, and cluster adequacy from communities plotted in $O$ and $W$ for Noordin cases 1-3. . . . . 89

Figure 4.22 Community size and adjusted conductance for Boko Haram control case. . . . . 90

Figure 4.23 Average community size, average adjusted conductance, and cluster adequacy for Boko Haram cases 1-3. . . . . 91

Figure 4.24 Community size and adjusted conductance for FARC control case. 93

Figure 4.25 Average community size, average adjusted conductance, and cluster adequacy for FARC cases 1-3. . . . . 94

Figure 5.1 Cost in hours for attack plans in uniform and hierarchical cost models. 104

Figure 5.2 Subcase 3.9 community properties. . . . . 108

Figure 5.3 Noordin original and simplified attack models. . . . . 110

Figure 5.4 Uniform cost results. . . . . 112

Figure 5.5 Hierarchical cost results. . . . . 113

Figure 6.1 Vertex splitting example. . . . . 120

# List of Tables


Table 3.1 Noordin Network topological characteristics by layer. . . . . 39

Table 3.2 Boko Haram Network topological characteristics by layer. . . . . 43

Table 3.3 FARC Network topological characteristics by layer. . . . . 47

Table 3.4 Noordin Network topological characteristics by category. . . . . 56

Table 3.5 Boko Haram Network topological characteristics by category. . . . 57

Table 3.6 FARC Network topological characteristics by category. . . . . 57

Table 5.1 Subcase 3.9 community properties. . . . . 107

Table 5.2 Subcase 3.9 community total influence summary. . . . . 107

Table 6.1 Noordin case 3 misfit elimination $\Delta$. . . . . 118

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Acronyms and Abbreviations

| | |
|---|---|
| **ACC** | Average Clustering Coefficient |
| **AD** | Average Degree |
| **APL** | Average Path Length |
| **AtN** | Attack the Network |
| **AWD** | Average Weighted Degree |
| **BCE** | Between Community Edge |
| **CI** | Community Influence |
| **FARC** | Fuerzas Armadas Revolucionarias de Colombia |
| **FF** | Fragmented Functionality |
| **JIEDDO** | Joint Improvised Explosive Device Defeat Organization |
| **KSC** | Knowledge Sharing Community |
| **LOC** | Lines of Communication |
| **NCP** | Network Community Profile |
| **NDV** | Number of Demand Vertices |
| **NMI** | Normalized Mutual Information |
| **NPS** | Naval Postgraduate School |
| **PDC** | Purpose Driven Community |
| **SCF** | State of Critical Functionality |
| **TEEC** | Total External Edge Count |
| **TF** | Total Functionality |

**TI** Total Influence

**TIEC** Total Internal Edge Count

# Executive Summary

Network science allows us to visualize large data sets in the form of a mathematical model. Partitioning a network into communities based on its topology, helps reduce the complexity of large networks by placing vertices into groups based on similar attributes. Detecting communities in single layer networks is a well-studied problem. However, detecting communities in multiplex networks that contain many layers is challenging. Network layer aggregation approaches reduce a multiplex network to a single weighted graph, which simplifies the network to a single layer community detection problem. However, aggregating all of the layers causes the detailed information associated with each layer to be lost. A new algorithm that detects communities in multiplex networks that reduces the cost of information loss is needed.

This thesis proposes a purpose-driven community detection algorithm for multiplex networks that is user-engaged at multiple steps to develop analytically useful communities. The algorithm focuses on a user-defined goal, which directs the algorithm to select and combine layers appropriately in support of that goal. In addition, the user selects weights and an information threshold that results in a spectrum of community numbers and sizes. To test our algorithm, we used three dark network data sets from the NPS Common Operational Research Environment Lab, with a user defined goal of network disruption. We specifically tailored the algorithm to reduce the effects of incomplete information on dark network analysis.

In total, we explored 81 subcases from our dark networks that included different weights and information threshold choices. To determine community quality, we measured cluster adequacy and average adjusted conductance of the resultant communities from each subcase. The community quality generally increased with the size of the community. The larger communities were developed under the provisions of the most relaxed threshold values. However, we also observed that graph components that were identified as communities resulted in perfect community quality scores regardless of the community size.

The main purpose of our community development was to disrupt a terrorist network. With this goal in mind, we formulated a community guided shortest path interdiction network flow model. Subcase 3.9 provided the necessary community compositions to guide the

shortest path interdiction model towards a faster solution. We recommend more trials using other subcases to reveal the optimal community composition. However, subcase 3.9 demonstrated the utility of our communities to reduce the complexity of our network model from 1196 to 698 edges. We believe there exists an optimal community composition for each network, which depends on the associated community purpose as well as community quality.

This thesis presented an alternative method for conducting community detection in multiplex networks. By analysing our resultant community properties, we enhanced current optimal shortest path interdiction results. The community guided approach achieved similar optimal results while significantly reducing solution time. Our focus on first defining a purpose for community detection helped guide our algorithm development into a working procedure with tangible results. We believe that detecting purpose-driven communities in multiplex networks by thresholding user-engaged layer aggregation is a promising area of research that should be continued and examined with more data sets in the future.

# Acknowledgments

This thesis would not have been possible without the support of my family. My wife, Lynsey, and daughter, Catherine, have been a constant source of strength throughout my time here at the Naval Postgraduate School. My family has fostered an environment that stimulates intellectual curiosity while still keeping me grounded in reality. The balance between work and family is difficult to maintain, yet my family's patience with me during my graduate school time has been amazing. Lynsey and Catherine are truly a blessing, and I'm looking forward to our next adventure together.

My advisor, Ralucca Gera, and my second readers, Matthew Carlyle, and Gerry Baumgartner, have guided me along this intellectual journey of discovery. Ralucca and Matthew allowed me the opportunity to combine applied mathematical theory with operations research application. Ralucca provided me with the mathematical foundation for this thesis and constantly pushed me to question my own understanding of network science and to keep searching for more knowledge. She has invested a lot of time in my development as a researcher and a teacher. Matthew helped me develop a modeling application to test the algorithm we developed for this thesis. His linear programming skills and network flow modeling were essential to the success of this thesis. Discussions with Gerry and his research team helped me view my research from a different perspective, fill in knowledge gaps, and enhance the overall quality of this thesis.

Special thanks to Akrati Saxena for her exceptional Python coding skills and research working group participation. She was instrumental in transforming our detection algorithm into code and provided additional research papers on community detection for consideration. Exchanging research ideas with her has been a rewarding experience that fosters the very spirit of learning in an academic environment.

I would also like to thank Sean Everton and Daniel Cunningham for providing the three dark network data sets. Without this data to test our algorithm, this thesis would have remained strictly theoretical. Discussions with Sean also helped steer our metrics for testing community quality and helped provide an intellectual sanity check on our community detection algorithm.

My journey to graduate school would not have been possible without the letters of recommendation from Richard Miksad, Jose Gomez, and Greg Phillips. I am tremendously grateful for their collective confidence in me to excel at the graduate level.

Although not directly involved in my thesis, I'd like to thank the entire Applied Mathematics and Operations Research Departments. My mathematical foundation, research skills, and curiosity were molded and refined through all of the time each of my professors invested in my education. My fellow classmates, Karoline Hood and Scott Warnke, invested countless hours in studying and reviewing concepts with me to complete our class assignments and grow in our knowledge together at NPS.

I would also like to thank Michelle Pagnani for the thesis writing workshops and reviewing my writing. Her guidance helped me to become a better writer. The writing process is challenging, and I could not have produced a coherent thesis without her assistance.

Finally, I'd like to thank the developers of Gephi, Pyomo, Python, and Gurobi for providing the networks science analytical platforms for developing, modeling, and testing our community detection algorithm. I appreciate all of the support I have received throughout my time at NPS and look forward to continuing my educational journey in the future.

# CHAPTER 1:
# Introduction

Beginning early in our lives, we are instructed to develop our skills to describe and understand the complex world around us. We label, or define, other people based on our type of connection or relationship to them. Social networks such as LinkedIn offer a platform for capturing our professional relationships, and we can visually describe or model these relationships using tools from *network science*. This abstract model can be created by representing each object as a dot or *vertex* with a corresponding label. This group of objects could then be divided or partitioned into smaller groups, or *communities*, based on similar attributes. Any two vertices that are related by the same attribute form a connection, or *edge*, between them. Modeling the information from LinkedIn as a graph and analyzing the graph using network science enables us to mathematically articulate relationships. This leads to increased understanding of the local and global importance of people, or groups, of people in our graph.

For example, we can arrange the people from a LinkedIn network into communities based on physical attributes or other characteristics such as age, gender, type of profession, educational background, job title, or history of employment. Which arrangement is correct? All arrangements are technically correct, but one relationship or set of relationships may be more appropriate depending on our goal. First, we need to understand why we are sorting people into communities. Imagine we are retiring from the military and searching for a new job that requires credible references. This end-user goal focuses our choice(s) of relationships or layers on type of profession, educational background, job title, and history of employment.

We can build a graph for each relationship by first plotting each vertex and then connecting vertices with an edge if they have that relationship. For example, in the graph of the profession layer, two people are connected if they have the same type of profession. This visualisation of relationships as graphs allows us to identify communities based on the type of profession.

Network science is concerned with describing systems, such as social interactions, by

representing objects and their relationships as graphs with information known as networks. Typically the term graph is used in theoretical contexts whereas the term network is used in reference to the application of graph theory. For the purposes of this thesis, graph and network are used interchangeably. If we want to describe our system using multiple relationships between the same vertices, Bianconi [1] explains we can build collective layers of graphs called *multiplex networks*. Defining and detecting communities across multiple layers with large and diverse information data sets can become increasingly complex. According to Radicchi et al. [2], *community detection* can be applied to help understand and solve numerous technical and social problems. In the next section, we illustrate the utility of community detection and describe the associated challenges of applying community detection to multiplex dark networks.

## 1.1 Problem Description

Fear is not a new concept, yet organizations whose purpose is to spread fear remain difficult to fully comprehend. These groups are known by many names, such as terrorists, insurgents, or simply criminal organizations. In social network analysis, Bakker et al. [3] refer to these organizations as dark networks. The complex structure of dark networks challenges network science to develop more precise analytical methods to model and enhance our understanding of these networks. Section 2.4 covers dark networks and their associated analytical challenges in more detail. The following motivational anecdote builds from my professional military experience with dark networks as an Explosive Ordnance Disposal Officer.

As the senior Counter-Improvised Explosive Device Officer for Nangarhar Province in Afghanistan, my mission required me to eliminate explosive threats produced by dark networks and to provide counsel to the combatant commander on predicting and preventing future attacks. To succeed at this mission, my teams required information to begin mapping out the networks.

The evidence and intelligence my teams gathered as part of sensitive site exploitation was catalogued and processed for the dual purpose of prosecuting members of dark networks and assembling targeting packages for future missions. My reports were supplemented by other intelligence sources and reports from various other units to form a collective database.

Over the years, this temporal database has grown to include a wide spectrum and high volume of information. How do we take advantage of this data to enhance our analysis of a given network? Researchers visualize each aspect of a diverse data set as independent graphs within a complex network.

Network science has developed several tools for analyzing single layers or aggregated weighted graphs, which Kivelä et al. [4] define as monoplex networks. Community detection is one such tool that, when wielded appropriately, can increase our understanding of dark networks. Community detection partitions the vertices of the graph into densely connected groups. The properties of these communities can be studied both locally and within the context of the global graph to build community profiles. The knowledge gained through community profiles has the potential to assist the analyst in developing more robust targeting packages for network disruption. Developing a method that maximizes the information gathered increases the depth of the analysis by producing more meaningful profiles of network communities.

Analyzing layers independently or collapsing all layers into a monoplex network both fail to capture the true details of the multiplex network. In the first case, we do not study the network holistically, and in the second, we lose information by oversimplifying the network. Researchers have made substantial progress on detecting communities in single layer networks. However, community detection in multiplex and multilayer networks has proven to be particularly challenging. Several algorithms have successfully detected communities on synthetic networks. However, when many of these algorithms are implemented on real networks, most have difficulty partitioning the network into the predetermined communities. In this thesis, we explore a mathematical approach to defining and detecting communities in multiplex networks.

## 1.2 Thesis Contribution

This research seeks a general purpose algorithm for multiplex networks that is detailed enough to detect meaningful communities as well as flexible enough to be applied to a variety of networks. The aim of this thesis is to build on research conducted on the merits of layer aggregation methodologies used in multilayer community detection.

This thesis proposes a new algorithm that sorts the layers into aggregate weighted categories

to enhance network data integrity and ultimately, to detect more meaningful communities. Our method allows the user to choose the appropriate community detection algorithm and the threshold that produces the most relevant community partition. We claim that this new algorithm enhances network data integrity, resulting in more analytically meaningful partitioned communities than current layer aggregation methods. Flexibility is achieved by engaging the user at multiple stages throughout the methodology implementation process, but also by offering a default. User input develops detailed and meaningful communities within the context of the user's analytical goals.

The goal of our proposed methodology is to increase the analytical depth of the resultant multiplex communities. This objective is achieved by first allowing the user to choose the appropriate combination of layers and weights per category. Next, the user picks the appropriate community detection algorithm based on the data. Finally, the user enhances both of these choices by selecting the threshold that gives the most relevant community partition in the multiplex. This thesis focuses on real network data sets and attempts to extend the methodology for general purposes. The resultant communities from this proposed algorithm have the potential to enhance our current understanding of multiplex networks. When this increased understanding is specifically applied to dark networks, it has the potential to aid analysts in network disruption and consequently, to restore safety and stability to terror inflicted regions.

In this thesis, we examine three dark multiplex network case studies to test our algorithm. The Noordin Top Network provided the inspiration for our method and is discussed in great detail. The Fuerzas Armadas Revolucionarias de Colombia (FARC) and Boko Haram Terrorist Networks were used as further verification of our methodology. For each network, we validate our algorithm by determining the adjusted conductance and cluster adequacy of the resultant communities. To demonstrate the utility of finding communities for network disruption purposes, we built a network flow shortest path interdiction model. The model determines the optimal strategy, given a finite number of attacks, to disrupt the flow of information from a set of supply sources to a set of demand destinations. We enhance the optimal solution strategy for this model by examining the properties of the detected communities in the Noordin Network. The goal of this enhancement is to achieve similar optimal solutions while increasing the algorithm performance efficiency.

## 1.3 Organization

This thesis is organized into six chapters including Introduction, Background, Data and Methodology, Results and Analysis, Modeling and Application, and Future Work and Recommendations. In the next chapter, Chapter 2, we examine prior work in network science on community detection algorithms and dark networks. Chapter 3 provides an overview of our data sets and a detailed explanation and justification for our methodology using the Noordin Top Network. Chapter 4 presents our community detection results for different threshold values in each data set and discusses the resultant community topological characteristics, modularity, and conductance plots. Chapter 5 develops an attack and defend model that demonstrates the application of the results from Chapter 4. The final chapter, Chapter 6, recommends some potential extensions of this research to improve our detection algorithm and our approach to disrupting networks using community properties.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 2:
# Background

In this chapter, we explore prior research contributions to community detection in multilayer networks. This research is organized into four sections including Network Science Overview, General Community Detection, Multilayer/Multiplex Community Detection, and Dark Networks. Network Science Overview introduces the reader to the basic terms and concepts used in this mathematical field. General Community Detection covers some of the challenges and current algorithms implemented for single layer networks. Multilayer/Multiplex Community Detection specifically focuses on efforts to develop algorithms that are applicable to multilayer or multiplex networks. Dark Networks highlights the prior research that has been conducted to specifically analyze the dark network case studies we examine in Chapter 3.

## 2.1 Network Science Overview

Network science is a relatively new and progressive area of study within the field of discrete mathematics. According to Newman [5], "network science is concerned with understanding and modeling the behaviour of real-world networked systems." Notably, depending on the field of study and context, many authors use nodes and vertices interchangeably in reference to an object. For clarity purposes, vertices is used exclusively in this thesis. Network science builds upon the mathematical framework established by graph theory. The study of graphs provides the foundation for all of the analytical tools network science has developed to describe complex systems. In Chapter 1, we introduced the concept of a graph using edges and vertices. According to Bollobás [6], a graph, $G$, is defined as:

**Definition 2.1.1.** Graph

> an ordered pair of finite disjoint sets $(V, E)$ such that $E$ is a subset of the set $V \times V$ of unordered pairs of $V$. The set $V$ is the set of vertices and $E$ is the set of edges. If $G$ is a graph, then $V = V(G)$ is the vertex set of $G$, and $E = E(G)$ is the edge set. An edge {x, y} is said to join the vertices $x$ and $y$ and is denoted

by $xy$. Thus $xy$ and $yx$ means exactly the same edge; the vertices $x$ and $y$ are the end vertices of this edge.

Definition 2.1.1 can be generalized to networks to describe complex systems, by capturing more than just the vertex-to-vertex relationship. Networks are more complex than graphs, but they still simplify reality to develop a mathematical model for analytical purposes. Newman [5] explains this connection by defining a network as:

**Definition 2.1.2.** Network

> a simplified representation that reduces a system to an abstract structure capturing only the basics of connection patterns and little else.

Definition 2.1.2 can be augmented to describe diverse data sets as multilayered networks. Kivelä et al. [4] introduce the term *aspect*, $d$, where an aspect represents a different level of dimensionality within a layer. For example, one aspect of a layer could be time, while another aspect of the same layer could be displacement. Kivelä et al. further explain that an *elementary layer* refers to one aspect and they distinguish the term *layer* to mean the combination of elementary layers that belong to all aspects, much like the category of elementary layers we will use in our research. The notation $L$ represents a sequence of sets of elementary layers, $L = \{L_a\}_{a=1}^{d}$, where one set of elementary layers, $L_a$, is identified for each aspect, $a$.

Kivelä et al. use the cartesian product $L_1 \times ... \times L_d$ to construct each layer in a multilayer network by building a set of all of the linear combinations of elementary layers. To allow for vertices to be absent in certain layers, they introduce $V_M \subseteq V \times L_1 \times ... \times L_d$. They add that two vertices are described as adjacent if they are connected to each other in the same layer. However, two vertices are described as incident to each other if the vertices are connected across different layers. Kivelä et al. provide the following notation to identify the layer of the source vertex and the terminal vertex of an edge relationship: the set of edges as $E_M$, where $E_M \subseteq V_M \times V_M$. Kivelä et al. use the preceding notation to define a multilayer network, $M$ as:

**Definition 2.1.3.** Multilayer Network

$$M = (V_M, E_M, V, L), \tag{2.1}$$

where $V$ is the total number of vertices in the Network, $L = \{L_a\}_{a=1}^{d}$ for elementary layers $L_a$ for each aspect $a$, $V_M \subseteq V \times L_1 \times ... \times L_d$, and $E_M \subseteq V_M \times V_M$ [4].

Figure 2.1 depicts an example of a multilayer network using Human HIV genetic interaction. Differentiating between these types of connections allows Kivelä et al. to define connections

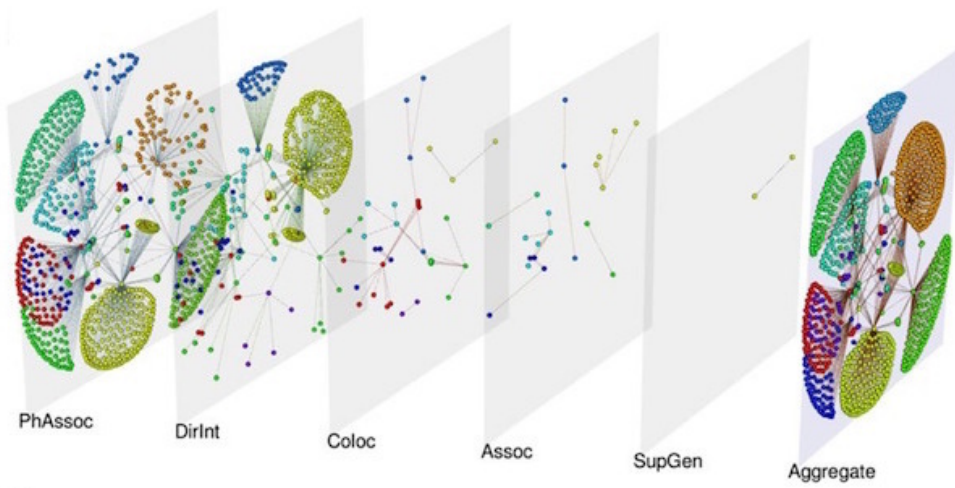



Figure 2.1: Human HIV-1 genetic interaction network. Adapted from [7].

between vertices within layers and between layers. If there is only one aspect type and the set of vertices considered in each layer are identical, then we can further classify the multilayered network as a multiplex. Kivelä et al. [4] state that a multiplex network is:

**Definition 2.1.4.** Multiplex Network

a sequence of graphs such that

$$\{G_\alpha\}_{\alpha=1}^{b} = \{(V_\alpha, E_\alpha)\}_{\alpha=1}^{b}, \tag{2.2}$$

where $E_\alpha \subset V_\alpha \times V_\alpha$ is the set of edges and $\alpha$ indexes the graphs.

Alternatively, Kivelä et al. [4] describe multiplex networks using the term*edge-colored multigraphs*, $G_e$. They define edge-colored multigraphs as:

$$G_e = (V, E, C), \tag{2.3}$$

where $V$ is the vertex set; $C$ is the color set, which is used for labelling the type of edge; and $E \subset V \times V \times C$ is the edge set.

Figure 2.2 depicts an example multiplex network using social interactions on Twitter and representing people as vertices, $V$. Connections between any two people, $E$, are plotted in three separate graphs corresponding to retweeting, replying, and mentioning. Kivelä et al. would consider these graphs as elementary layers that belong to the same aspect. However, if temporal data was collected for when each action of retweeting, replying, and mentioning occurred then we could build an additional layer of data as a separate aspect of the network. Gray lines in Figure 2.2 depict vertices of one layer incident to vertices of another layer. For a deeper explanation of this network see the paper written by Domenico et al. [8], *The Anatomy of a Scientific Rumor*.

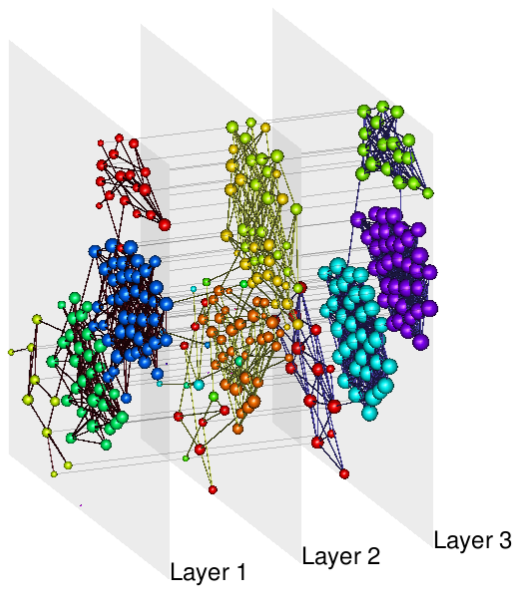



Figure 2.2: HIGGS multiplex social interaction Twitter data. Source: [8].

When all the layers are collapsed into a single network with parallel edges or single edges with weights, we can further classify the network as monoplex. Kivelä et al. [4] define a monoplex network, $O$, as:

**Definition 2.1.5.** Monoplex Network

> the aggregation of all of the layers of a multiplex network into a single weighted layer. Aggregation is achieved by defining edge weights, $m$, between vertices in each layer and expressing the final weight as a linear combination of $m$ from each layer.

When there is no order in layer importance, Kivelä et al. propose a default uniform distribution of weights where $m = 1$ for each layer. We incorporate this default weight concept into our methodology development in the next chapter. Figure 2.3 depicts a multiplex network example, which represents airports as vertices, $V$; direct connections between airports as, $E$; and airline names as colors, $C$, for each layer. The far right layer labelled *Aggregate* in Figure 2.3 is the resulting monoplex network after aggregating all of the layers of the multiplex European Airport Network. The previous definitions have explained our complex systems as networks and categorized them according to the types of information they display. In our next set of definitions, Newman [5] describes some of the vocabulary used to analyze network topology. He defines the topology of the network as:

**Definition 2.1.6.** Network Topology

> the physical or logical arrangement and structure of the network.

Network topology can be described using a variety of quantities and measures of features within a network. Some of these measures include centrality, components, diameter, density, average path length, and clustering coefficient. The preceding list of network topological characteristics is not intended to be exhaustive. However, the topological characteristics defined in Section 2.1.1 provide the reader with enough background to understand their application within the context of this thesis.

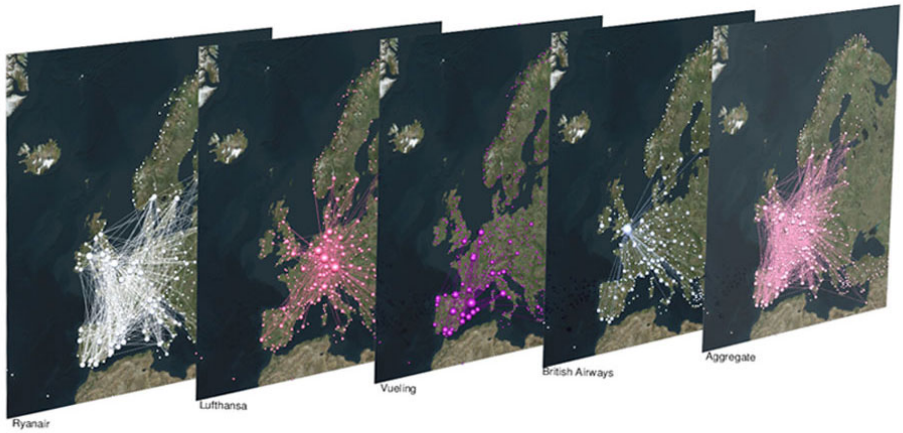



Figure 2.3: Multiplex European Airport Network. Adapted from [7].

### 2.1.1 Topological Characteristics

Centrality refers to how influential or important a vertex is within the scope of the network. The influence of a vertex can be described locally amongst its neighbors or globally within the context of the entire network. Some of the more popular measures of centrality include *degree*, *eigenvector*, and *betweeness centrality*. Newman [5] defines degree and Eigenvector centrality and Orman et al. [9] define betweeness centrality measures in the following manner:

**Definition 2.1.7.** Centrality

> the degree centrality, $k_i$ of vertex $i$, measures the involvement of a vertex in a network by the number of vertices connected to it. Eigenvector centrality calculates a degree centrality score proportional to the sum of the degree centrality scores of its neighbors. Betweeness centrality asserts the ability of a vertex to play a 'broker' role in the network by measuring how well it lies on the shortest paths connecting other vertices.

For our purposes, betweeness centrality is particularly interesting since it involves the connections between vertices on the shortest paths. Many network disruption techniques involve some variation on increasing the lengths of the shortest paths. This makes vertices with high betweeness centrality excellent targets. We attempt to apply this reasoning to measure community centrality in Chapter 5 to build community targeting profiles. In addition to betweeness centrality, another useful metric that involves shortest paths is the average path length. Newman [5] defines the average path length as:

**Definition 2.1.8.** Average Path Length

> the mean geodesic or shortest-path distance between pairs of vertices.

Average path length is a global measure that determines on average, the fewest number of edges required to traverse between any two vertices. A small average path length number implies there exists multiple redundancies in paths to connect vertices. This is typically the case unless the network is not completely connected. It is often useful to describe a graph by the number of independently connected groups of vertices or components. If a path exists between every vertex in the graph to all other vertices in the graph, then the graph is referred to as a connected graph and has only one component. Newman [5] defines the component in an undirected network as:

**Definition 2.1.9.** Components

> a maximal subset of vertices such that each is reachable by some path from each of the others.

In the context of community detection, small components of graphs typically form their own communities since they have no outward connections to the rest of the graph. Graphs with one component are desirable for many analytical algorithms that rely upon high connectivity and the ability to detect the shortest path in the network. Another characteristic that assists in describing the relative size of a given network is the network diameter.

According to Newman [5], the diameter is:

**Definition 2.1.10.** Diameter

> the length of the longest finite geodesic path anywhere in the network.

We can visualize this metric by thinking of the network as a road map where each edge represents a length of road between cities or vertices. The diameter is essentially the maximum of the lengths of all the shortest paths between vertices in the network without repeating edges. Another metric that involves connections between vertices is density. Newman [5] refers to network density as:

**Definition 2.1.11.** Density

> the fraction of edges that are actually present, out of the total number of possible edges.

If every vertex in the graph is connected to all of the other vertices then it is referred to as a clique with a density value of one. Density is also highly correlated to the clustering coefficient. Clustering coefficient is the probability that vertices in the graph cluster together. According to Newman [5], the clustering coefficient, $C$:

**Definition 2.1.12.** Clustering Coefficient

> measures the average probability that two neighbors of a vertex are themselves neighbors:
>
> $$C = \frac{\text{(number of triangles)} \times 3}{\text{number of connected triples}}, \tag{2.4}$$
>
> where connected triples means three vertices $uvw$ with edges $(u, v)$ and $(v, w)$.

In a social context, this is analogous to the probability of friendship transitivity. This metric determines the likelihood that person $a$ is friends with person $c$ given that person $a$ is friends with person $b$ and person $b$ is friends with person $c$. Now that we have explored some definitions related describing networks, in the next sections we focus on the research conducted on community detection.

## 2.2 General Community Detection

Fortunato et al. [10] reveal that one of the difficulties of community detection is that a detailed and comprehensive definition of community does not currently exist in network science. Many authors, including Kivelä et al. [4], agree that a universal definition potentially constrains the creativity and applicability of the development of community detection algorithms. For the purposes of this thesis, the community definition developed by Radicchi et al. [2] is used as a foundation. Their definition of community covers the general concept without being too specific as to hinder the development of our methodology. Radicchi et al. define communities as:

**Definition 2.2.1.** Communities

> a subset of vertices within the graph such that connections between vertices within the community are denser than connections with the rest of the network.

Radicchi et al. further classify communities as either weak or strong based on degree counts. The subgraph, $V$, is a *community in a weak sense* if the sum of all of the degrees within $V$ is greater than the sum of the degrees towards vertices outside of $V$, where $k_i^{in}(V)$ and $k_i^{out}(V)$ represent the degrees of the verticies inside and outside of the community respectively. They symbolically represents this relationship as:

$$\sum_{i \in V} k_i^{in}(V) > \sum_{i \in V} k_i^{out}(V). \tag{2.5}$$

Radicchi et al. explain that the subgraph $V$ is only considered a *community in a strong sense* if each vertex has more connections, $k_i$, inside the community, $k_i^{in}(V)$, than the vertex has with the remainder of the network outside of the community, $k_i^{out}(V)$.

$$k_i^{in}(V) > k_i^{out}(V), \forall i \in V. \tag{2.6}$$

In graph theory this concept of strong and weak communities has been previously introduced by Eroh et al. [11] as an *alliance*. This concept provides a rough metric for understanding the value of the communities that form as a result of our methodology.

Du et al. [12] point out that algorithms are usually compared based on computational complexity and accuracy. They refer to computational complexity as the time it takes an algorithm to perform all of its mathematical operations as a function of an input size, $n$. Du et al. further explain that this concept is known in the scientific community as big $O$ notation. While the computational complexity of community detection algorithms is relatively easy to measure, accuracy is much more difficult to quantify.

The known community structure is often referred to as the *ground truth*. Ideally, the ground truth can be used to compare and validate the resultant communities from the detection algorithm. This has not been an effective method as ground truth communities are hard to detect just from the topology of the network. Another challenge is establishing a ground truth for comparison. In many cases the ground truth is not known, which makes it difficult to determine if the algorithm successfully partitioned the network into appropriate communities.

If the ground truth is known, Orman et al. [13] suggest Normalized Mutual Information (NMI) as an algorithm performance measure. According to Ana et al. [14], this metric compares the degree of similarity between two different partitions, $P^a$ and $P^b$, of the same set of data. Ana et al. define NMI by Equation 2.7 as:

$$NMI(P^a, P^b) = \frac{-2\sum_{i=1}^{k_a}\sum_{j=1}^{k_b} n_{i,j}^{ab}\log(\frac{n_{i,j}^{ab}\cdot n}{n_i^a\cdot n_j^b})}{\sum_{i=1}^{k_a} n_i^a\cdot\log(\frac{n_i^a}{n}) + \sum_{j=1}^{k_b} n_j^b\cdot\log(\frac{n_j^b}{n})}, \tag{2.7}$$

where $n_{i,j}$ counts the false positives: the vertices identified by the algorithm to be community $i$ when in reality, the vertices belong to community $j$.

Orman et al. further explain that NMI values range from 0 to 1, with 1 meaning the algorithm matches the ground truth. Jeub et al. [15] argue that in many cases, the ground truth is not known and thus cannot be used as a comparison metric. Under these circumstances Jeub et al. recommend measuring the conductance of the communities to develop the Network Community Profile (NCP). Jeub et al. reveal that the purpose of the NCP is to establish a pairing criteria for matching a given network with an appropriate community detection algorithm.

To define conductance, Jeub et al. introduce the following terminology. They begin by defining a graph, $G$, as $G = (V, E, w)$, where $G$ has a weighted adjacency matrix $A$. Jeub et al. add that the volume (vol) between two given sets of vertices $S_1$ and $S_2$ is equal to the total weight of edges that connect $S_1$ and $S_2$. They [15] represents this relationship as:

$$vol(S_1, S_2) = \sum_{i \in S_1} \sum_{j \in S_2} A_{i,j}. \tag{2.8}$$

Jeub et al. use the idea of volume to further develop the concept of conductance for a set of vertices, $S$, as a ratio of the surface area of the hypothesized community to the volume of the community. They define the surface area of the community as the volume between the vertices that belong to the community denoted by the set, $S$, and the vertices that do not belong to the community in the set, $\overline{S}$. Under this construct, they [15] define conductance, $\phi$, of a set, $S$, as:

$$\phi(S) = \frac{vol(S, \overline{S})}{min(vol(S), vol(\overline{S}))}. \tag{2.9}$$

Applying Equation 2.9, Jeub et al. conclude that the conductance of $G$ is equivalent to the minimum conductance of any subset of vertices.

$$\phi(G) = \min_{S \subset V} \phi(S). \tag{2.10}$$

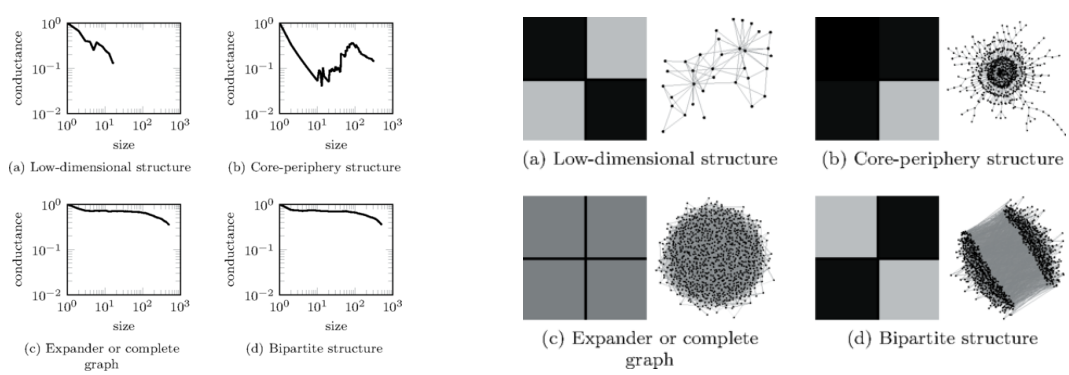
Jeub et al. explain that conductance values range from 0 to 1, with smaller values corresponding to better quality communities. Unfortunately, calculating conductance is considered an Non-deterministic Polynomial-time (NP) hard problem. Essentially this means that for large networks, the operational complexity is too large for computers to calculate in real time. The dark networks used in this thesis are small enough that conductance can be calculated directly using Equation 2.10. However, Jeub et al. offer a solution for calculating the conductance of larger networks. Fortunately, Chung [16] has successfully approximated $\phi(G)$ using the second smallest eigenvalue, $\lambda_2$ of the normalized Laplacian. Building upon

conductance, Leskovec et al. [17] developed the concept of the NCP to produce a community quality score of the best community of a given size, $k$, as a function of the community size, $k$:

$$\phi_k(G) = \min_{S \subset V, |S|=k} \phi(S). \tag{2.11}$$

Plotting NCP values results in three meaningful behavioral trends regarding the best community size [15]. These behaviors can be identified in the following figure from Jeub et al., Figure 2.4.

1. Increasing slope: small communities are optimal Networks (b) Figure 2.4
2. Horizontal line: community quality is independent of $k$ networks (c-d) Figure 2.4
3. Decreasing slope: large communities are optimal Network (a-b) Figure 2.4




Network Community Profiles

Adjacency Block Models

Figure 2.4: Block Models and Network Community Profiles: (a) Zachary Karate Club (b) Core-periphery structure example (c) Erdős-Rényi graph. (d) bipartite block model example. Adapted from [15].

According to Kivelä et al. [4], some of the more popular community detection algorithms are centered around the modularity function. Newman [18] uses the concepts of modularity and spectral graph properties to partition the network into modules or communities. He defines network modularity as:

**Definition 2.2.2.** Network Modularity

> the difference between the actual number of edges in a partitioned group and the expected number of edges in a partitioned group for a similar network with the same number of vertices, where the edges are randomly generated [18].

In the following example, Newman explains the concept of network modularity involving a graph that is divided into two partitions. Given a graph $G$ that has $n$ vertices, $G$ can be partitioned into two groups, $G_1$ and $G_2$, where $s_i = 1$ if vertex $i \in G_1$ and $s_i = -1$ if vertex $i \in G_2$.

In order to explain this algorithm, Newman first defines the number of edges between $i$ and $j$ as $a_{i,j}$. The $a_{i,j}$ values represent entries in the adjacency matrix $A$. If the degree of $i$ and $j$ is $k_i$ and $k_j$, respectively, and the total number of edges in $G$ is $m = \frac{1}{2}\sum_i k_i$, then the expected random number of edges between $i$ and $j$ can be expressed as $\frac{k_i k_j}{2m}$. The modularity, $Q$, is the result of summing $a_{ij} - \frac{k_i k_j}{2m}$ for all pairs $(i, j)$ in the same group.

After several observations and manipulation, Newman expresses modularity, $Q$, in matrix form as:

**Definition 2.2.3.** Modularity Matrix

$$Q = \frac{1}{4m}\vec{s}^T B_{\vec{s}}, \tag{2.12}$$

> where the modularity matrix represents $\vec{s}^T$ as the transpose of the column vector with the group membership entries $\vec{s}_i$, and $B$ is a real symmetric matrix with elements $b_{i,j} = a_{i,j} - \frac{k_i k_j}{2m}$.

Newman mathematically extended the modularity concept and expressed it using the spectral properties of the graph. He defines the eigenvalues of $B$ as $\beta_i$ and the corresponding

eigenvectors of $B$ as $u_i$. And using them, he represents the membership vector $\vec{s}$ as a linear combination of $u_i$: $\vec{s} = \sum_{i=1}^{n} a_i u_i$ with $a_i = u_i^T \cdot \vec{s}$ in order to rewrite modularity in the following form:

$$Q = \frac{1}{4m} \sum_i a_i u_i^T B \sum_j a_j u_j = \frac{1}{4m} \sum_{i=1}^{n} (u_i^T \cdot \vec{s})^2 \beta_i. \tag{2.13}$$

The established mathematical convention is to order the eigenvalues in non-increasing order, $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_n$. According to Chung [16], the largest eigenvalue, $\beta_1$, and its corresponding eigenvector, $u_1$, capture the eigenvector centrality of vertices in a graph. In order to maximize the modularity value, $Q$, in Equation 2.13, the dot product of $u_1^T$ and $\vec{s}$ needs to produce the largest value possible. Since the elements of $\vec{s}$ are restricted to $\pm 1$, the maximum value results when the sign of the corresponding elements of $s_i$ and $u_1$ match. A direct consequence of maximizing $Q$ is the partition of the network into two groups. Any vertex that has corresponding positive elements is assigned to $G_1$ and the remainder are assigned to $G_2$. Newman reveals that this concept can be adapted for communities with overlap, where the vertices corresponding to the zero entries are assigned to both communities.

While this method of partitioning the network using Definition 2.2.2 is only described for two communities, the idea can be extended to form further partitions. Blondel et al. [19] assert that by examining $G_1$ and $G_2$ independently as sub-graphs and applying the modularity method using Newman's Equation 2.13, it is possible to further subdivide the network in a hierarchical fashion until there are no more positive eigenvalues in the modularity matrix. Under this procedure, a non-positive eigenvalue corresponds to an eigenvector filled only with 1's, which results in all vertices belonging to the same community.

Newman [20] describes a methodology that incorporates network modularity to partition the network into communities referred to as the *fast greedy* algorithm. This algorithm optimizes modularity by first assuming every vertex is its own community and begins merging communities at each step until all of the communities have been merged into a single large community. Orman et al. [13] explain that the largest increase or smallest

decrease in modularity is recorded and compared at each step to determine the final best partition of the network into communities.

Barabási [21] explains that modularity can be used as a metric for determining the optimal partition of the network into communities. The closer the modularity value is to the value one, the more optimal the partition. He demonstrates this concept in Figure 2.5 using a small graph with nine vertices and 13 edges. Part *a* of Figure 2.5 depicts the optimal partition of the graph into two communities with a modularity value of 0.41. This value is considered optimal since there is no other arrangement of the vertices into communities that creates a higher modularity value. Part *b* of Figure 2.5 demonstrates another partition of the graph into communities that results in a smaller modularity value of 0.22 and is thus labeled suboptimal. In part *c* of Figure 2.5 we can observe the effects of grouping all of the vertices into a single community, which results in a modularity value of 0. Conversely, part *d* of Figure 2.5 displays the case where all vertices belong to their own community, which results in a negative modularity value of −0.12.

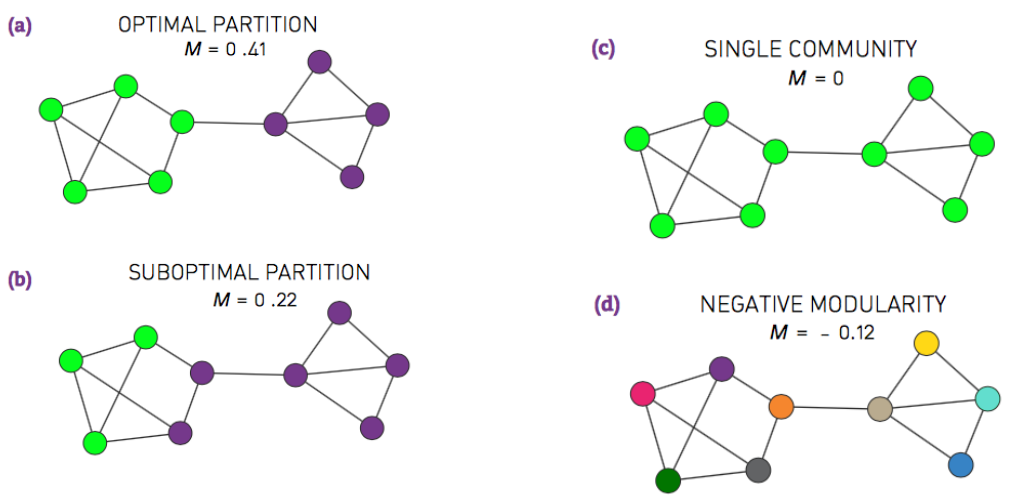



Figure 2.5: Understanding modularity using different community partitions of a network: (a) Optimal $M = 0.41$ (b) Suboptimal $M = 0.22$ (c) One Community $M = 0$ (d) Negative $M = -0.12$. Adapted from [21].

Orman et al. [22] cautions against using modularity as a community quality metric. They argue that the modularity value is highly dependant on the size of the community. Fortunato

et al. [23] support this assessment and state that for large networks, there is a limit imposed by modularity for a detecting communities of a certain size. For example, a large scale network with a high degree of interconnectedness between communities results in poor modularity values for the network. This is problematic when the network structure ground truth is partitioned into many small communities. This also makes it difficult to compare modularity values between networks with different numbers and sizes of communities. To combat this, Everton [24] believes a normalization metric needs to be considered.

Everton expands upon the idea of using modularity as a community quality metric by advocating a similar metric, cluster adequacy. Cluster adequacy, $Q'$, normalizes graph modularity, $Q$, by dividing the measured $Q$ by the best possible $Q$ for a given number of communities, $m$. The best possible $Q$ is determined as a function of the number of communities. According to Siems [25], UCINET [26] defines cluster adequacy, $Q'$, as:

$$Q' = \frac{Q}{1 - \frac{1}{m}}. \tag{2.14}$$

Looking purely at the measured $Q$, it is possible to mistakenly conclude that the communities are mediocre quality. However, by comparing the measured value of $Q$ to the best possible modularity for a given number of communities, cluster adequacy reveals that the community quality is much higher. For example, in a graph of 100 vertices, the most ideal $Q$ for two communities is 0.5. Thus if the measured $Q$ is also 0.5 then $Q'$ is computed to be 1. The measured $Q = 0.5$ may seem mediocre, but when compared to the best possible $Q$, $Q' = 1$ reveals this modularity and consequently community quality square to be ideal.

Cluster adequacy allows us to compare the quality of communities in different graphs by normalizing the value relative to the number of communities in each graph. However, cluster adequacy continues to introduce bias into the community quality measure. Cluster adequacy favors a uniform distribution of vertices into equal sized communities, which is rarely possible in real networks. Orman et al. [22] argue that similar to a degree distribution, community size tends to follow a power-law distribution as well. However, Orman et al. concede that some real networks deviate from this trend. Although this metric contains flaws, it's basis in modularity is particularly helpful in comparing communities that are the result of modularity optimization based algorithms.

Orman et al. [13] describe another modularity based algorithm known as the *Louvain* method, which is an extension of the fast greedy algorithm using a two phased approach. During the first phase, communities are initially identified using the fast greedy optimization process. The second phase constructs an entirely new network by replacing all of the vertices that belong to each community with a single vertex. The multiplicity of edges between the newly formed vertices is preserved. The fast greedy is then applied to this new network until all of the communities have been aggregated into a single community. The communities in the original network are then created by collecting all of the original vertices that were identified into the vertex in the fast greedy algorithm. Barabási [21] visually describes two iterations of the the Louvain method using Figure 2.6.

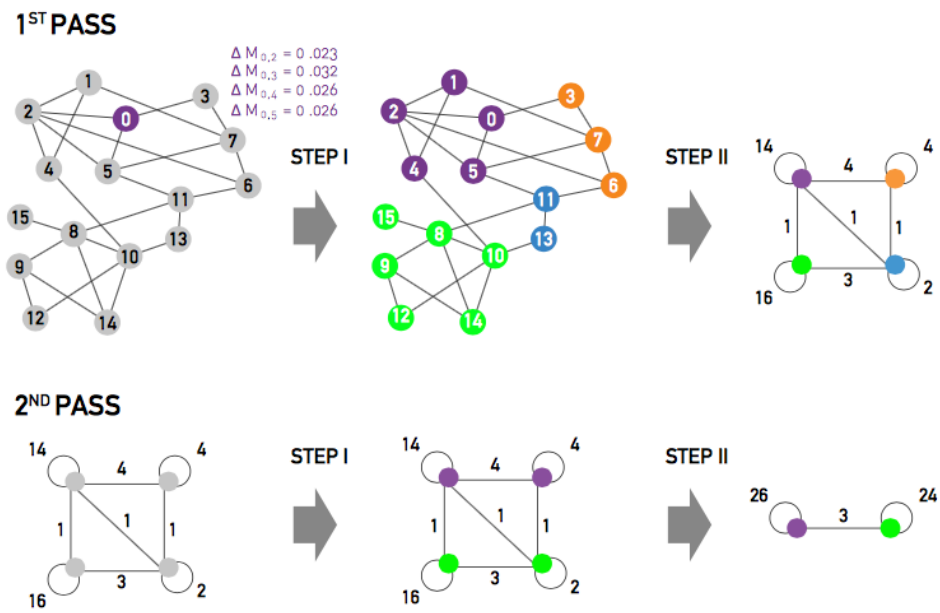



Figure 2.6: Louvain method using fast greedy at Step 1 and collapsing communities at Step 2 for a total of two iterations. Source: [21].

According to Newman [5], the early implementations of modularity based algorithms resulted in an operational complexity of at best $O(n^2)$. However, Clauset et al. [27] determined that if the fast greedy algorithm is applied to a sparse network, the algorithm can be modified to a reduced complexity of $O(n \log^2 n)$. Newman [5] describes additional algorithms such as the *power method* that seek to improve the speed of modularity based algorithms by calculating eigenvector centrality without wasting computational efforts on

calculating the remainder of the eigenvalues of $B$. According to Blondel et al. [19], the Louvain method's operational complexity can be approximated as $O(n \log n)$. How does this operational complexity translate for large networks? Blondel et al. explain that if the Louvain method is applied to a large network with 2 million vertices, the algorithm will finish in approximately 2 minutes. Barabási [21] claims Louvain can actually be implemented even faster with a complexity of $O(m)$. This means that the algorithm speed is linearly proportional to the number of edges in the network. This is a tremendous advantage when applying the algorithm to large scale networks. Although the networks we consider in this paper are relatively small we would like to design a methodology that works for any size networks. Louvain's desirable operational complexity enhances its credibility as an efficient single layer community detection algorithm.

The relative speed of modularity based algorithms enable them to be applied to very large networks. Additionally, Newman [18] advocates modularity based algorithms for community detection because it does not require the user to input the size of the communities. He further explains that this method does not preclude the possibility that there may only be one community. This revelation means that not every network should to be partitioned into communities. Newman reminds us that the absence of a partition is also useful in describing the topological characteristics of the network. Another feature of this algorithm is that every vertex is assigned to exactly one community. Barabási [21] argues that this features limits the potential of the modularity algorithm to produce the best possible partition of the network into communities. He supports this claim by demonstrating that a vertex with a high degree and high clustering coefficient will naturally have a lower modularity based on the definition of modularity. Consequently, these low modularity vertices will most likely decrease the modularity of any communities that seek to incorporate them during the fast and greedy portion of the algorithm. The area of research known as fuzzy community detection also considers this to be a limitation.

According to Zhang et al. [28], fuzzy community detection allows each vertex to be simultaneously assigned membership to multiple communities. Zhang et al. propose an iterative approach to propagating community membership degrees of all vertices in the network. They incorporate the use of topological characteristics as a selection criteria for identifying the starting vertex. New communities emerge under this construct from adjacent vertices to the start vertex, and start vertices are updated at each iteration based on modularity

performance. The algorithm developed by Zhang et al. claims to be highly flexible between performance and computational complexity. Our research in this paper focuses on non-overlapping communities. However, the framework of our methodology is potentially applicable to overlapping communities as well, as we allow the user the freedom to choose the community detection algorithm for each layer.

## 2.3 Multilayer Community Detection

As a single layer of information might not provide enough information for community detection, researchers have proposed various methods to detect community structure in multilayer networks. Multilayer networks have a lot of information that can be helpful in getting more modulated structure. Conversely, researchers such as Taylor et al. [29] believe too much information can actually bring noise to the network while trying to identify communities. Consequently, there must be a threshold of information that provides enough and not too much information for community detection. However, this threshold may not be the same for all types of networks, and we will address it in this research.

There are mainly two types of approaches to detect communities in multilayer networks: Layer Aggregation Approach, and Non-Aggregation Approach. In layer aggregation approaches, all layers are merged into a single network to detect community structure. However, these approaches are limited to multiplex networks (because of vertex repetition) and typically lose some information during the merging process. As a result, most researchers focus on non-aggregation approaches.

### 2.3.1 Aggregation Methods

According to Kivelä et al. [4], one of the first pseudo-community detection methodologies implemented on social networks was a concept known as *blockmodeling*. Batagelj [30] describes blockmodeling as a general technique for partitioning the vertices of the network into groups based on an identified common pattern. This methodology is technically not community detection because it does not rely specifically on the density of connections as described in Definition 2.2.1. Prescott et al. [31] successfully applied this process to biochemical systems by building a multilayer network from a monoplex network.

One of the benefits of layer aggregation is the ability to exploit established community detection algorithms for single layer networks. Mucha et al. [32] used a modularity based algorithm to detect communities on individual layers or slices of a network. Their work resulted in vertices identifying with different communities depending on the layer. Tang et al. [33] used a similar approach by isolating individual layers to perform *utility integration*. Utility integration constructs a utility matrix for each layer and then calculates an optimal objective function using the summation of all of the utility matrices. Kivelä et al. [4] comment that this approach allows flexibility in the definition of utility, allowing users to choose established methods such as modularity to define utility.

The multislice approaches used by Mucha et al. and Tang et al. are beneficial for analyzing individual layers of the network, but they do not provide a satisfactory platform for comprehensive layer analysis. Slicing captures all of the details of the layers by analyzing them separately, but still requires a procedure for intersecting the analysis of these layers to produce more meaningful results together. Analysing the details of each layer within the context of the entire network is extremely challenging. According to Didier et al. [34], many of the proposals for combining the analysis of individual layers include calculating the intersection, union, or sum of the analysis of the layers. Our methodology expands upon intersection aggregation techniques.

Kivelä et al. [4] introduce another class of aggregation methods called *inverse community detection*, which makes use of the ground truth to cluster vertices into communities. Cai et al. [35] describe that this method determines an optimal linear combination of weights, $m$, that are applied during the layer aggregation process. An appropriate single layer community detection algorithm that considers weights is then applied to the aggregate network. This method is repeated, resulting in an optimal weight for each layer and consequently an ordering of the layers. Rocklin et al. [36] built upon this method by clustering multiple randomly weighted aggregate networks. They identified communities by constructing a distance matrix between pairs of different clusters. The obvious limitation of this method is that it requires knowledge of the realistically unattainable ground truth.

Taylor et al. [29] suggest it is possible to aggregate layers based on similarity to enhance the identification of community structure in the network. They observe that utility integration in the form of adjacency matrices quickly approaches a community detection limit as the

number of layers in the network increases. They observed that aggregating similar or redundant layers into a single layer enhances the performance of the utility based methods. The algorithm introduced in this thesis builds off of the notion that layers can be combined in a logical manner that maintains data integrity.

### 2.3.2 Non-Aggregation methods

Non-aggregation approaches seek to detect communities without combining the layers of the network. Many of the proposed algorithms build from the foundation of single layer detection. Howison et al. [37] build from the success of modularity-based algorithms and proposes multilayer modularity maximization as a solution for detecting communities in temporal multilayer networks. They explains that temporal multilayer networks are another special class of multilayer networks that represents each layer as a different time step of the network. Temporal layers are considered ordinal and uniform. This means layers are sequentially related and all equally weighted. Mucha et al. [32] define multilayer modularity maximization as:

$$\max_{C \in C} \sum_{s=1}^{|\tau|} \sum_{i,j=1}^{N} B_{i,j,s} \delta(C_{i_s}, C_{j_s}) + 2\omega \sum_{s=1}^{|\tau|-1} \sum_{i=1}^{N} \delta(C_{i_s}, C_{i_{s+1}}), \tag{2.15}$$

where $B_s$ is the single layer modularity matrix computed on layer $s$ defined in Equation 2.12, $B_{i,j,s}$ is the $(i,\ j)^{th}$ entry of $B_s$, $\tau$ is a sequence of adjacency matrices for each layer, $C$ is a partition of $K$ sets of vertices, $\delta(C_{i_s}, C_{j_s})$ is the Kronecker delta function for each layer, and $N$ is the number of vertices.

Didier et al. [34] also build off of Newman's modularity concept by applying it to multiplex biological networks. Didier proposes measuring the strength of the individual community structure of each layer represented as a separate graph. This first step is similar to multi-slicing approaches described in Section 2.3.1. This strength value is calculated by examining for each community the sum of the proportions of within-community edges over all the graphs minus the expectations of this sum. Using the logic that the sum of random variables is congruent to the sum of their expectations, Didier et al. [34] derived the following equation for multiplex modularity, $Q^M$, of a multiplex network, $X^{(g)}$,:

**Definition 2.3.1.** Multiplex Modularity

$$Q^M(X^{(g)}, C) = \sum_g \frac{1}{2m^g} \sum_{\substack{\{i,j\} \\ i \neq j}} (X_{i,j}^{(g)} - \frac{k_i^g k_j^g}{2m^g}) \delta_{C_i, C_j}, \tag{2.16}$$

where $m^g$ is the total number of edges of the graph $X^{(g)}$, and $k_i^g$ is the corresponding degree of vertex $i$ in the graph $X^{(g)}$.

Didier et al. [34] implemented this methodology on a 4-layer Biological network, in which each layer represents information from a different subset of genes or proteins. The multiplex modularity method was compared against aggregation approaches using an adjusted Rand index and verifying consistency with known biological processes. According to Santos et al. [38], the adjusted Rand index is used to compare similarity between two partitions. The results of the comparison by Didier et al. supports multiplex modularity as a more accurate than common aggregation methods for detecting strong communities in a biological multiplex network.

The preceding work described on how community detection – in both aggregated and non-aggregated approaches – overwhelming supports the inclusion of modularity into the design process of algorithms. This thesis recognizes modularity as an extremely powerful partitioning tool and incorporates it into our proposed methodology. For more information on community detection in general and more details on multilayer community detection approaches we recommend the paper by Kivelä et al. [4]. Now that we have explored some research on community detection, in Section 2.4 we explore some background information on dark networks to provide context for our algorithm development.

## 2.4 Dark Networks

There are many difficulties associated with mapping and analysing dark networks. Krebs [39] uses the September $11^{th}$ 2001 terrorist attack in the United States as a case study in dark network mapping and analysis. Krebs describes three challenges previously identified by Sparrow [40] that are specifically associated with mapping and analysing criminal social networks. Krebs identifies these challenges as incompleteness, fuzzy boundaries,

and dynamic behavior.

Krebs states that incompleteness is a huge factor since criminal networks do not want to be discovered [39]. As a consequence, network mapping and analysis are limited by the volume and availability of relevant and accurate data. Krebs clarifies the term fuzzy boundaries by connecting it to the process of data filtration. Knowing which relationships are important and which are not can have profound impacts on modeling and analyzing a network. The importance of implementing a data filtration process led us to incorporate filtration into our methodology in order to analyze the relationship data from the Noordin Network. Krebs points out that another key idea from fuzzy boundaries is that not every vertex needs to be included when mapping a terrorist network. This supports the idea that not all vertices are essential to dark network analysis and thus do not need to be sorted into communities. This idea will be incorporated in the development of our detection algorithm to sort some vertices into a misfit community. The misfit community essentially acts as a theoretical storage location for vertices that are not sorted into communities by our algorithm. The last challenge Krebs identifies for dark network analysis is the dynamic behavior of the network. The idea is that one finite viewing of the network results in an incomplete picture of a dark network. Sparrow [40] suggests that overlaying temporal snapshots of the network will enhance the overall understanding of the relationships between vertices in the network. This follows because the actors of the dark network particularly keep interactions at a minimum, thus temporal analysis has the potential to discover connected neighborhoods. However, obtaining temporal data on dark networks has proven to be a difficult challenge for researchers.

Krebs [39] utilizes a project team based approach to sort data on the network into four main categories. These categories are tasks, resources, strategy, and expertise links. Data that represents the meaning of one of the identified categories is placed in the same respective category. Krebs believes overt project team analysis can be applied with some modification to reveal information about covert project teams in a dark network. We apply this idea of categorizing similar relationships in the development of our methodology.

Krebs focuses his analysis on relationship data in "trusted prior contacts" based upon the research conducted by Erickson [41]. Erickson believes the densest and most meaningful under-layer of a dark network is trusted prior contacts. According to Krebs [39], this

layer is usually not visible in current snapshots of the network. These dormant layers are critical to maintaining secrecy and resilience through adaptability. For example, if one contact becomes unavailable, a new contact can be established through accessing a dormant trusted prior contact relationship. We explore these concepts further in our methodology development and dark network modeling.

Krebs [39] further subdivided his trusted prior contacts relationship data based on the strength of each relationship. He determined the strength of each relationship according to the duration and relevancy to building trust. Classmates, living together, and training met the criteria for the strongest ties. Moderate strength included traveling partners, and meetings. Dormant strength included financial transactions and occasional meetings. Visually, Krebs represented the strength of each tie by the thickness of the connecting edge between people in the graph of the network. This idea of assigning importance to layers or categories is implemented in our methodology as well.

Krebs [39] describes dark networks such as the September $11^{th}$ networks as sparse. There was a noticeably high distance between hijackers on the same team. Usama bin Laden explains the reason for sparseness in the network, stating that "Those who were trained to fly didn't know the others. One group of people did not know the other group" [39]. The main idea is that if one member of the network is caught, the remainder of the network cannot be compromised [39]. Generally this means there are very few brokers or connectors between teams. As we apply community detection, ideally these brokers will become visible as connectors between communities.

Sparseness in dark networks contributes to the idea that the definition of community should not be applied for the purpose of identifying sub-organizational groups such as teams within the network. Instead, we focused our definition of community based upon the end-user analytical goals. Sparseness also makes it difficult to establish ground truth communities for community detection algorithm accuracy comparison.

Meetings are used by dark networks to temporarily connect groups for collaboration and coordination [39]. A vertex that may have seemed inactive, suddenly becomes important after adding edges from the meetings layer of the network. Krebs explains that usually one representative from each group is sent to a meeting, which again makes it even harder to identify density based communities. His analysis revealed that the individuals selected

to attend the meeting were usually connected by a trusted prior contact relationship. The sparseness of the dark network emphasizes the need for combining data sets from multiple types of relationships in order to help increase the density of network for analytical purposes. To mitigate the impact of sparseness, we aggregate many similar types of relationships during our methodology process and develop a procedure for inferring additional ties.

Everton [24] points our that one of the main purposes of dark network analysis is network disruption. He comments that many dark network analysts take the approach of using high degrees of centrality to identify key actors. Everton contends that this tactic may not always be effective and encourages the analyst to examine more of the topographical characteristics of the network to identify more appropriate targets for disruption purposes. Krebs [39] concurs with Everton's assessment and indicates that using centrality as a metric for dark networks is likely to fail. He believes this is due to the incomplete nature of dark networks and the high sensitivity of centrality computations based on small changes to the network, especially for small networks.

The resiliency of a dark network is qualitatively described by Krebs [39] as strong due to the high redundancy of trust relationships which includes classmates, kinship, or participating in terrorist related training and operations. Krebs highlights the differences in social network and covert network analysis. The classification of relationships as strong or weak ties is entirely dependant on the type of network being analyzed. He maintains that for dark networks, trusted prior contacts is typically considered a strong tie between two vertices whereas the two vertices connected by the same nationality could be view as a weak tie. The strong tie clearly emphasises a close relationship, whereas a weak tie viewed by itself may offer only ambiguity on the relationship status. Analysis of strong ties in social networks usually produces the "cluster of network players" [39]. However, Krebs believes network players in dark networks may visibly appear to only have weak ties. Everton [42] supports Mark Granovetter's claim that an optimal combination of both weak and strong ties is ideal for dark network analysis. This claim highlights the notion that multiple layers of data must be included when analyzing the network. The incomplete and secret nature of the dark network requires weak ties to help illuminate potential hidden strong ties.

Krebs offers a strategy for disrupting terrorist networks through information aggregation and knowledge sharing. Under this strategy, the key vertices to target in the network are

vertices with unique skills and vertices that have deep rooted trust relationships with other groups. For more information on understanding dark networks and using topographical characteristics to disrupt them see Everton's book [24].

Gerdes [43] highlights another strategy that focuses on the detection of covert communities in a dark network over a five year time period. This strategy uses a hierarchical agglomeration algorithm based on finding optimal network modularity in order to detect communities. Gerde's methodology was evaluated using a True Positive Rate that labels one of the communities as covert, and calculates a performance ratio based on covert and background population members found in the community. This strategy is particularly interesting because it incorporates the temporal data recommended by Krebs in order to facilitate a more comprehensive approach to detecting a target community. Our data set includes some temporal information, but focuses on detecting multiple communities in the network.

The research highlighted in this chapter served as the foundational understanding and inspiration that enabled us to develop our methodology for detecting communities in multiplex networks. Krebs and Sparrow helped us understand the sparse nature of dark networks. Sparsity provides justification for several steps throughout our methodology. We used sparsity and the arguments provided by Taylor et al. as reasoning for aggregating similar relationships into categories and for converting cliques into communities. Taylor et al. established the need for our algorithm to be selective on our layer choices and warned us against the dangers of too many layers and redundancy. Mucha et al. and Kivela et al. confirmed our intuition on layer aggregation and described how multi-slicing and single layer community detection could be applied to a network. Didier et al. explained the problems associated with aggregating layers, which allowed us to build from his research to determine a new method for combining the multi-slicing and aggregation approaches without compromising analytical depth. Krebs and Erickson guided our category selection and ordering by establishing that trust was essential for dark networks to function. Newman provided our foundational understanding of modularity in order to compare and contrast modularity based algorithms. Our choice of single layer community detection using the modularity based Louvain method was largely guided by Blondel et al. and Barabási's performance assessment and comparison against other existing algorithms. Barabási also assisted us in understanding the weakness associated with modularity based algorithms by

placing every vertex into a community. Identifying this weakness led us to develop the concept of a misfit community. Incorporating this idea allows us to theoretically achieve higher modularity values than traditional modularity algorithms make possible. Now that we have explored the research that enabled this thesis, in the next chapter we apply our understanding of this research in greater detail. This allows us to justify and explain our algorithm to detect communities in multiplex networks.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 3:
# Data and Methodology

This chapter describes our network data sets, explains our multiplex community detection methodology, and presents our detection algorithm.

## 3.1 Data Description

Three small, real, and dark multiplex network case studies were examined in this thesis. The Noordin Top Network data we use is a subset of the original data compiled by Roberts et al. [44]. Based on the discussion in Section 3.1.1, the Noordin Network is represented by 133 vertices and 2451 edges. The Boko Haram Terrorist Network data set contains 44 vertices, and 99 edges and the FARC Terrorist Network data set contains 142 vertices and 1650 edges.

Each network data set is graphically represented as monoplex network $O$ using an analytical graphing and network visualization tool called Gephi [45]. Following each monoplex visualization is a degree distribution plot using JMP [46] statistical software. We highlighted this network characteristic to visually demonstrate some of the differences between the three dark networks. We constructed a global overview of the network presented in Table 3.1 that captures the Total Number of Vertices (V), Total Number of Edges (E), Average Degree (AD), Average Weighted Degree (AWD), Network Diameter (Di), Graph Density (De), Modularity (M), Average Clustering Coefficient (ACC), Average Path Length (APL), and Number of Partitioned Components (P) for each layer. These network topological characteristics and the algorithms used to find them are explained by Chevren's Book, *Network Graph Analysis and Visualization with Gephi* [47]. For a more detailed explanation of these topological characteristics terms, see Lewis' book, *Network science: Theory and applications* [48].

### 3.1.1 Noordin Top Network

The Noordin Top Network Data set contains the relationship information of 139 terrorists that belong to five major parent terrorist organizations operating in Indonesia [44]. The

network is named after the key broker, Noordin Top, who was known for coordinating between terrorist organizations for training and operations. This network was primarily developed from the information provided by an article published by the International Crisis Group in 2006, *Terrorism in Indonesia: Noordin's Networks* [24]. Roberts et al. [44] used this information to construct a possible total of 36 relationship types and attributes. We re-organized the relationships and attribute data into edge lists to build the layers of the Noordin Network used in this thesis.

The layers of the network are defined to be the different relationship types connecting one vertex (person) to another vertex (person). The layers of the network are used in the community definition. The attributes are the properties assigned to each vertex. The vertex attributes will be used to measure community effectiveness and resilience. As it will be discussed in Section 3.2.1, only 14 of these layers are used in this thesis. As a consequence of using 14 layers, only a subset (133) of the known 139 terrorists is examined. Figure 3.1 illustrates an overview of the Noordin Network by collapsing the 14 selected layers into a weighted aggregate monoplex network $O$. The individual layers are colored based on their corresponding category, which is explained in Section 3.2.2. The monoplex network is viewed in greater detail in Figure 3.2. Each type of relationship has an edge list. Multiple occurrences of the same edge for each layer edge list results in a thicker line representation of edge in $O$. The vertices in Figure 3.2 are colored based on degree. Higher degrees are colored blue while lower degrees transition to the smallest degree color in red. Typically the degree distribution follows the power law. However, the Noordin Network follows a more sporadic distribution as illustrated in Figure 3.3. Notice that the highest degree on the far right is Noordin Top. A summary of the associated properties of each of the separate 14 layers and the average of the layers is captured in Table 3.1.

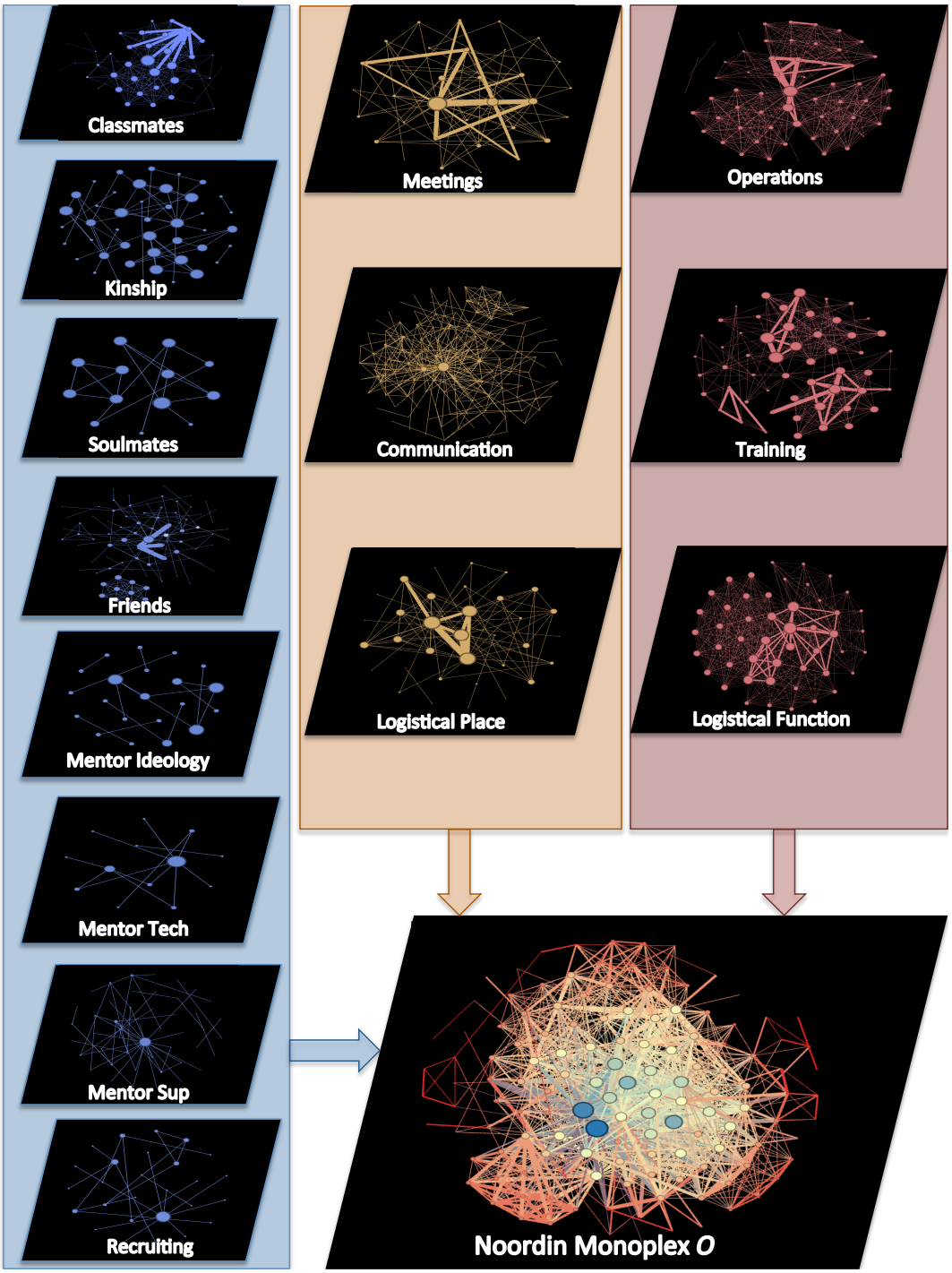



Figure 3.1: An overview of the Noordin Network. The 14 layers organized from left to right by category color with the monoplex $O$ representing the aggregation of all the layers.

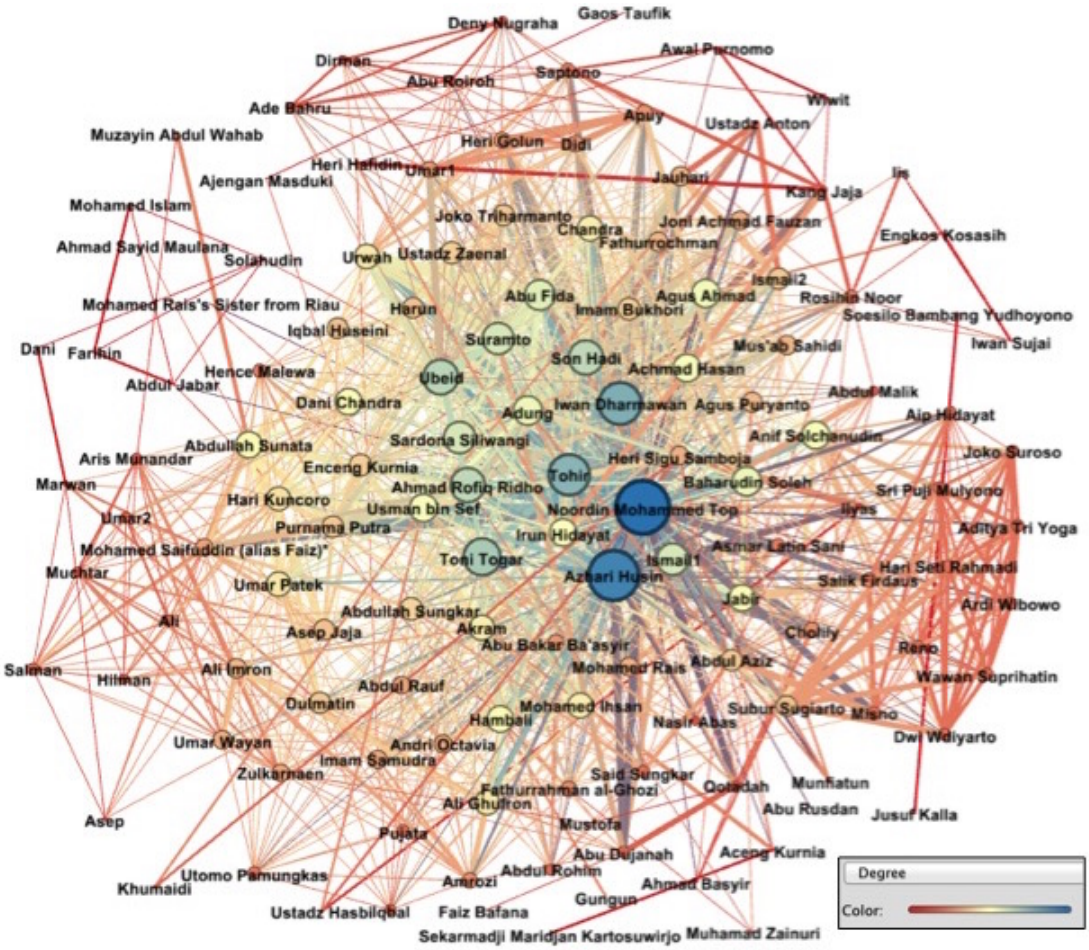



Figure 3.2: Noordin Network monoplex, $O$.

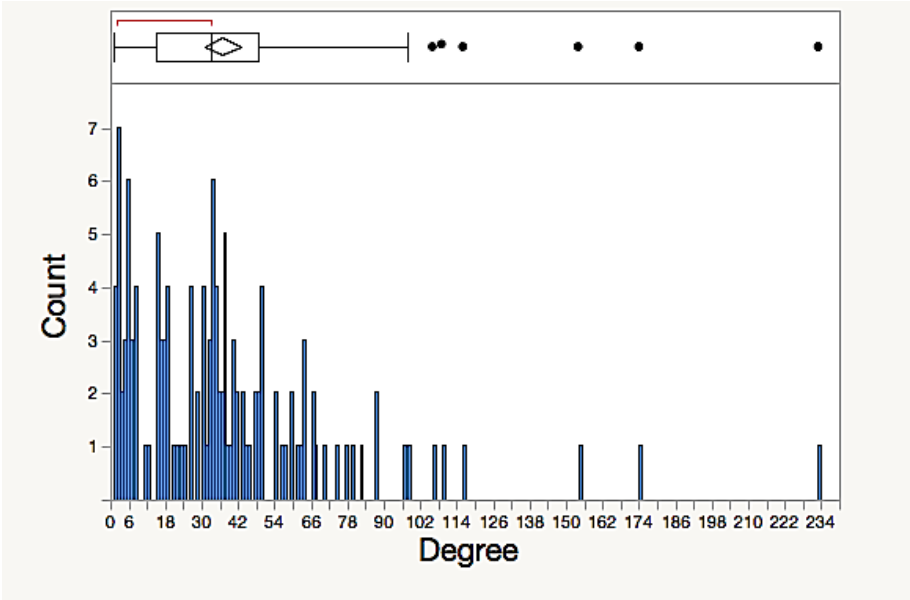



Figure 3.3: Noordin Network weighted degree distribution.

Table 3.1: Noordin Network topological characteristics by layer.

| Layer Name | V | E | AD | AWD | Di | De | M | ACC | APL | P |
|---|---|---|---|---|---|---|---|---|---|---|
| Classmates | 44 | 217 | 9.32 | 9.86 | 7 | 0.22 | 0.35 | 0.76 | 2.48 | 1 |
| Kinship | 44 | 49 | 2.23 | 2.23 | 2 | 0.05 | 0.87 | 0.95 | 1.09 | 15 |
| Soulmates | 13 | 17 | 2.62 | 2.62 | 2 | 0.22 | 0.65 | 0.89 | 1.23 | 3 |
| Friends | 83 | 158 | 3.71 | 3.81 | 9 | 0.05 | 0.71 | 0.55 | 4.01 | 3 |
| Mentor Ideological | 21 | 15 | 1.43 | 1.43 | 5 | 0.07 | 0.68 | 0.00 | 2.03 | 7 |
| Mentor Supervisory | 46 | 51 | 2.22 | 2.22 | 6 | 0.05 | 0.57 | 0.40 | 2.50 | 6 |
| Mentor Technological | 13 | 13 | 2.00 | 2.00 | 5 | 0.17 | 0.34 | 0.00 | 2.20 | 2 |
| Recruiting | 27 | 24 | 1.78 | 1.78 | 3 | 0.07 | 0.75 | 0.37 | 1.78 | 5 |
| Meetings | 33 | 110 | 5.33 | 6.67 | 4 | 0.17 | 0.33 | 0.84 | 2.16 | 1 |
| Communication | 120 | 318 | 5.30 | 5.30 | 8 | 0.05 | 0.54 | 0.53 | 3.10 | 1 |
| Logistical Place | 34 | 106 | 5.71 | 6.24 | 3 | 0.17 | 0.28 | 0.83 | 1.73 | 5 |
| Operations | 60 | 490 | 15.63 | 16.33 | 2 | 0.27 | 0.51 | 0.94 | 1.67 | 4 |
| Training | 54 | 291 | 9.74 | 10.78 | 4 | 0.18 | 0.58 | 0.89 | 2.33 | 2 |
| Logistical Function | 49 | 592 | 22.61 | 24.16 | 2 | 0.47 | 0.28 | 0.89 | 1.53 | 1 |
| Average | 46 | 175 | 6.40 | 6.82 | 4 | 0.16 | 0.53 | 0.63 | 2.13 | 4 |

### 3.1.2 The Boko Haram Terrorist Network

The Boko Haram Terrorist Network data set contains the relationship information of 44 terrorists that belong to an Islamic sect that primarily operates in northern Nigeria since 2002. According to Walker [49], the group believes the current government in Nigeria is corrupted by false Muslims. The network is extremely sparse due to its relatively young cell-like structure, and lack of collective leadership. This network data set was created by Cunningham [50] using a variety of open source documents. We re-organized the available relationship data into edge lists to build 9 separate layers for the case study on the Boko Haram Terrorist Network.

Figure 3.4 illustrates an overview of the Boko Haram Network by collapsing the 9 layers into a weighted aggregate monoplex network $O$. Figure 3.5 enhances the representation of the monoplex to include the label identification of each vertex. This network follows a power law distribution as depicted in Figure 3.6. A summary of the associated properties of each of the separate 9 layers is captured in Table 3.2.

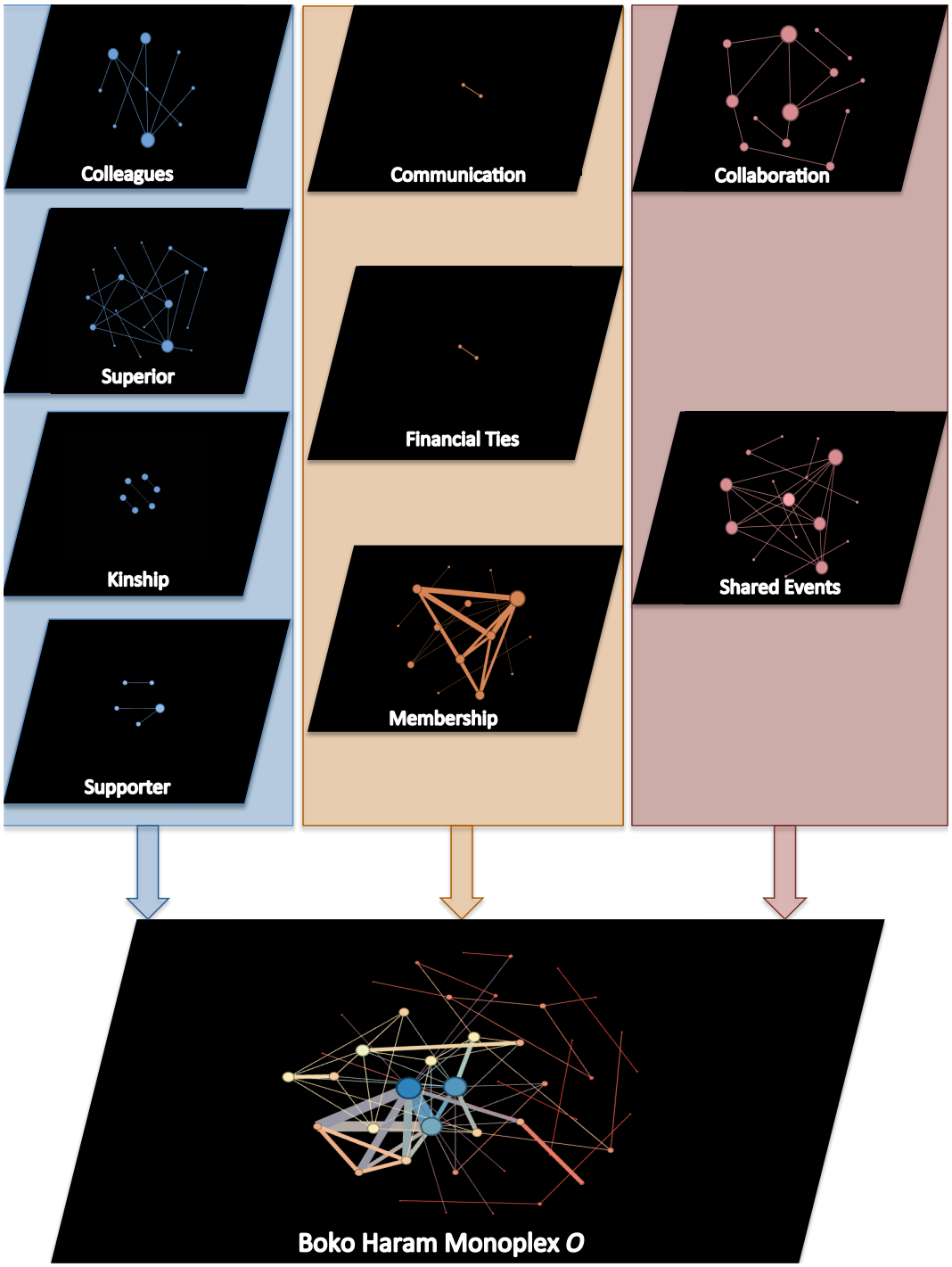



Figure 3.4: An overview of the Boko Haram Network. The 9 layers organized from left to right by category color with the monoplex $O$ representing the aggregation of all the layers.

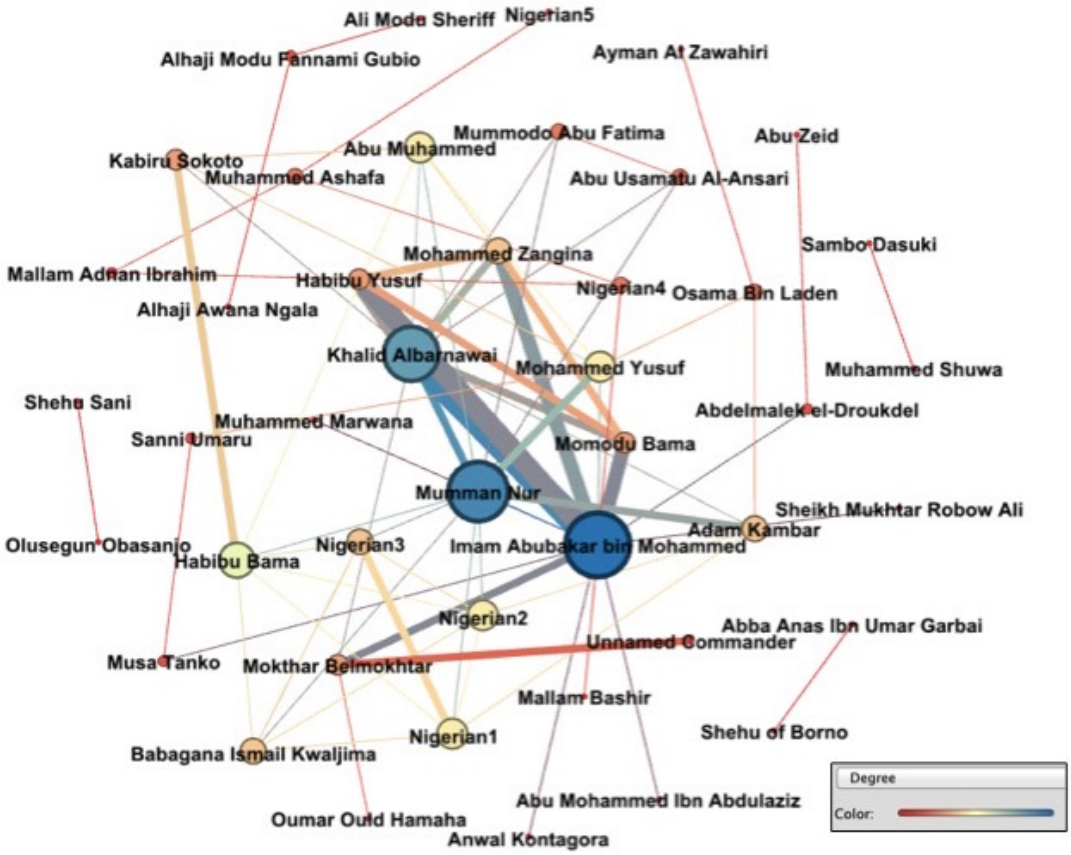



Figure 3.5: Boko Haram Network monoplex, $O$.

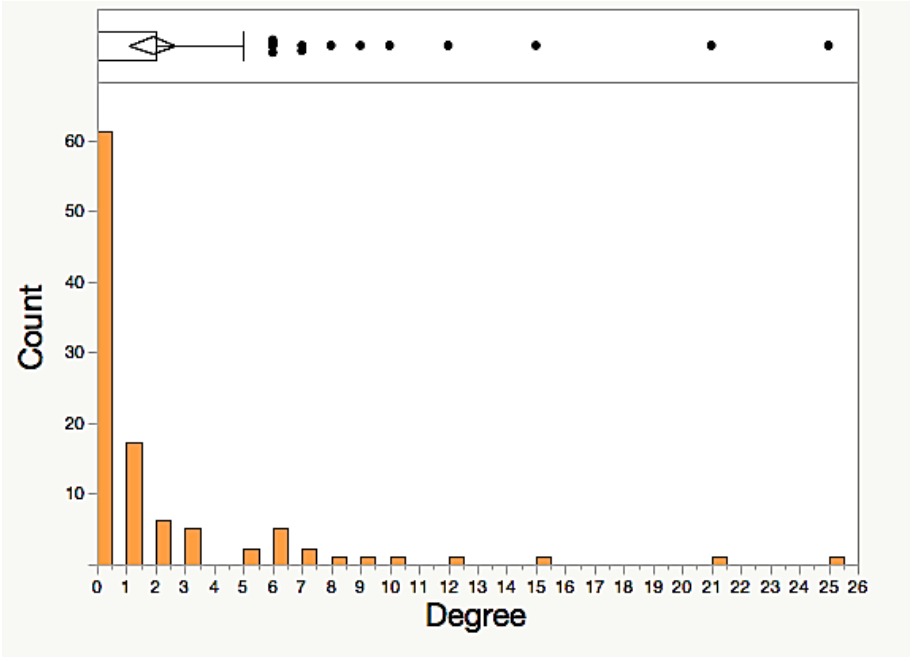



Figure 3.6: Boko Haram Network weighted degree distribution.

Table 3.2: Boko Haram Network topological characteristics by layer.

| Layer Name | V | E | AD | AWD | Di | De | M | ACC | APL | P |
|---|---|---|---|---|---|---|---|---|---|---|
| Colleagues | 9 | 8 | 1.78 | 1.78 | 4 | 0.22 | 0.41 | 0.00 | 2.33 | 1 |
| Kinship | 6 | 3 | 1.00 | 1.00 | 1 | 0.20 | 0.67 | NA | 1.00 | 3 |
| Superior | 18 | 17 | 1.89 | 1.89 | 3 | 0.11 | 0.54 | 0.18 | 1.93 | 4 |
| Supporter | 5 | 3 | 1.20 | 1.20 | 2 | 0.30 | 0.44 | 0.00 | 1.25 | 2 |
| Financial Ties | 2 | 1 | 1.00 | 1.00 | 1 | 1.00 | 0.00 | NA | 1.00 | 1 |
| Communication | 2 | 1 | 1.00 | 1.00 | 1 | 1.00 | 0.00 | NA | 1.00 | 1 |
| Membership | 14 | 32 | 2.71 | 4.57 | 2 | 0.21 | 0.30 | 0.93 | 1.34 | 4 |
| Shared Events | 16 | 21 | 2.63 | 2.63 | 2 | 0.18 | 0.40 | 0.81 | 1.22 | 5 |
| Collaboration | 13 | 13 | 2.00 | 2.00 | 7 | 0.17 | 0.47 | 0.35 | 2.84 | 2 |
| Average | 9 | 11 | 1.69 | 1.90 | 3 | 0.38 | 0.36 | 0.38 | 1.55 | 3 |

### 3.1.3 The FARC Terrorist Network

The FARC Terrorist Network data set includes the relationship information of 142 terrorists known as the Revolutionary Armed Forces of Colombia that primarily operates in Columbia and Venezuela since 1964. According to Weimann [51], the organization believes in Marxist ideology and seeks to overthrow the Colombian government. The network is sparse for most layers, but has a well-documented hierarchical structural layer due to social media [51]. This network data set was created by Cunningham et al. [52] using a variety of open source documents. We re-organized the available relationship data into edge lists to build 10 separate layers for the case study on the FARC Terrorist Network.

Figure 3.7 illustrates an overview of the FARC Network by collapsing the 10 layers into a weighted aggregate monoplex network $O$. Figure 3.8 provides an enhanced visualization of the monoplex by increasing the size of the diagram. The labels are non-existent in this figure due to the anonymity of the available data set. This network has an unusual degree distribution that is illustrated in Figure 3.9. It has 42 vertices in the middle of the distribution with relatively high degrees of 35. This is most likely due to the highly visible hierarchical leadership structure of the network depicted as yellow vertices in Figure 3.8. A summary of the associated properties of each of the separate 10 layers is captured in Table 3.3.

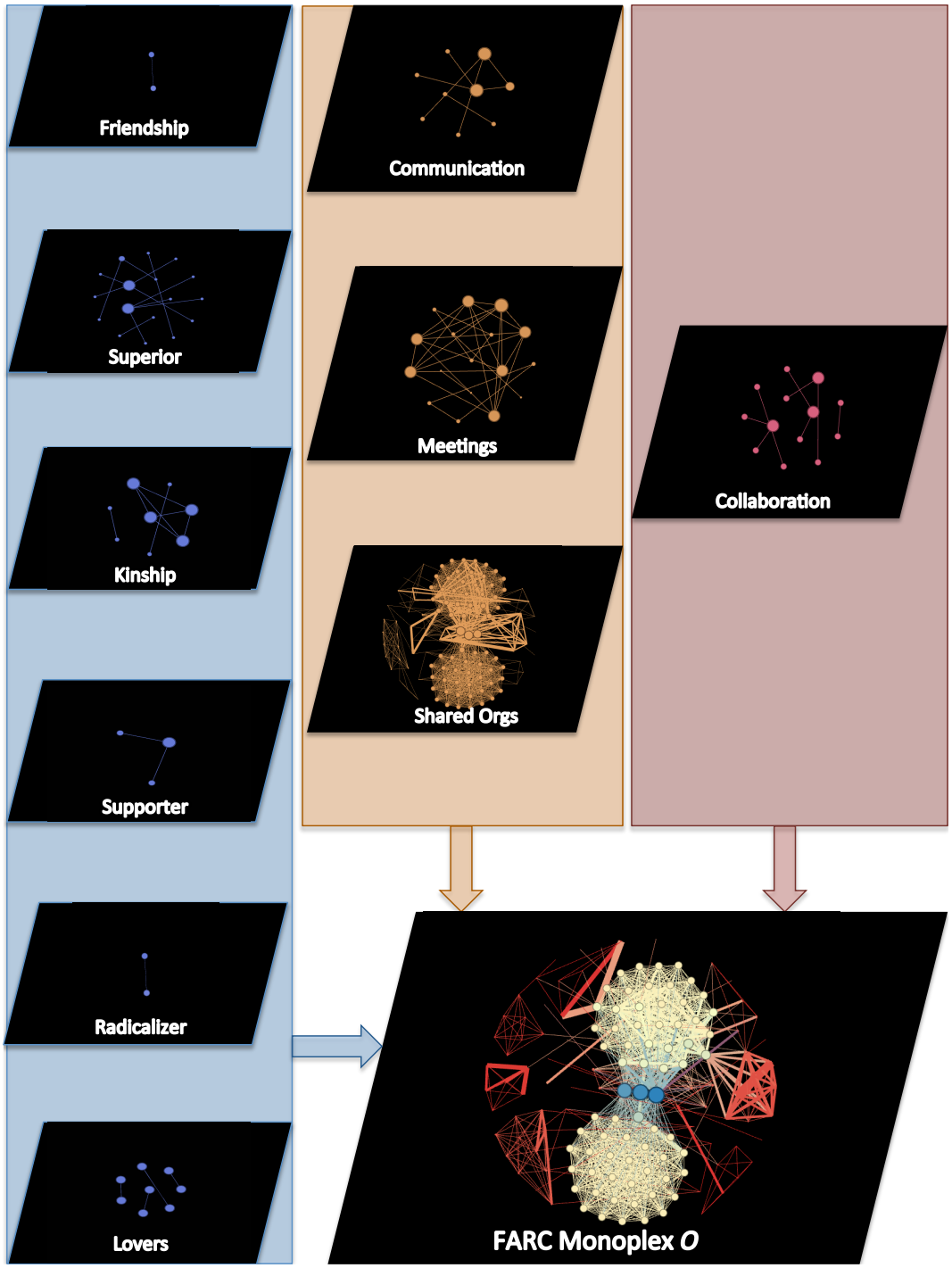



Figure 3.7: An overview of the FARC Network. The 10 layers organized from left to right by category color with the monoplex $O$ representing the aggregation of all the layers.

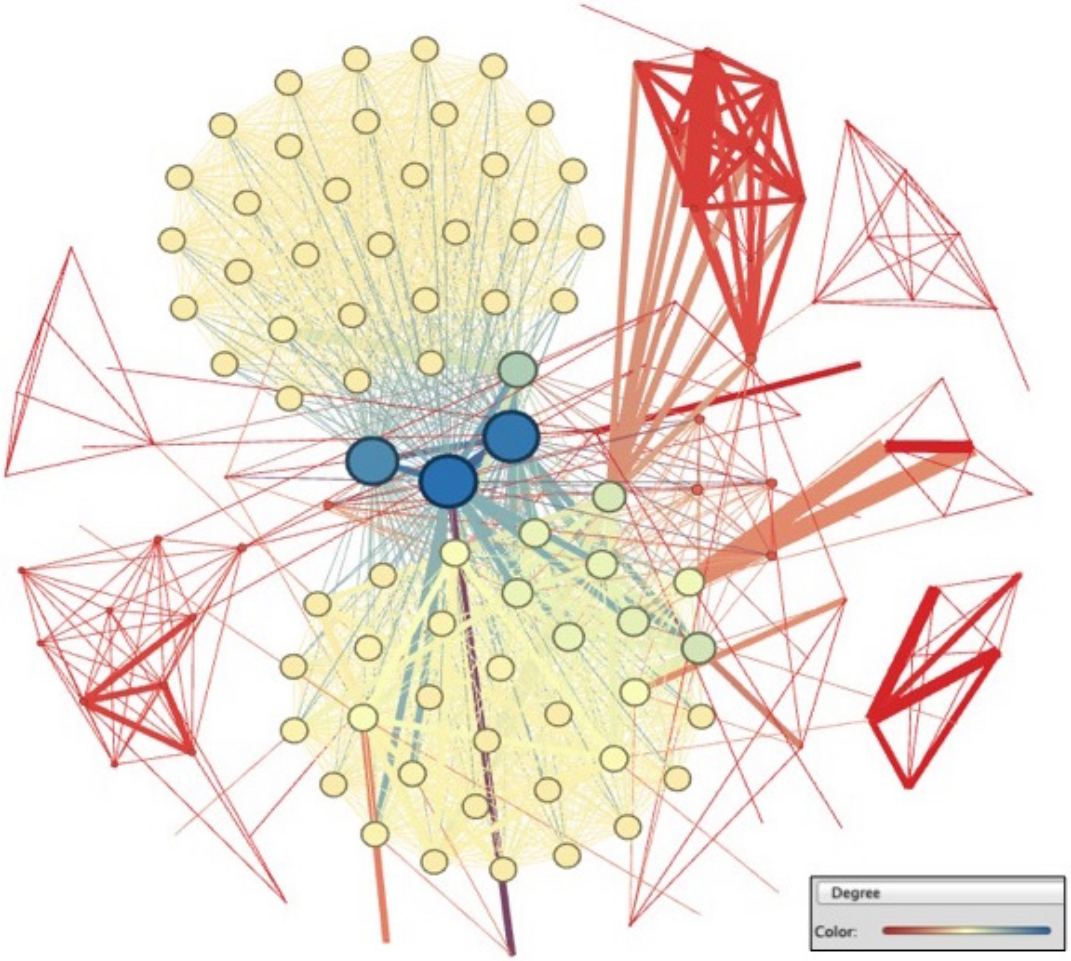



Figure 3.8: Boko Haram Network monoplex, $O$.

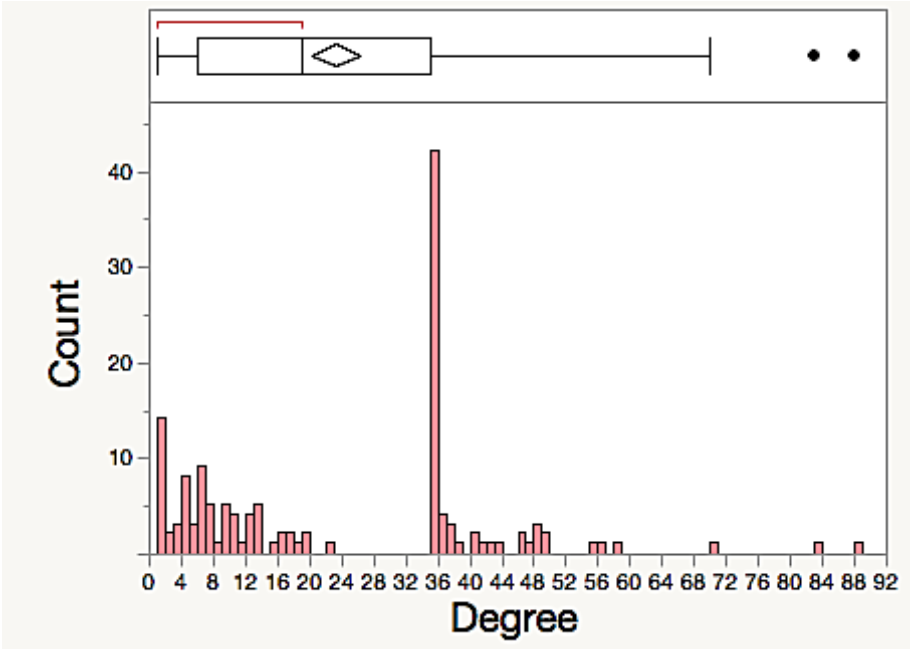



Figure 3.9: FARC Network weighted degree distribution.

Table 3.3: FARC Network topological characteristics by layer.

| Layer Name | V | E | AD | AWD | Di | De | M | ACC | APL | P |
|---|---|---|---|---|---|---|---|---|---|---|
| Friendship | 2 | 1 | 1.00 | 1.00 | 1 | 1.00 | 0.00 | NA | 1.00 | 1 |
| Kinship | 8 | 8 | 2.00 | 2.00 | 1 | 0.29 | 0.41 | 1.00 | 1.00 | 3 |
| Superior | 17 | 12 | 1.41 | 1.41 | 2 | 0.09 | 0.74 | 0.00 | 1.52 | 5 |
| Supporter | 3 | 2 | 1.33 | 1.33 | 2 | 0.67 | 0.00 | 0.00 | 1.33 | 1 |
| Lovers | 8 | 4 | 1.00 | 1.00 | 1 | 0.14 | 0.75 | NA | 1.00 | 4 |
| Radicalizer | 2 | 1 | 1.00 | 1.00 | 1 | 1.00 | 0.00 | NA | 1.00 | 1 |
| Communication | 9 | 7 | 1.56 | 1.56 | 4 | 0.19 | 0.46 | 0.00 | 2.23 | 2 |
| Meetings | 17 | 30 | 3.53 | 3.53 | 3 | 0.22 | 0.43 | 0.91 | 1.44 | 4 |
| Shared Orgs | 120 | 1577 | 24.6 | 26.3 | 4 | 0.21 | 0.50 | 0.95 | 1.87 | 5 |
| Collaboration | 13 | 8 | 1.23 | 1.23 | 2 | 0.10 | 0.78 | 0.00 | 1.27 | 5 |
| Average | 20 | 165 | 3.87 | 4.04 | 2 | .39 | .41 | .41 | 1.37 | 3 |

## 3.2 Methodology Overview

In this section we introduce the process, or algorithm, used to transform the data from a multiplex network into meaningful partitioned communities according to user-based analytical goals and objectives. Figure 3.10 provides an overview of our methodology and Algorithm 1 presents the pseudo-code for implementing this algorithm. This method takes layers of multiplex network $M$ as an input and produces threshold controlled communities as an output.

Aggregating all of the layers of $M$ creates the simple weighted graph $G$. The layers of the $M$ are sorted into weighted categories ($Category_{w_i}$). The layers of each $Category_{w_i}$ are aggregated to form a simple graph for individual category community detection. These communities in each category are converted into weighted cliques based on the assumption that edges are missing as minimal information is usually captured on terrorist networks. The aggregation of all of these cliques results in the weighted graph $W$. Choice of a threshold $\varepsilon$ results in components in $W_\varepsilon$ that create the final communities, which we then identify in $G$. The vertices in $G$ are partitioned into the recently identified communities in $W_\varepsilon$. The algorithm is designed to identify communities of order two and larger. Any vertex that is not sorted into a community is placed in the Misfit Community by default.

A user-based approach is implemented to increases flexibility of the algorithm and heighten the defined categories ability to capture the user-intended meaningful communities as a result (with a default built in). Cheever et al. [53] reveal that the user-based approach philosophy has been implemented by many solution directed companies such as Decision Lens. They further explain that Decision Lens incorporates user feedback at multiple stages during the model development process to develop a product that accurately reflects the user's goals.

Building user input into our algorithm is essential to producing meaningful communities. This understanding is critical in developing and selecting an appropriate detection algorithm. Failure to skip this contextual step in the algorithm developmental process will result in a misleading product, identifying communities that might have different reasons for clustering together.

Each step is described in detail using the following format. The step is first introduced and explained using a general network illustrated in Figure 3.10. The general case is then followed by a specific example using the Noordin Network to demonstrate some of the features of the algorithm in more detail. Figure 3.11 illustrates the application of all of the algorithm steps to the Noordin Network.

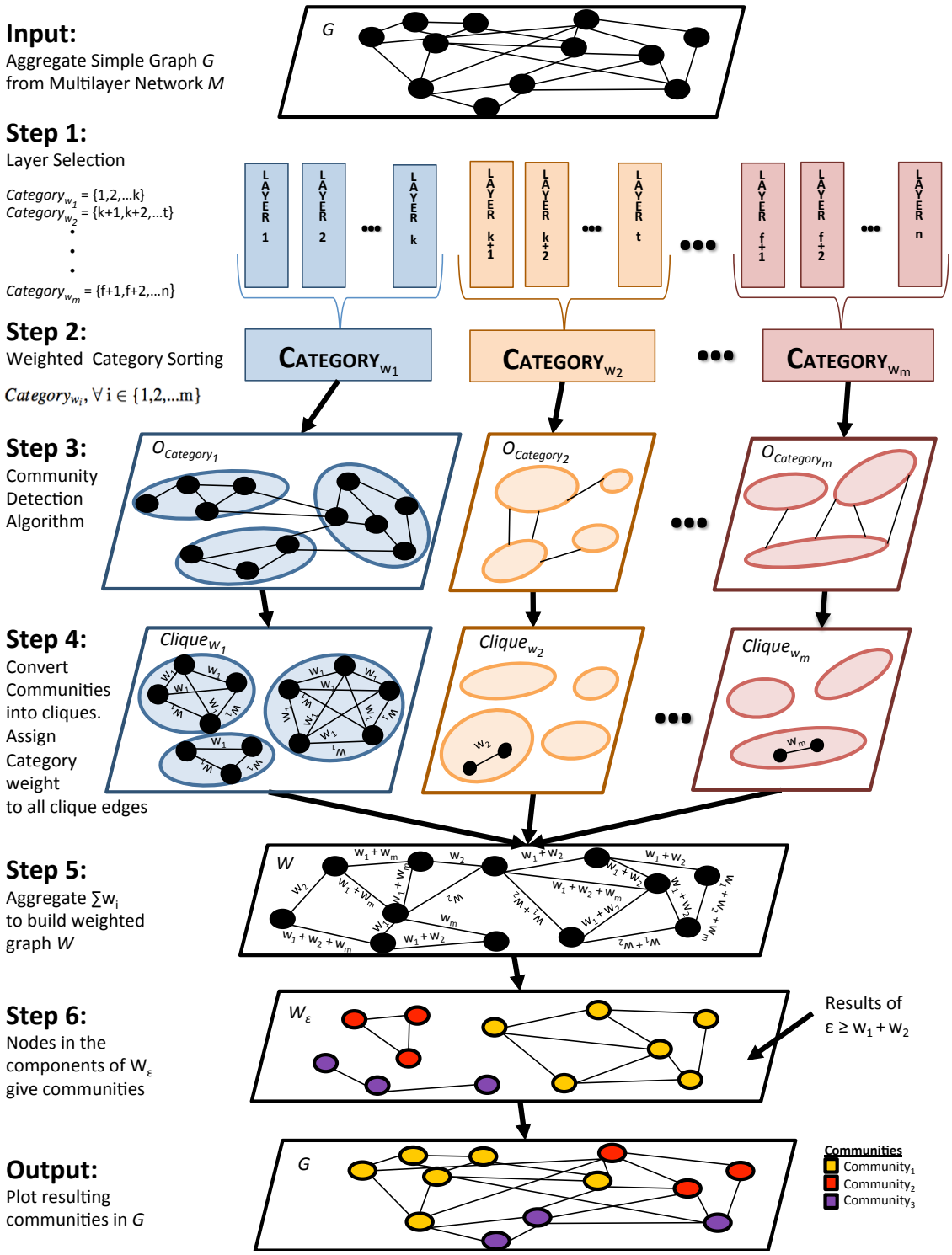



Figure 3.10: Algorithm overview (general case).

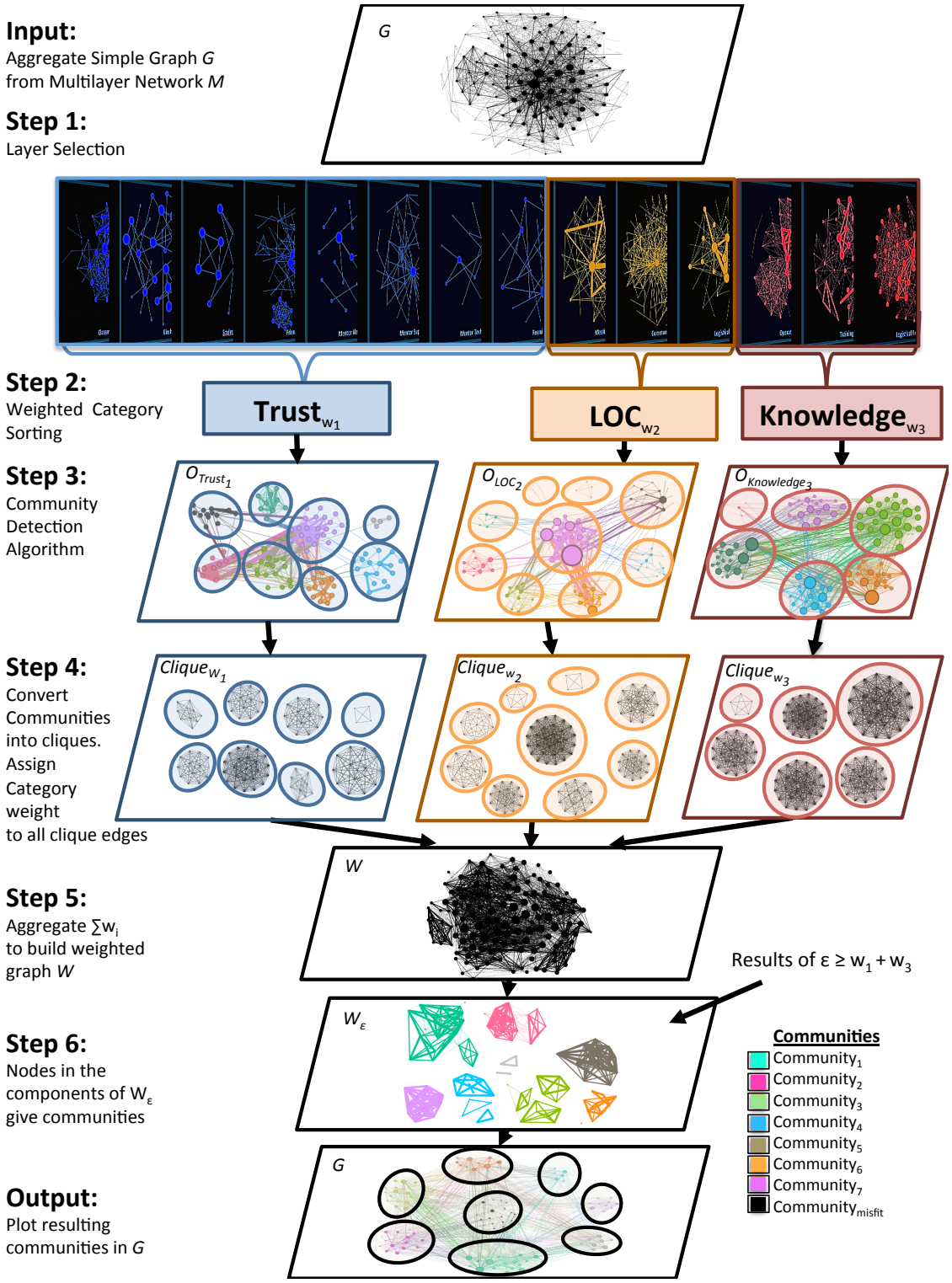



Figure 3.11: Algorithm overview (Noordin example)

**Algorithm 1** Multiplex community detection algorithm.

**Input:** Aggregate simple graph, $G$; multiplex layers, $l$; set of all layers in the multiplex, $S$; single layer community detection algorithm; category weights, $w_i$, where $i = \{1, 2, ..., m\}$; and threshold, $\varepsilon$; and PDC definition.

**Steps 1-2:** Manual layer selection and category formation from similar layers.

**for** $l$ in $S$ **do**: ▹ for each layer in the multiplex
  **if** $l$ supports the PDC definition **then**: ▹ user judgement
    Append $l$ to the layer selection list, $S'$ ▹ $S' \subset S$

**for** $l_i$ in $S'$ **do**: ▹ for each selected layer
  **for** $l_j$ in $S'$ **do**: ▹ for each selected layer
    **if** $l_i$ is similar in meaning to $l_j$ **then**: ▹ user judgement
      Append $l_i$ and $l_j$ to the same category

**Step 3:** Discover Communities in each category.

**for** each category **do**:
  Aggregate all layers ▹ creates sub-monoplex graph
  Perform single layer community detection ▹ user choice, default Louvain

**Step 4:** Convert communities to cliques and assign $w_i$.

**for** all communities in each sub-monoplex **do**:
  Remove external community edges ▹ creates component graphs
  Connect all vertices inside each community ▹ creates clique components
  **for** all edges in each clique component **do**:
    Assign value to $w_i$ ▹ user choice, default $w_i = 1 \forall i$

**Step 5:** Form consolidated weighted graph, $W$, by merging cliques from all categories with appropriate category weighting factor.

**for** each weighted edge in each sub-monoplex clique component **do**:
  Merge cliques from all categories ▹ builds $W$

**Step 6:** Threshold weighted graph, $W$.

**for** edges in $W$ **do**
  **if** $\varepsilon \geq \sum_{i=1}^{m} w_i$ **then**
    plot edge in graph $W_\varepsilon$ ▹ removes $\varepsilon < \sum_{i=1}^{m} w_i$, builds $W_\varepsilon$
    **if** $W_\varepsilon$ is sufficiently partitioned **then**: ▹ user judgement
      Components are the final communities
    **if** execute single layer community detection **then**: ▹ user choice, default Louvain
      New communities are the final communities
    **if** vertices from $G$ are not in communities **then**:
      place vertices in $Misfit_Community$ ▹ accounts for all vertices in $G$

**Output:** Final communities plotted in $G$

### 3.2.1 Step 1: Layer Selection

This first step is focused on preparing and selecting the network data that is most appropriate based on the user's goals. First, we examine the user's goals to understand the motivation behind identifying communities. From this we introduce the concept of the *purpose-driven communities* (PDC).

**Definition 3.2.1.** Purpose-Driven Communities

Given a multiplex network $M$ and a user $U$, the PDC are the intersection of user-inspired categorical communities based upon the analytical needs of $U$. The PDC's structural properties enhance the customer's understanding of the network in order to achieve the customer's objective.

For the general case depicted in Figure 3.10, a total of $n$ layers ($2 \leq n \leq |V(G)|$) are selected from the available network database of layers from the choices of $\{1, 2, ..., k, k + 1, ..., t, t + 1, ..., f, f + 1, ...n\}$. The selection of theses layers is entirely dependant on the user's analytical goals. The detection algorithm's success and subsequent depth of the community property analysis are also dependent on the available relationship data. This step may be revisited to include new layers or exclude current layers as appropriate. Once the layers have been selected, they will be sorted into one of $m$ categories ($m \leq n$) as described in Step 2. Layers are sorted into categories based on Kreb's observation that dark networks are sparse. The aggregation of similar layers into categories reduces sparseness and increases network density for more accurate community detection.

$$
\begin{aligned}
Category_{w_1} &= \{1, 2, ...k\} \\
Category_{w_2} &= \{k + 1, k + 2, ...t\} \\
&\cdot \\
&\cdot \\
&\cdot \\
Category_{w_m} &= \{f + 1, f + 2, ...n\}
\end{aligned}
$$

We apply Step 1 to the Noordin Top Network as an example. In the absence of a physical

customer, this thesis uses the Joint Improvised Explosive Device Defeat Organization (JIEDDO) Attack the Network (AtN) philosophy to infer the customer objectives.

Martin et al. [54] describe the mission of JIEDDO is "to focus, lead, advocate, and coordinate all Department of Defense actions in support of the Combatant Commanders' and their respective Joint Task Forces' efforts to defeat Improvised Explosive Devices as weapons of strategic influence." This thesis adapts the JIEDDO mission statement to include the prevention of coordinated terrorist operations. The United States Joint Forces Command [55] summarizes JIEDDO's AtN objectives to:

1. Identify key leaders in the network
2. Understand influence and relations
3. Identify and capitalize on vulnerabilities
4. Disrupt activities
5. Eliminate the ability for the network to function.

Understanding these customer objectives provides context and focus for our PDC definition. Based on the inferred assumptions about the customer's objectives, a more focused definition of PDC can be established. JIEDDO essentially wants to learn more about the terrorist network for the purposes of disrupting its ability to function. Based on this reasoning, our PDC for the Noordin Network are knowledge sharing communities (KSC).

**Definition 3.2.2.** Knowledge Sharing Communities

Given the Noordin Network and JIEDDO, the KSC are the intersection of Trust, Lines of Communication, and Knowledge communities based on the need to disrupt intra-organizational coordination in the Noordin Network.

Using the definition of KSC, we selected the following 14 layers illustrated in Figure 3.12 from the available 36 layers in the Noordin Network Data Set. Layers not included either were classified as weak and redundant or irrelevant layers. For example, the classmate layer was chosen over the education layer because the classmate layer included people who were in the same class in the same school whereas the education layer included people who went to the same school. Classmate thus established a stronger relationship tie and education was excluded as a redundant weaker layer. External communication is an example of an irrelevant layer that does not support the KSC definition and was thus not included in the

selected layers. This layer is irrelevant because it focuses on relationships outside of the network. In Step 2, these layers are sorted into weighted categories.

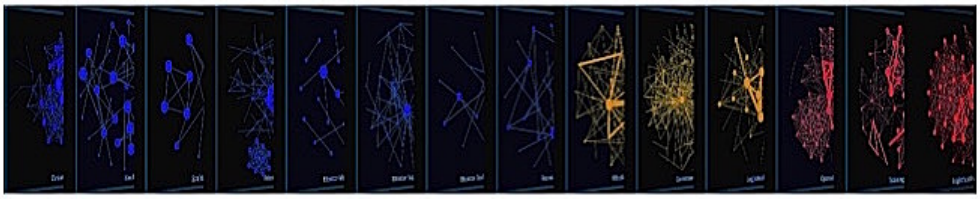


Figure 3.12: Step 1: Layer selection (Noordin example).

### 3.2.2 Step 2: Layer Sorting into Weighted Categories ($Category_{w_i}$)

Now that we have identified the $n$ layers, each layer is placed into exactly one of $m$ weighted categories

$$Category_{w_i}, \forall i \in \{1, 2, ...m\}.$$

Following our user-based philosophy, categories are chosen based on their relevance to the user's analytical goals. Weights are assigned to each category based on the degree of importance and associated contribution toward forming the PDCs. As a default, all categories are assigned a weight value of one. If a foundational category is identified, then the respective weight of the foundational category should be chosen to be greater than the summation of the remaining categories.

$$w_{foundation} > \sum_{i=1}^{m} w_i - w_{foundation}$$

**Definition 3.2.3.** Foundational Category

A category is labelled foundational if the relationships in this category are critical to the definition of the PDC.

A minimum of two categories is recommended for achieving analytical depth. If only one category were used then the resulting analysis would be the same as a collapsed simple graph. While an upper-bound is not mandated, it is suggested to be no higher than 50% of

the total number of layers.

$$2 \leq m < 0.5n.$$

An unbalanced number of categories may result in too many small community intersections. After the categories have been established, Step 1 may be revisited to adjust layer selection from the given network data set.

Applying Step 2 to the Noordin Network results in the 14 layers being sorted into three categories, as illustrated in Figure 3.13. The layers were sorted into categories based on the KSC definition. In the case of our dark networks, three categories was optimal for grouping similar layers. However, the PDC definition and available network data may lead the user to create more categories.

1. **Trust:** Members of the KSC must trust each other in order to develop the will to communicate. Layers included build or demonstrate trust between members.
2. **Lines of Communication (LOC):** Members of the KSC require a communication medium to share knowledge. Layers included in this category allow members to share knowledge.
3. **Knowledge:** Members of the KSC need meaningful information or knowledge to share. Layers included are tasks, events, and resources that members want or need to share using one of the LOC layers.

The use of these three categories allows our algorithm to produce members of a KSC that have knowledge that they are capable of and willing to share with other members of the KSC. The topological characteristics for the categories and the aggregate monoplex, $O$, for Noordin, Boko Haram, and FARC terrorist networks are represented in Table 3.4, Table 3.5, and Table 3.6 respectively.

Table 3.4: Noordin Network topological characteristics by category.

| Category Name | V | E | AD | AWD | Di | De | M | ACC | APL | P |
|---|---|---|---|---|---|---|---|---|---|---|
| Trust | 111 | 544 | 7.53 | 9.80 | 7 | 0.07 | 0.51 | 0.66 | 3.10 | 3 |
| LOC | 121 | 534 | 6.33 | 8.83 | 7 | 0.05 | 0.38 | 0.57 | 2.92 | 1 |
| Knowledge | 106 | 1373 | 22.4 | 25.9 | 5 | 0.21 | 0.41 | 0.79 | 1.93 | 3 |
| Average | 113 | 817 | 12.0 | 14.8 | 6 | 0.11 | 0.43 | 0.89 | 2.65 | 2 |
| Monoplex ($O$) | 133 | 2451 | 22.5 | 36.9 | 5 | 0.17 | 0.35 | 0.71 | 2.13 | 1 |

Table 3.5: Boko Haram Network topological characteristics by category.

| Category Name | V | E | AD | AWD | Di | De | M | ACC | APL | P |
|---|---|---|---|---|---|---|---|---|---|---|
| Trust | 29 | 31 | 2.07 | 2.14 | 4 | 0.07 | 0.56 | 0.26 | 2.28 | 5 |
| LOC | 17 | 34 | 2.47 | 4.00 | 2 | 0.15 | 0.33 | 0.92 | 1.48 | 5 |
| Knowledge | 21 | 34 | 2.95 | 3.24 | 7 | 0.15 | 0.46 | 0.56 | 2.64 | 4 |
| Average | 22 | 33 | 2.50 | 3.13 | 4 | 0.12 | 0.45 | 0.58 | 2.13 | 5 |
| Monoplex ($O$) | 44 | 99 | 3.32 | 4.50 | 5 | 0.08 | 0.50 | 0.50 | 2.42 | 6 |

Table 3.6: FARC Network topological characteristics by category.

| Category Name | V | E | AD | AWD | Di | De | M | ACC | APL | P |
|---|---|---|---|---|---|---|---|---|---|---|
| Trust | 32 | 28 | 1.68 | 1.75 | 3 | 0.05 | 0.81 | 0.39 | 1.61 | 8 |
| LOC | 130 | 1614 | 23.2 | 24.8 | 6 | 0.18 | 0.51 | 0.93 | 2.28 | 3 |
| Knowledge | 13 | 8 | 1.23 | 1.23 | 2 | 0.10 | 0.78 | 0.00 | 1.27 | 5 |
| Average | 58 | 550 | 8.70 | 9.26 | 4 | 0.11 | 0.70 | 0.44 | 1.72 | 5 |
| Monoplex ($O$) | 142 | 1650 | 21.5 | 23.2 | 8 | 0.15 | 0.52 | 0.91 | 2.90 | 1 |

We tested several sets of cases on the Noordin Network using different weights. During one case study, the Trust category was given the highest weight of $w_1 = 4$, followed by LOC with $w_2 = 2$ and Knowledge with $w_3 = 1$. This weighting system was applied based on the reasoning that Trust is the foundational category required to build KSCs. The dynamic nature of the dark network allows two people that are only connected by trust to potentially develop LOC and Knowledge and ultimately build a KSC. The combined weight of LOC and Knowledge is intentionally less than Trust to further establish Trust as a foundational category ($w_1 > w_2 + w_3$).

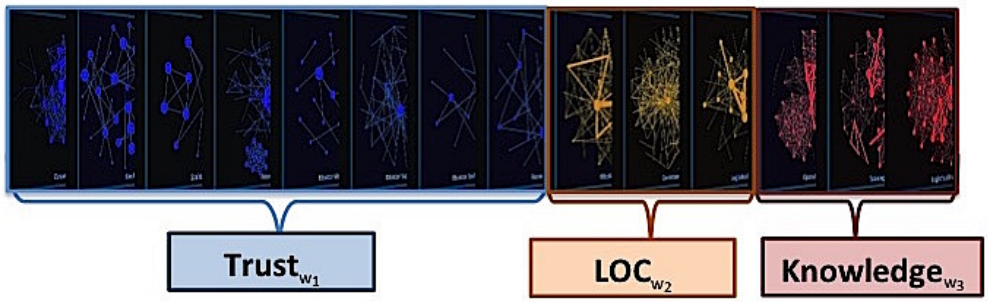



Figure 3.13: Step 2: Weighted category sorting (Noordin example).

### 3.2.3 Step 3: Community Detection on Categories

All of the layers of each $Category_{w_i}$ are aggregated to form a sub-monoplex network $O_{Category_i}$. The user is given the option to choose which single layer community detection algorithm to implement on each $O_{Category_i}$. This research recommends the well established and used *Louvain* method described in Section 2.2 due to its relatively efficient computational complexity based on accuracy.

The results of applying this step to the Noordin Network Categories are depicted in Figure 3.14.

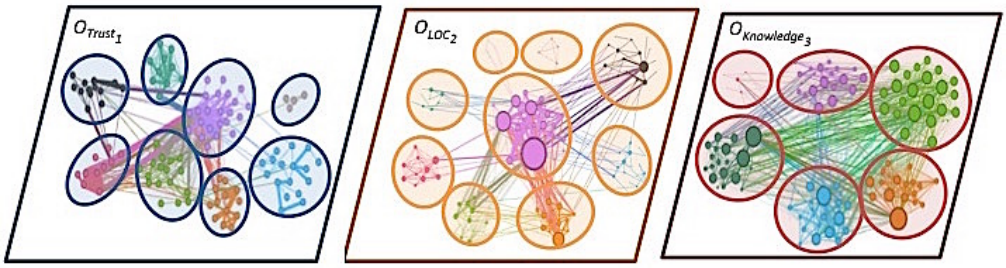



Figure 3.14: Step 3: Community detection algorithm (Noordin example).

### 3.2.4 Step 4: Community to Clique Conversion

The resultant communities for each category are converted into cliques. Each community within the categories is represented as a complete graph to emphasize the edge relationship

of belonging to the same community. The edges within each category are given the same respective categorical weight.

Applying Step 4 to the Noordin Network results in the complete community cliques for Trust, LOC, and Knowledge illustrated in Figure 3.15.

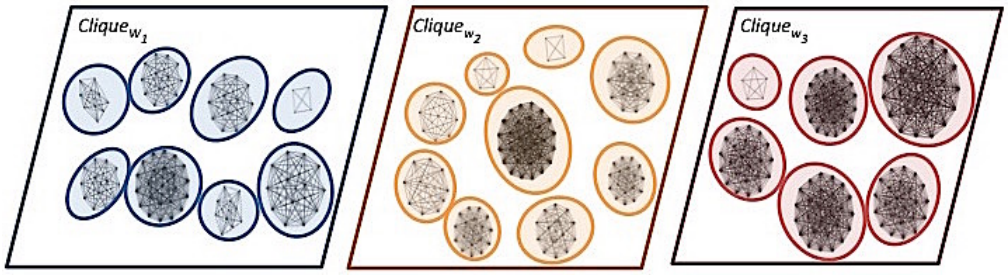



Figure 3.15: Step 4: Community to clique conversion (Noordin example).

### 3.2.5 Step 5: Build the Weighted Graph $W$

This step combines the resultant clique communities from all of the categories into an aggregate weighted graph $W$. The edge weight, $e_{w_{jk}}$ between any two vertices $v_j$ and $v_k$ in $W$ is the summation of the edge weights between $v_j$ and $v_k$ from each category $m$.

$$e_{w_{jk}} = \sum_{i=1}^{m} w_i, \forall jk \in Category_{w_i}, \forall i \in \{1, 2, \ldots, m\}.$$

Applying Step 5 to the Noordin Network results in the aggregate graph $W$ pictured in Figure 3.16.

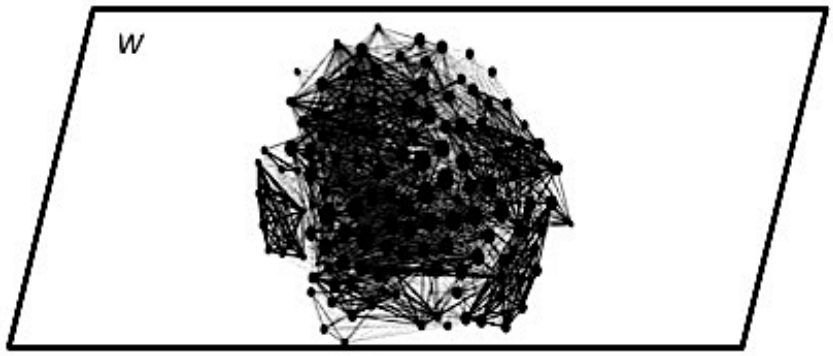



Figure 3.16: Step 5: Weighted graph, $W$ (Noordin example).

### 3.2.6 Step 6: Communities Through Tolerance $\varepsilon$ Selection

The final step of this method is again user-driven to determine an acceptable threshold tolerance, $\varepsilon$. Choosing different thresholds creates a constraint on the graph that limits the amount of data considered to build communities. A choice of $\varepsilon = \sum_{i=1}^{m} w_i$ carries the strongest meaning and true intersection of communities across the $m$ categories. The user is left to decide an acceptable value for $\varepsilon$ and may want to experiment with different $\varepsilon$ values to produce the desired meaningful communities. The sum of all of the category weights serves as a logical upper-bound for $\varepsilon$. If a foundational category exists, then the weight of the foundational category is recommended as a lower bound for $\varepsilon$. This prevents the other categories from forming communities without including the foundational category.

The threshold selection of $\varepsilon$ results in partitioning $W$ into components. These components of $W$ are the PDCs. Any components that contain only one vertex are placed into a misfit community. As a final output, the algorithm plots the resultant PDCs' nodes onto $O$ to observe inter-community relations in the Network to create the final communities plotted in $O$.

Applying Step 6 to the Noordin Network results in the KSC identification and the subsequent plot in $O$ illustrated in Figure 3.17. In order to demonstrate this step, $\varepsilon \geq w_1 + w_2$ was selected as an example to partition the graph into KSCs. The following bounds are

recommended for cases with a foundational category for $\varepsilon$:

$$max\{w_i\} \leq \varepsilon \leq \sum_{i=1}^{m} w_i, \forall i \in \{1, 2, ...m\}.$$

**Optional Step:** If the output results in one component or the network is not sufficiently partitioned according to the user's goals, we suggest executing single layer detection once more on $W_\varepsilon$. Why would we do this? Lets consider what $W_\varepsilon$ actually represents. This is a graph with vertices that are related because of the chosen threshold, $\varepsilon$, intersection of categorical community relations. Performing community detection once more will partition $W_\varepsilon$ into groups of vertices that are more related by this intersection of categorical communities inside their group than to other vertices outside their community. This optional process is consistent with the integrity of our definition of KSC.

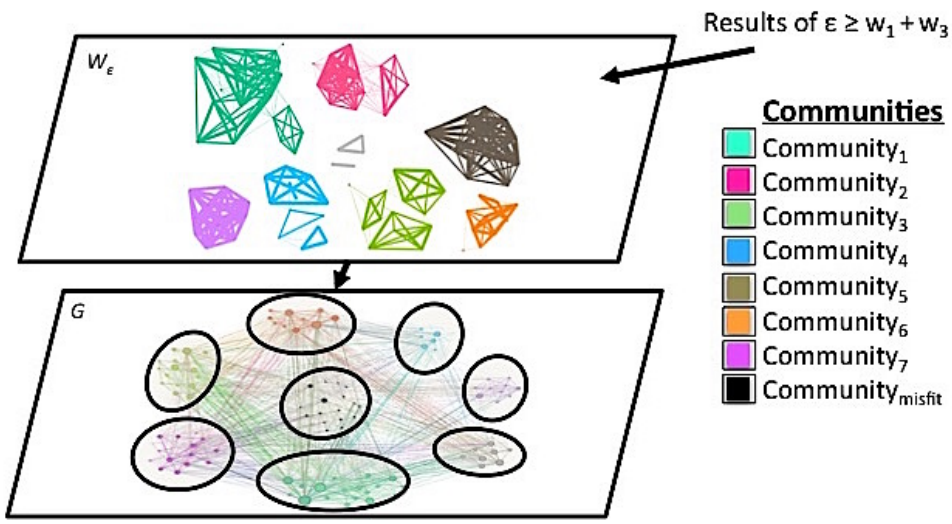



Figure 3.17: Step 6: KSCs and plot in $O$ (Noordin example).

Algorithm 1 was applied to all three dark network case studies. The results of this algorithm are studied in the next chapter, Chapter 4.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 4:
# Results and Analysis

This chapter focuses on our experiment design, displaying and analyzing our results. First, we define our experiment in Section 4.1. All sections that follow contain the results and subsequent analysis of applying our community detection methodology on the three dark networks from Chapter 3.

## 4.1 Experiment Design

Three case studies were considered for each dark network. Each case study represents a different selection of weight values for $w_1$, $w_2$, and $w_3$ as described in Step 3 of our methodology. We then studied nine subcases for each case that corresponds to different choices for epsilon as described in Step 6 of our methodology. Figure 4.1 depicts the organization of the different weight cases and threshold subcases studied in this chapter.

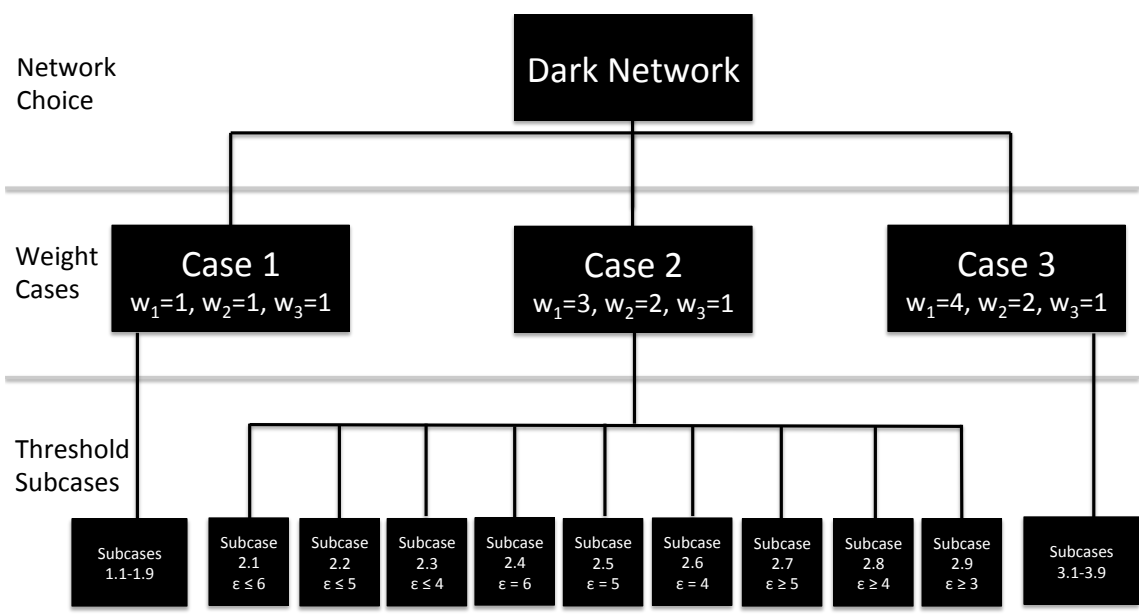



Figure 4.1: Chapter 4 case study organization.

We grouped the subcases in sets of three based on $\varepsilon$ values corresponding to : $\varepsilon \leq w_1 + w_2 + w_3$, $w_1 + w_2$, and $w_1 + w_3$; $\varepsilon = w_1 + w_2 + w_3$, $w_1 + w_2$, and $w_1 + w_3$; and $\varepsilon \geq w_1 + w_2$, $w_1 + w_3$, and $w_1$ respectively. For display purposes, we focused on plotting the individual subcase community results for the Noordin Network. First, we show the output community graphs in the original monoplex network from Step 6 of our methodology. For visual clarity, communities of size less than or equal to three are colored grey in these plots. A legend with the corresponding community name, community size percentage, and color accompanies each community output graph.

We follow the community output graphs with the community size and adjusted conductance plots per community. The community names are the independent variables on the $x$-axis with $M$ representing the misfit community. The community size and adjusted conductance values are the dependant variables on $y$-axis. We plotted two different adjusted conductance values. Adjusted conductance $W$ is the result of re-plotting the resultant communities from our methodology back into the weighted graph $W$. Adjusted conductance $O$ is the result of re-plotting the resultant communities from our methodology back into the original monoplex graph $O$. These values are represented as red circles connected by a dotted red line. Displaying the community size information allows us to potentially correlate adjusted conductance to size and identify specific communities that are present in multiple subcases. We also believe there may be some correlation between the average adjusted conductance and the size of the misfit community. The community size is represented as blue diamonds with a solid blue line. All plots are ordered by either increasing adjusted conductance $W$ or increasing average adjusted conductance $W$.

After all of the network cases have been examined, we display the average results for the network in a summary plot. This includes the average community size, cluster adequacy $W$, cluster adequacy $O$, average adjusted conductance $W$, and average adjusted conductance $O$ represented as blue diamonds and a solid line, green squares and a solid line, green circles and a dotted line, red squares and a solid line, and red circles and a dotted line respectively.

Recall from Section 2.2 that the conductance, $\phi$, ranges from 0 to 1 and that the smaller the conductance value, the better the community quality. Since cluster adequacy values range from 0 to 1 with larger values resulting in stronger communities, we adjusted the conductance values for metric comparison clarity. We achieved this by subtracting the

conductance values from 1. As a result, larger values for adjusted conductance, $\phi^1$, and for cluster adequacy are considered consistent with higher quality communities.

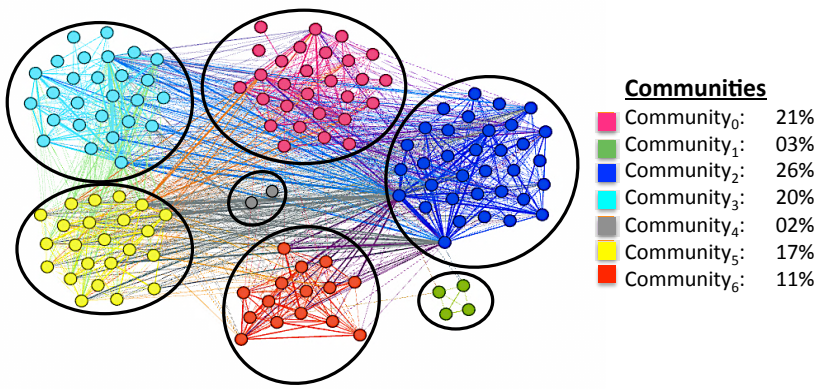
We compare each plot against an established control case. For the purposes of this thesis, we use the communities that result from implementing the Louvain method on the original monoplex network as our control. The control case average cluster adequacy $O$ and average adjusted conductance $O$ are displayed on the summary plots as a dashed green and dashed red lines respectively. This control provides us with an established benchmark for community quality comparison.

Before conducting this experiment, we established our hypothesis for the quality of communities that are produced by the different cases and subcases. Qualitatively, we believe Case 3 will produce the most meaningful communities based on our definition of KSC. Case 3 provides the necessary category weight distribution for the trust foundation category to dominate the remaining community information from the other categories. Krebs and Everton established the importance of trust to the functionality and resilience of dark network in Section 2.4. Forcing trust to be included as the lower bound for epsilon choices in this case is consistent with their convictions.

Quantitatively, our intuition is that for a given value, $v$, subcases that involve $\varepsilon \leq v$ will result in a small number of large sized communities. We believe this threshold will be too relaxed of a choice for $\varepsilon$. The subcases for $\varepsilon = v$ will produce many communities that are very small as we exclude particular relationships from enforcing an equality in the treshold versus inequality. Also, it prevents vertices from being neighbors in certain categories in order to achieve equality, which doesn't seem to be realistic, but we consider them for completeness. Consequently, these cases may be too restrictive of a choice for $\varepsilon$. Finally, we believe the subcases for $\varepsilon \geq v$ will produce better communities since these subcases are more relaxed than $\varepsilon = v$, yet more restrictive than $\varepsilon \leq v$, requiring that vertices are friends in at least that many categories, but possibly more.

## 4.2 Noordin Results and Analysis

In this section we display the results and analysis of applying our methodology to the Noordin Network. First, we display the community graph results and the size and conductance values for the control case in Figure 4.2. We follow the control case with the results and preliminary observations from case 1, case 2, and case 3.



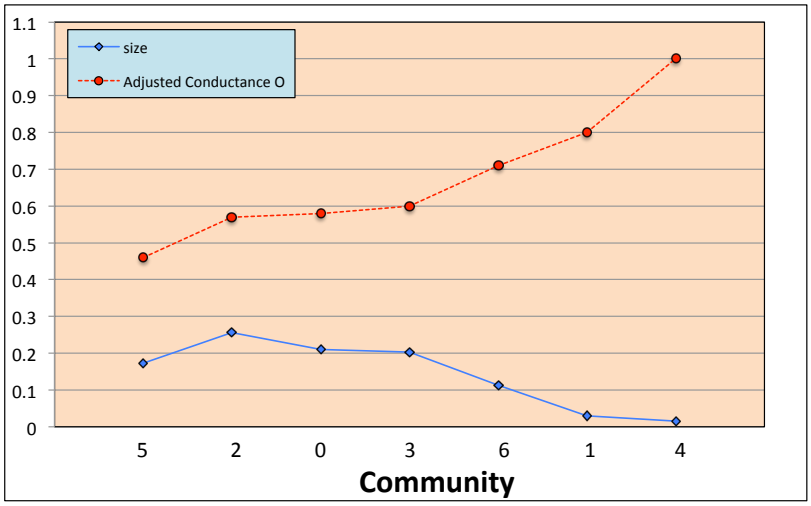




Figure 4.2: Noordin control case community output plot and size, and conductance plot.

The control case yielded a total of seven communities, five of which were large and two relatively small. We observe that the smallest communities have the highest adjusted conductance in Figure 4.2 and that the smallest community, $community_4$, has a perfect adjusted conductance value of one. It is important to note that $community_4$ is an isolated component of the network. Component communities have no external community connections, which results in no neighbors in $\overline{S}$. Since $\overline{S}$ is empty, the conductance calculation simplifies to zero, which yields a value of one for adjusted conductance. $Community_4$ from the control case is also visible in subcases where $\varepsilon \leq v$. For example, in case 1, subcases $1.1, 1.2, and 1.3$ as $community_3$, $community_2$, and $community_2$ respectively in Figure 4.4. Upon further inspection, this two vertex component relationship exists only in one category, trust. This community does not exist in the remaining subcases due to the increased restriction imposed by $\varepsilon = v$ and $\varepsilon \geq v$.

### 4.2.1 Noordin Results: Case 1

Case 1 examines a uniform distribution of weight values including: $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. This case serves as a default if the user is unable to determine a logical ordering for category importance. Consequently, this case represents equal importance amongst all categories. We examine the subcase community graphs for case 1 in Figures 4.3, 4.5, and 4.7. We follow these graphs with the size and adjusted conductance per community subcase plots in Figures 4.4, 4.6, and 4.8.

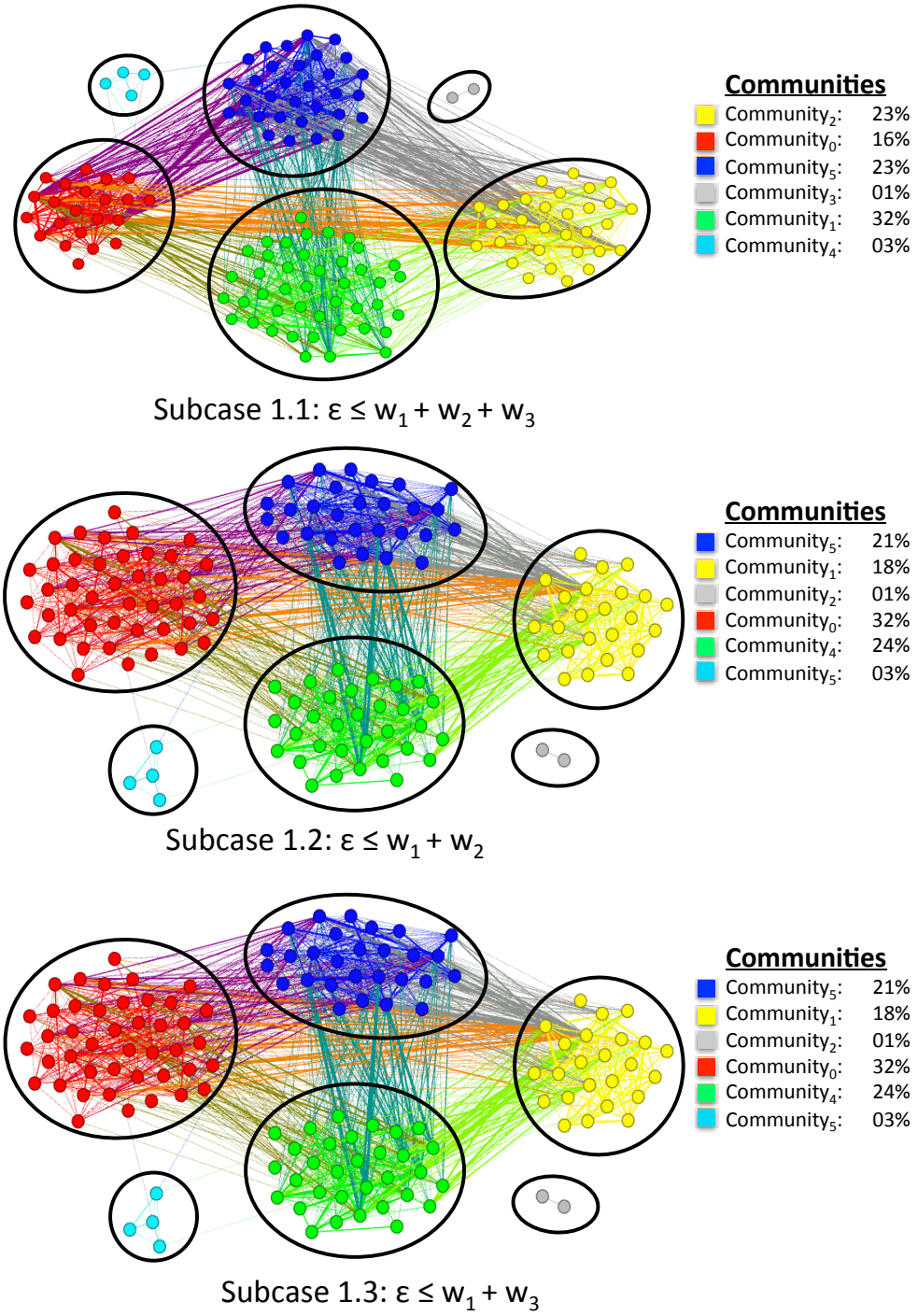
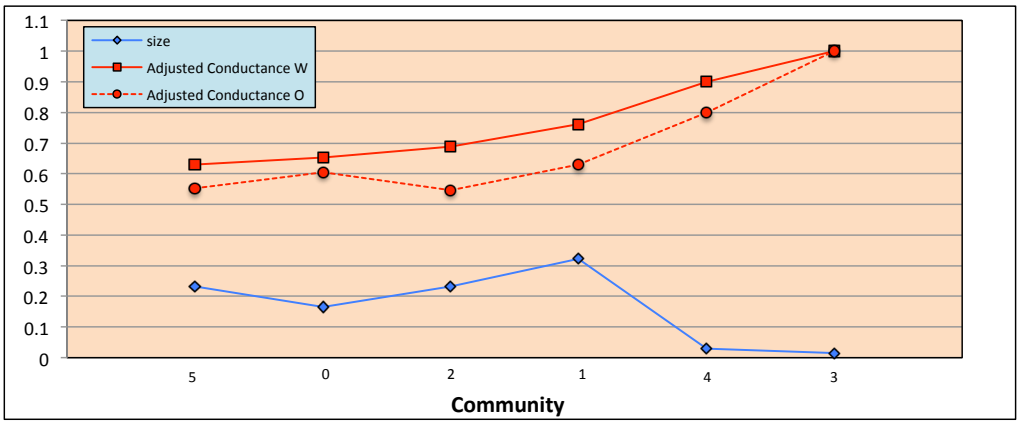
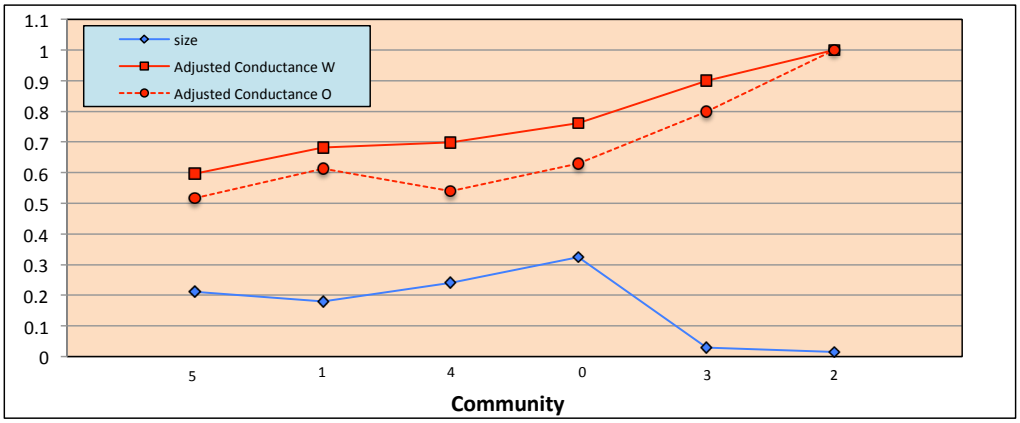


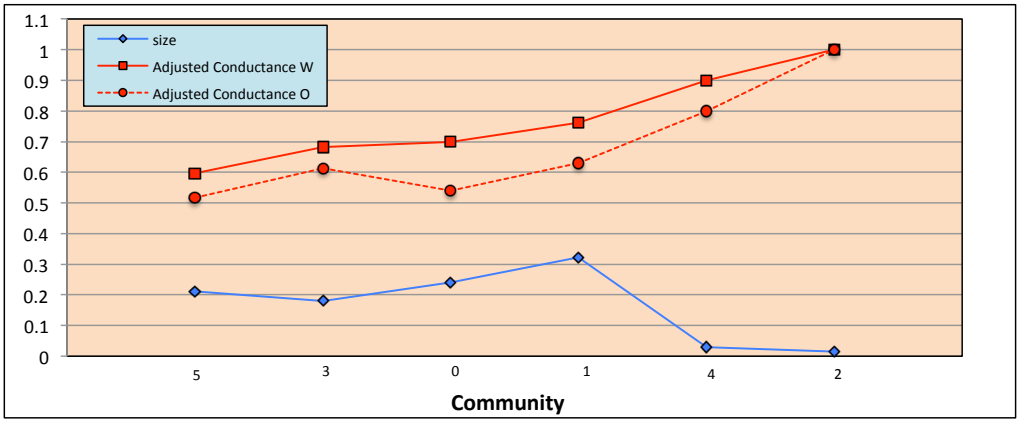

Figure 4.3: Noordin community output plot for subcases 1.1-1.3 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$.




Subcase 1.1: $\varepsilon \leq w_1 + w_2 + w_3$




Subcase 1.2: $\varepsilon \leq w_1 + w_2$




Subcase 1.3: $\varepsilon \leq w_1 + w_3$

Figure 4.4: Noordin community size and normalized conductance for subcases 1.1-1.3 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$.

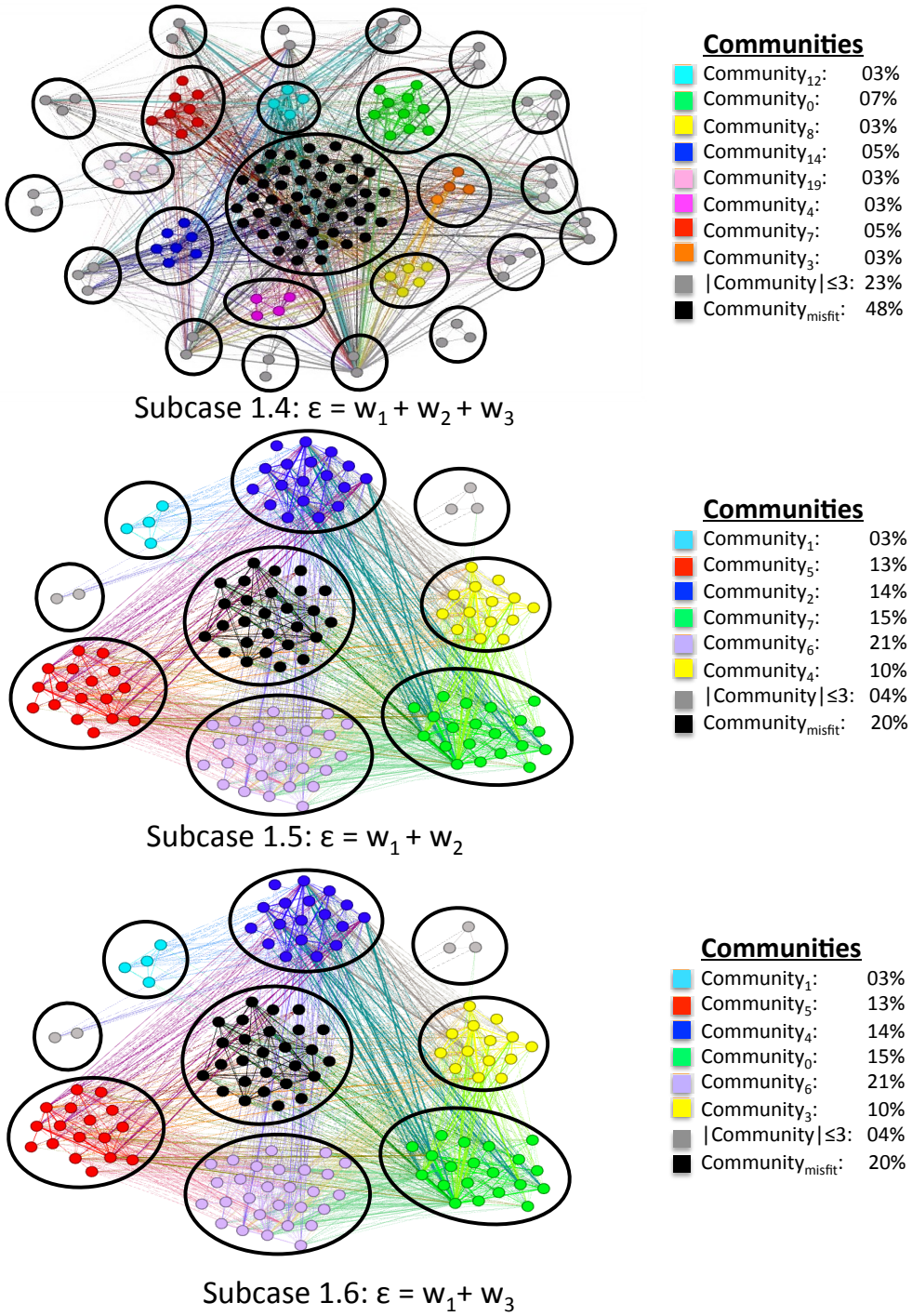
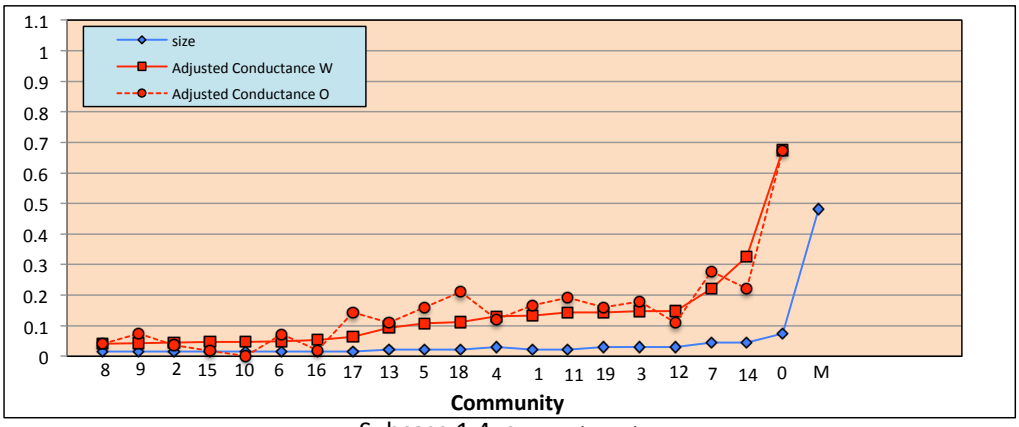
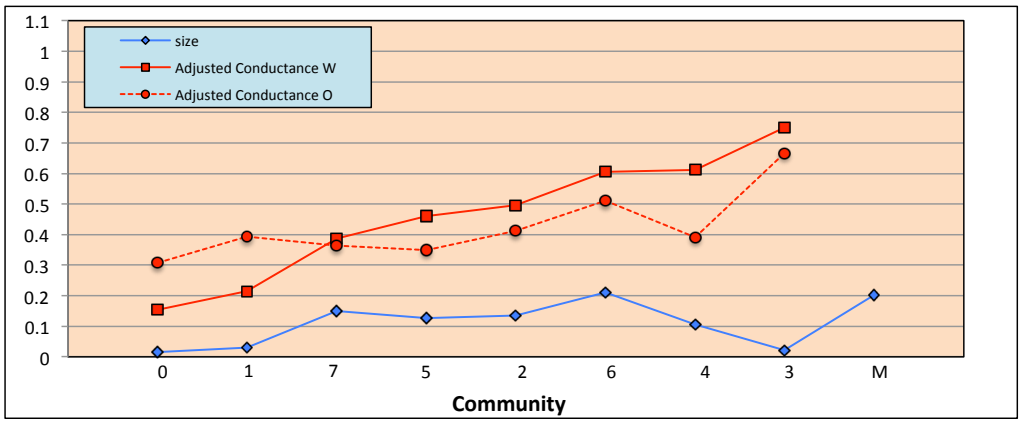


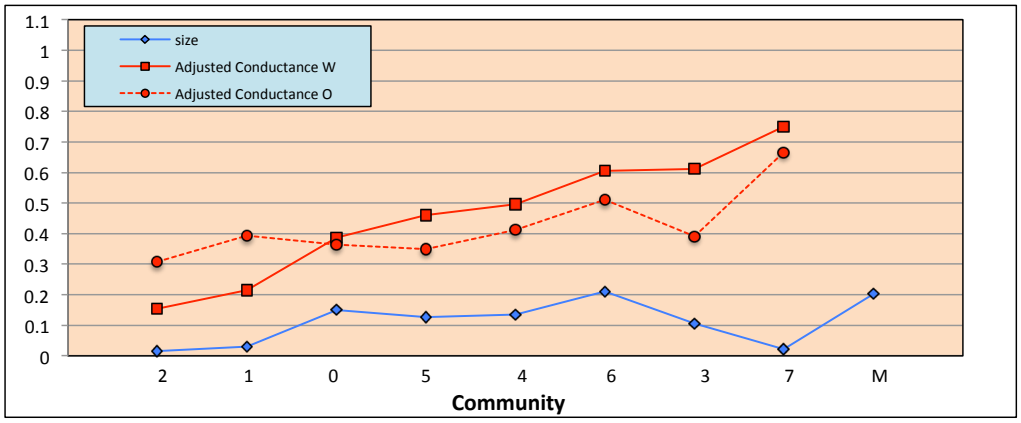

Figure 4.5: Noordin community output plot for subcases 1.4-1.6 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$.




Subcase 1.4: $\varepsilon = w_1 + w_2 + w_3$

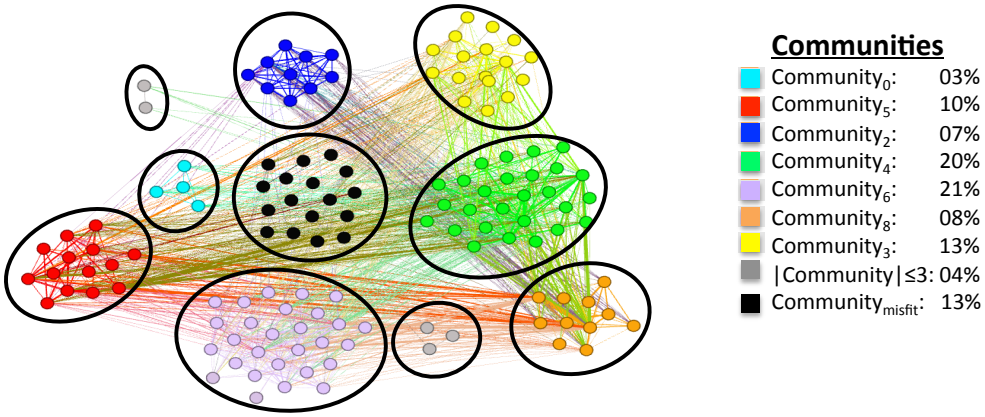



Subcase 1.5: $\varepsilon = w_1 + w_2$

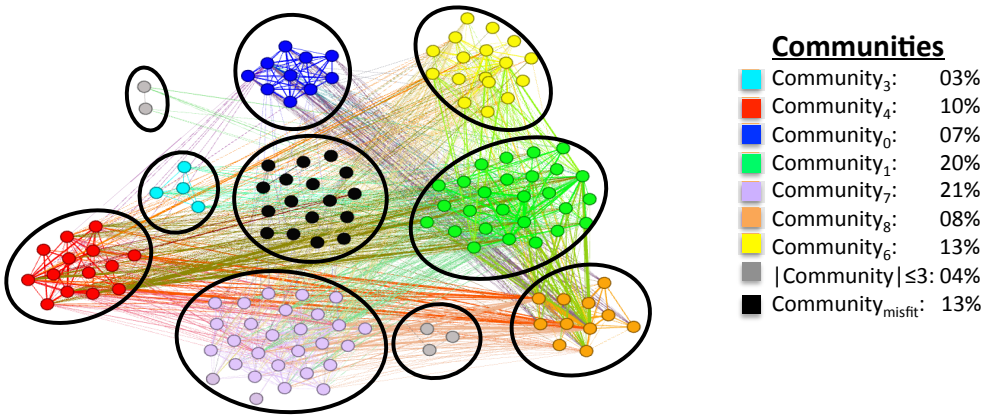



Subcase 1.6: $\varepsilon = w_1 + w_3$

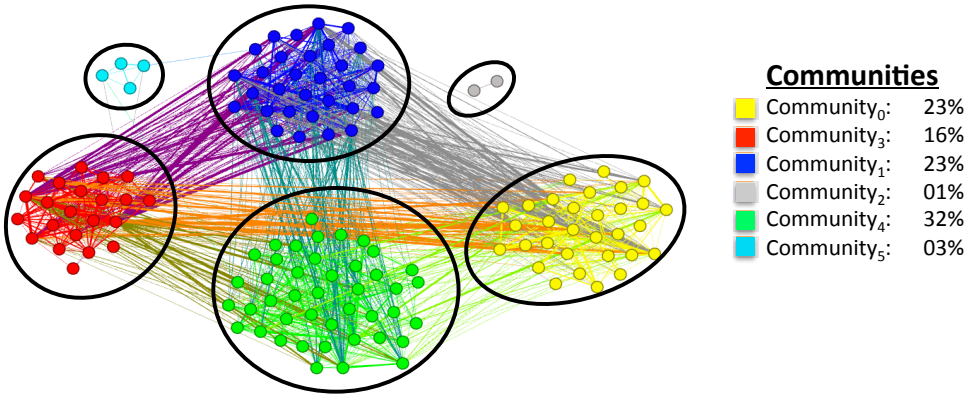
Figure 4.6: Noordin community size and normalized conductance for subcases 1.4-1.6 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$.




Subcase 1.7: $\varepsilon \geq w_1 + w_2$

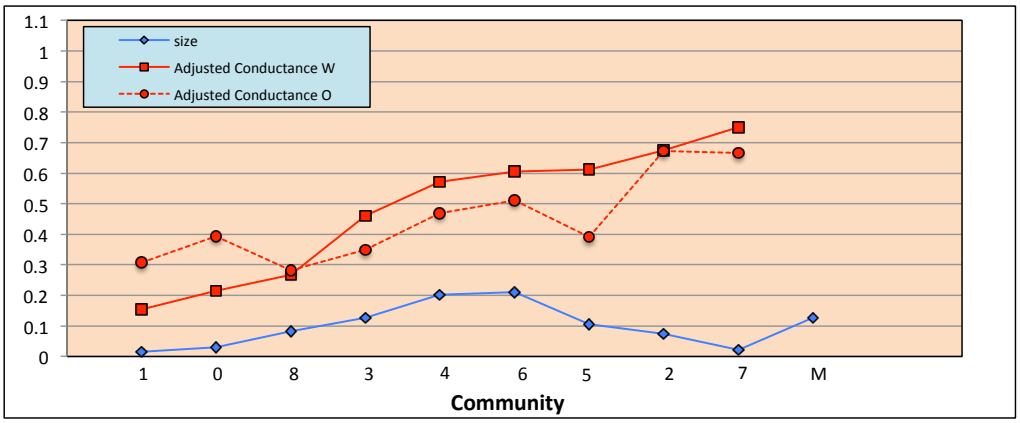



Subcase 1.8: $\varepsilon \geq w_1 + w_3$

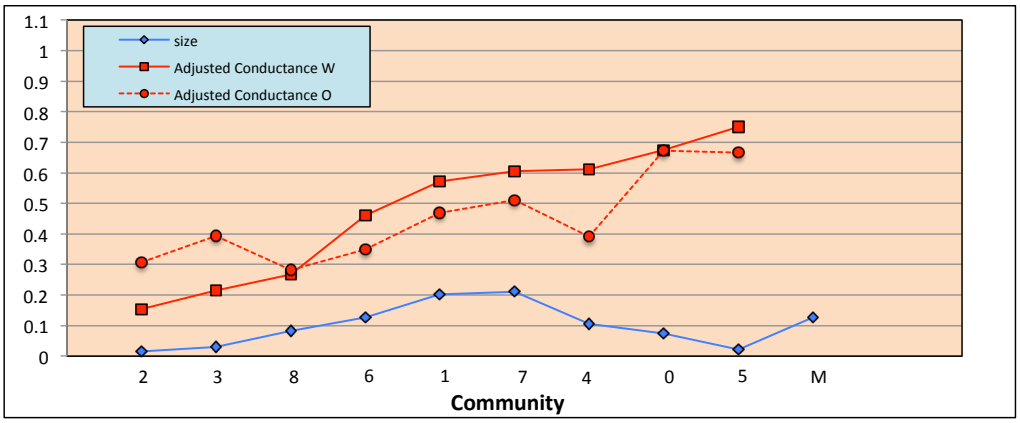



Subcase 1.9: $\varepsilon \geq w_1$

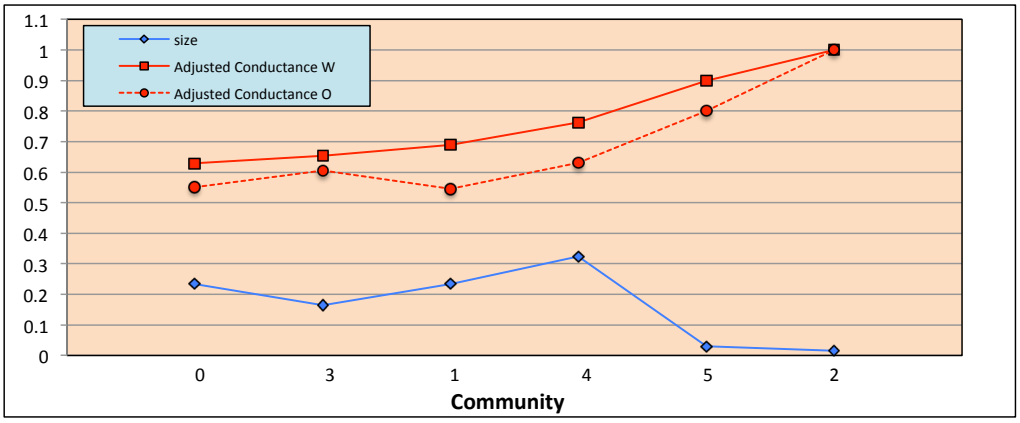
Figure 4.7: Noordin community output plot for subcases 1.7-1.9 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$.




Subcase 1.7: $\varepsilon \geq w_1 + w_2$




Subcase 1.8: $\varepsilon \geq w_1 + w_3$




Subcase 1.9: $\varepsilon \geq w_1$

Figure 4.8: Noordin community size and normalized conductance for subcases 1.7-1.9 with $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$.

Case 1 subcase graphs and plots provide an initial basis of comparison for the weighted cases 2 and 3. Due to equal weighting, some redundancy is observed in case 1. Algebraically, $w_1 + w_2 = 2$ and $w_1 + w_3 = 2$ are equivalent. Consequently, we observe the same behavior for subcases 1.2 and 1.3, 1.5 and 1.6, and 1.7 and 1.8. In Figure 4.6 we observe that the $\varepsilon = v$ subcases produce many small communities with relatively low conductance. In particular, subcase 1.4 performed the poorest in terms of adjusted conductance and it also has the highest misfit community. We also observe that the adjusted conductance $W$ values typically are higher than the adjusted conducted $O$. This is to be expected since the communities were formed using the edges from $W$. By plotting the communities in $O$, we no longer have the inferred edges we artificially attached during our community detection process. The adjusted conductance $W$ values also tended to follow the same shape as the adjusted conductance $O$ values. After examining case 1, it appears that subcase 1.1 produces the best communities in terms of adjusted conductance when plotted in both $W$ and $O$.

### 4.2.2 Noordin Results: Case 2

Case 2 examines the weight distribution of $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$. This case represents an established ordering of category importance. However, when both the LOC and knowledge categories are included in the subcase, they are collectively equivalently weighted to the trust category. Thus the trust category dominates individual categories, but not the coalition of other categories. We examine the subcase community graphs for case 2 in Figures 4.9, 4.11, and 4.13. We follow these graphs with the size and conductance per community subcase plots and summary tables in Figures 4.10, 4.12, and 4.14.

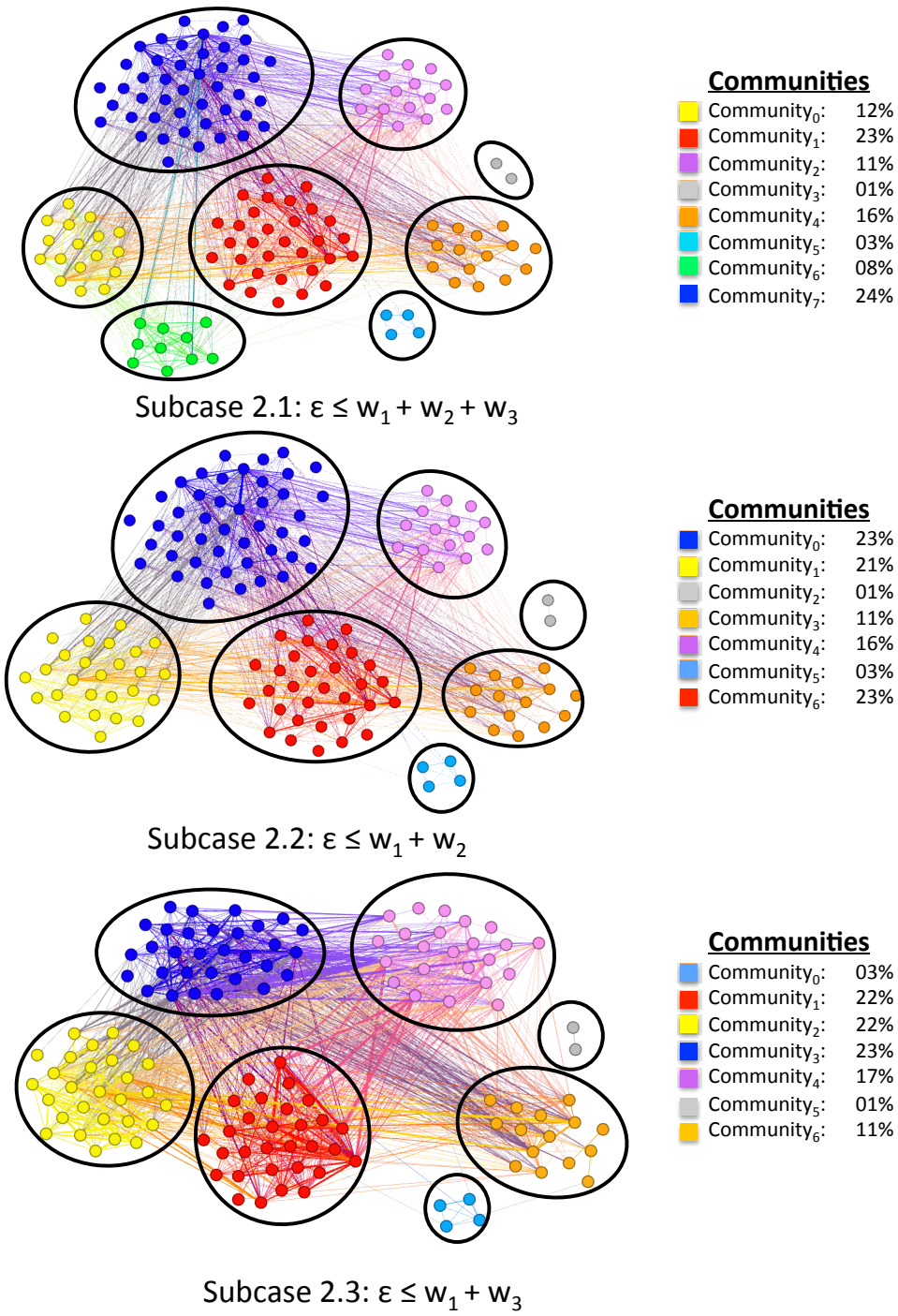
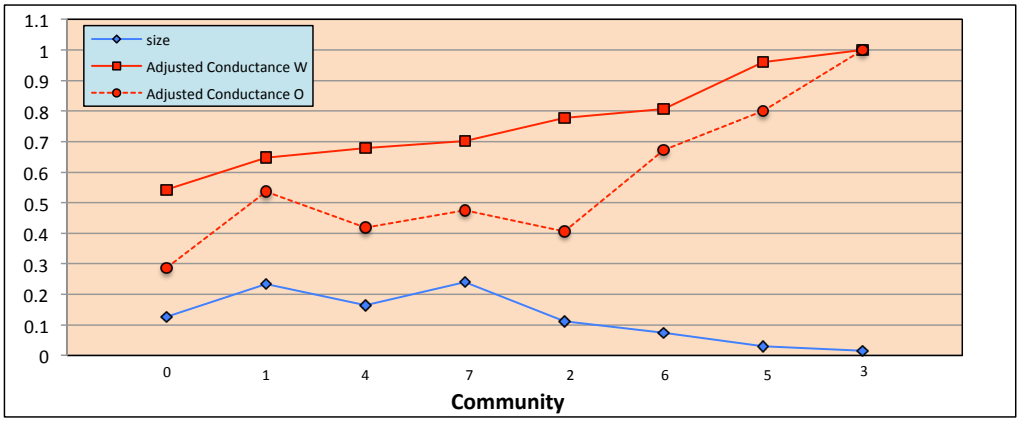
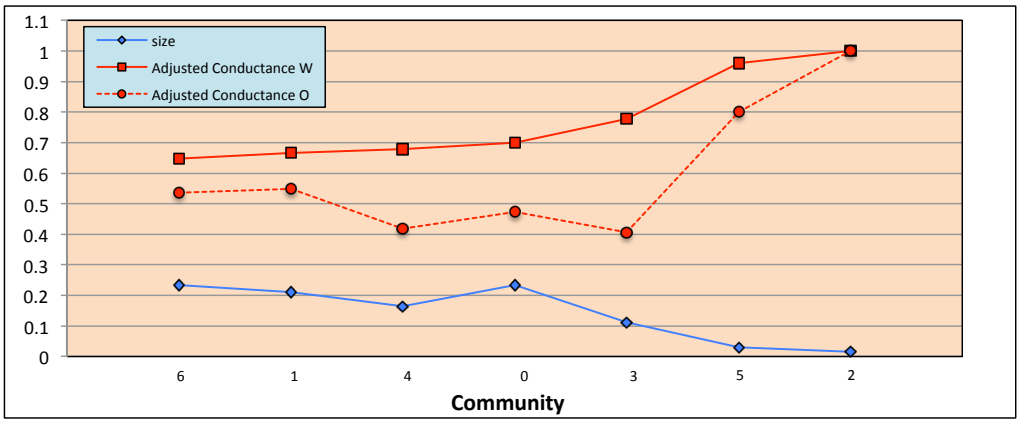


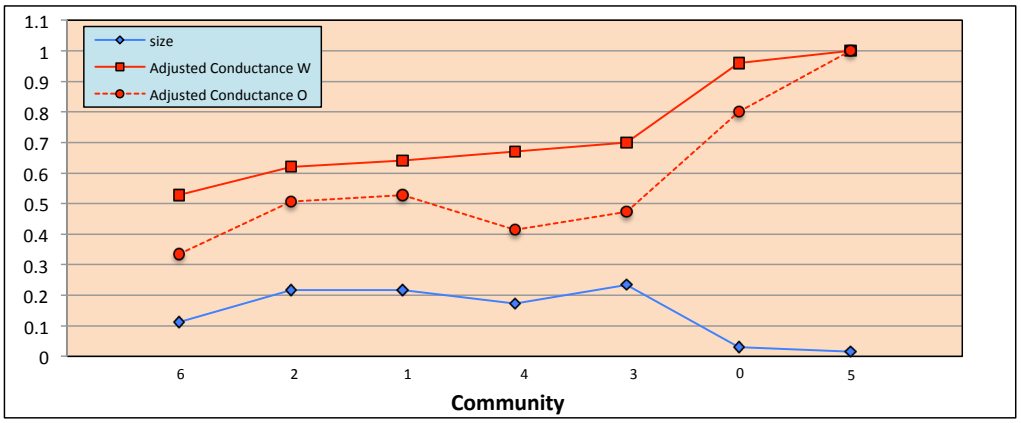

Figure 4.9: Noordin community output plot for subcases 2.1-2.3 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$.




Subcase 2.1: $\varepsilon \leq w_1 + w_2 + w_3$




Subcase 2.2: $\varepsilon \leq w_1 + w_2$




Subcase 2.3: $\varepsilon \leq w_1 + w_3$

Figure 4.10: Noordin community size and normalized conductance for subcases 2.1-2.3 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$.

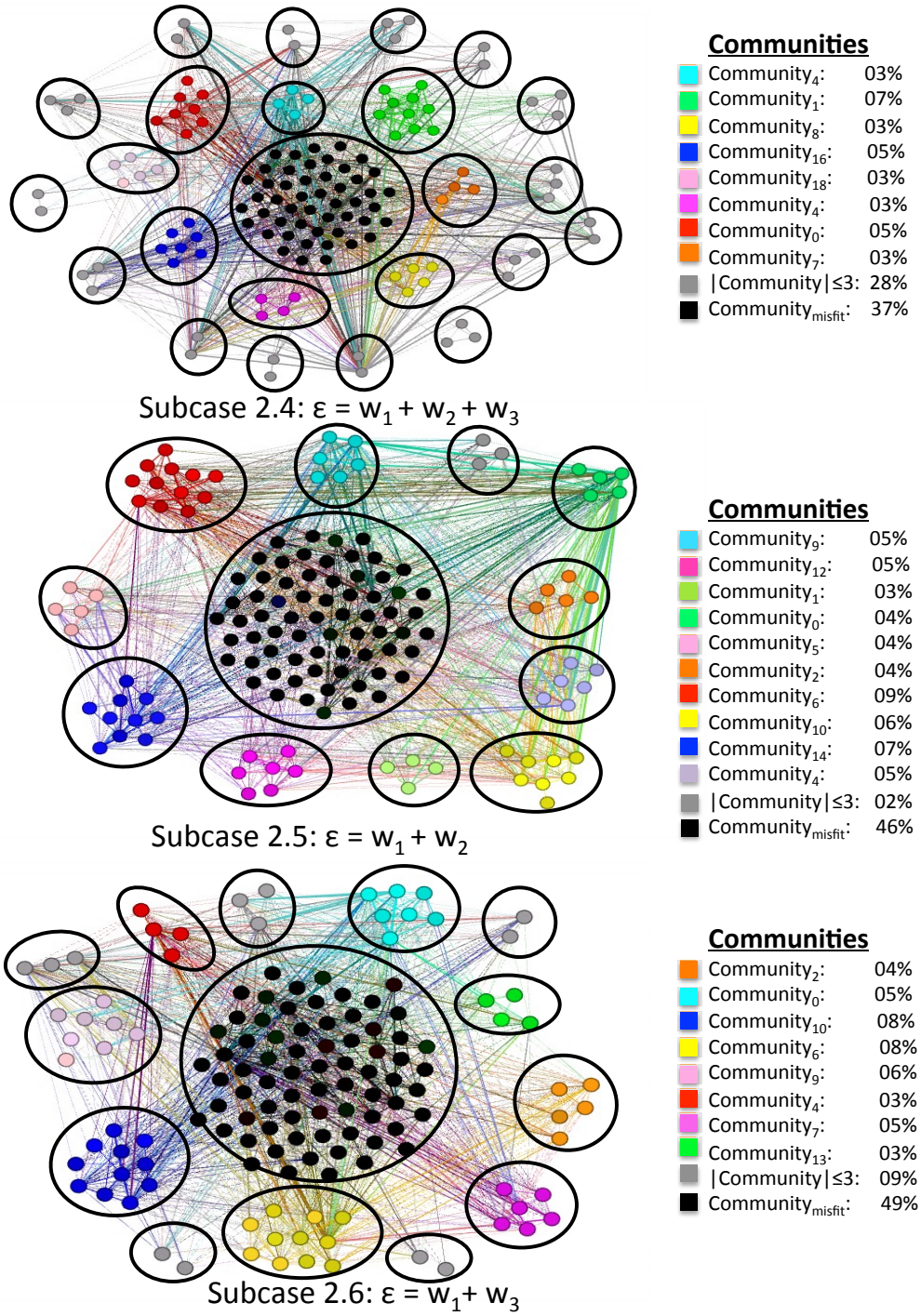
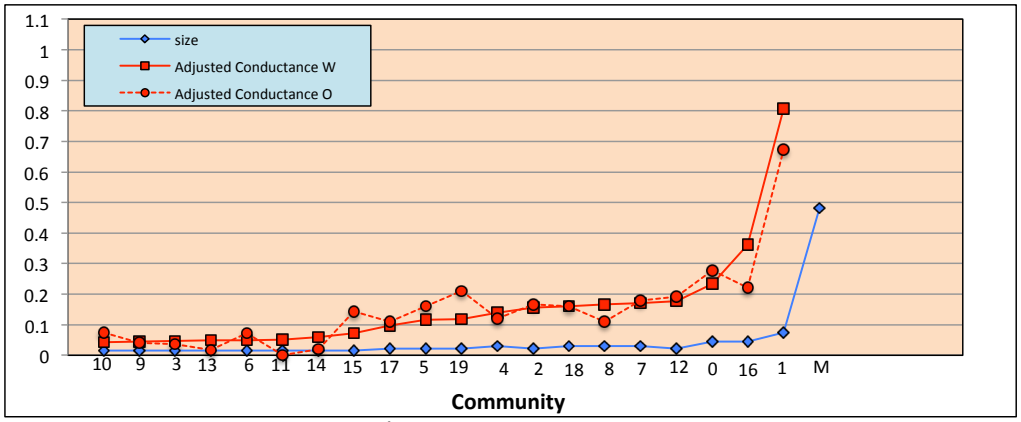
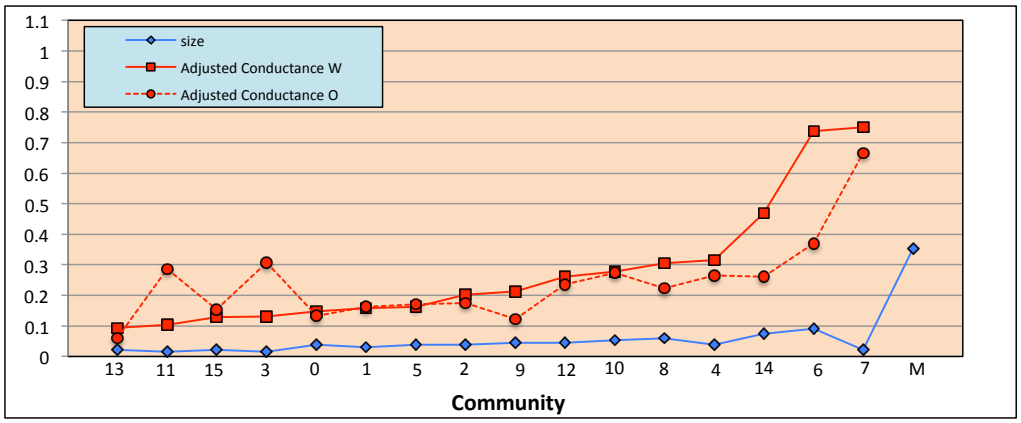


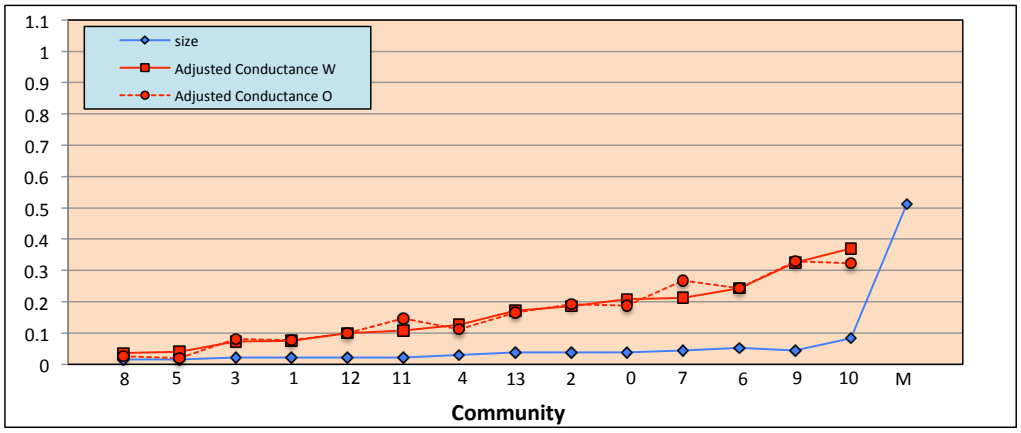

Figure 4.11: Noordin community output plot for subcases 2.4-2.6 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$.




Subcase 2.4: $\varepsilon = w_1 + w_2 + w_3$




Subcase 2.5: $\varepsilon = w_1 + w_2$




Subcase 2.6: $\varepsilon = w_1 + w_3$

Figure 4.12: Noordin community size and normalized conductance for subcases 2.4-2.6 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$.

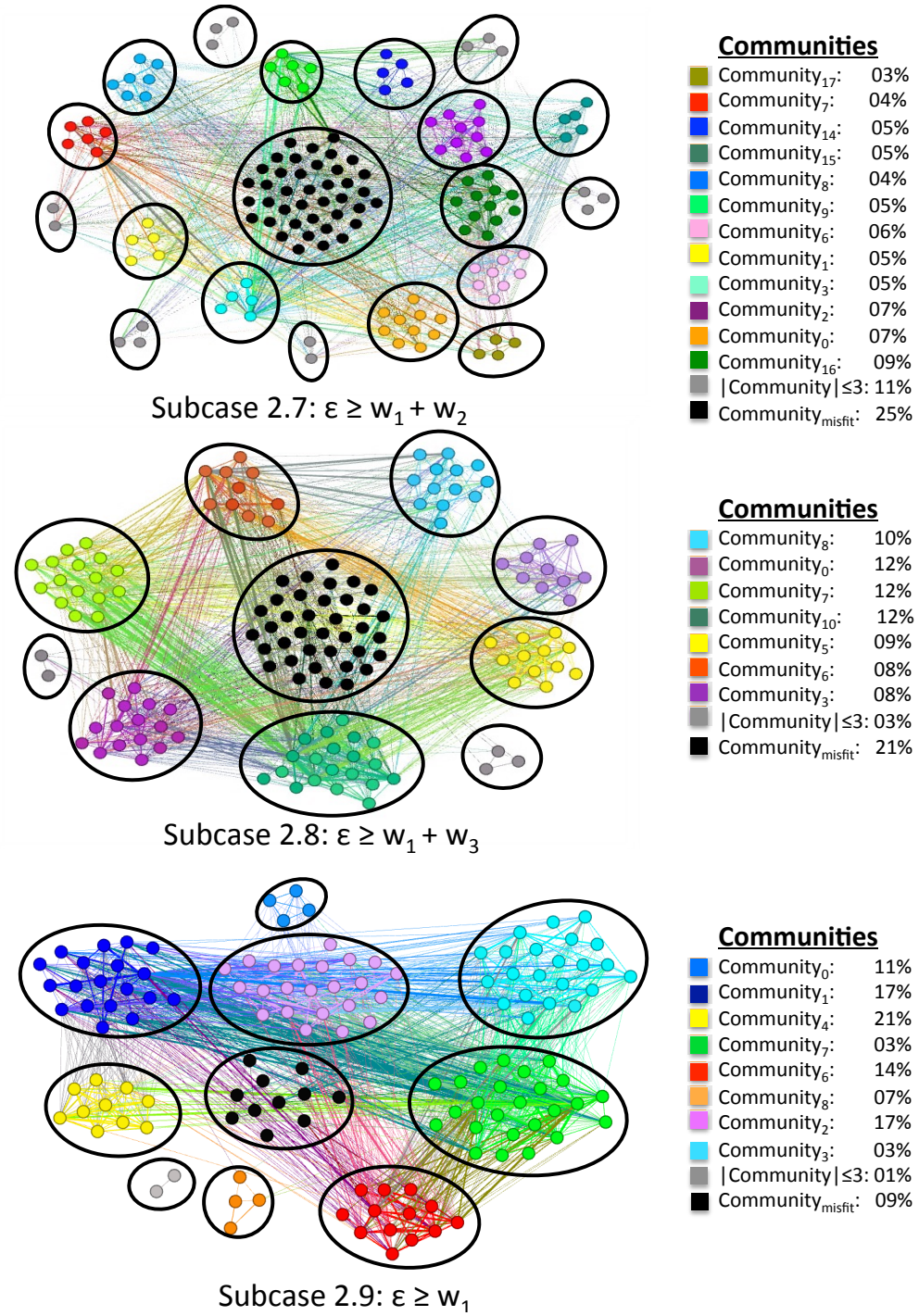
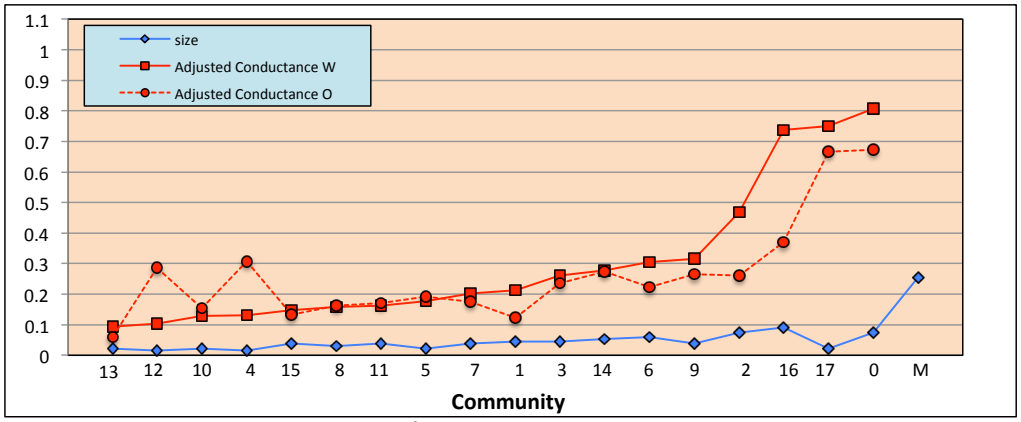
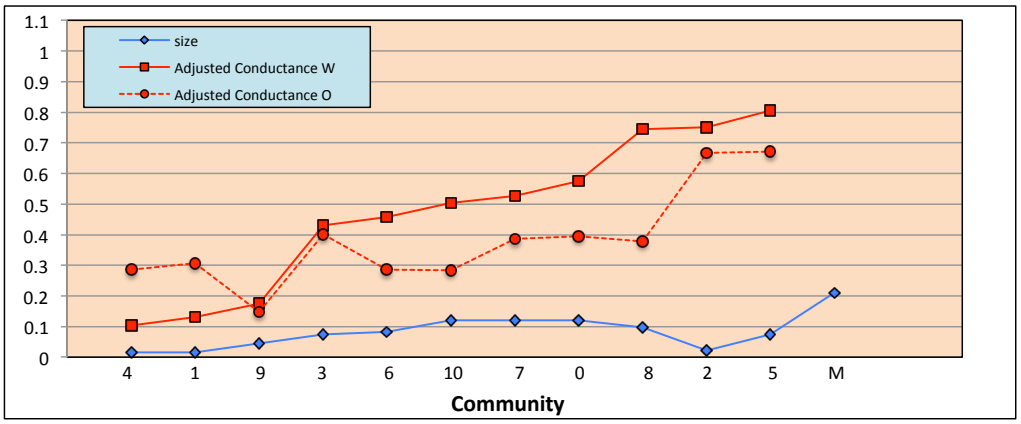


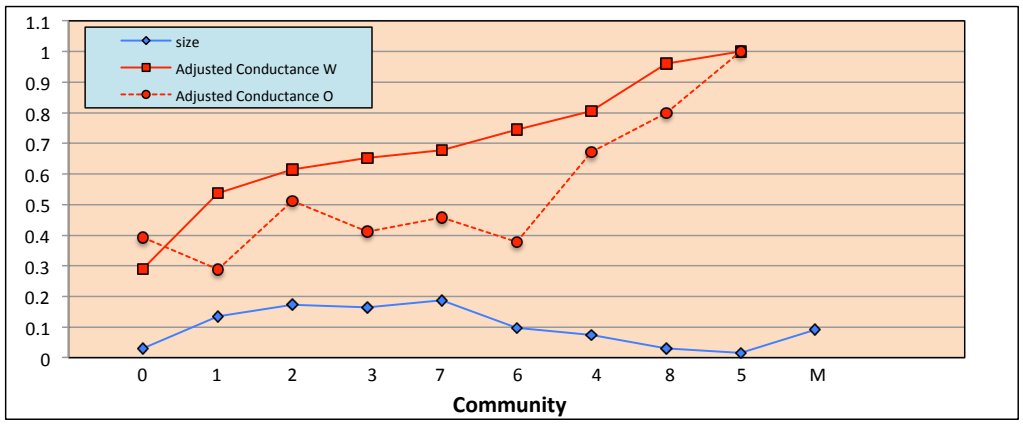

Figure 4.13: Noordin community output plot for subcases 2.7-2.9 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$.




Subcase 2.7: $\varepsilon \geq w_1 + w_2$




Subcase 2.8: $\varepsilon \geq w_1 + w_3$




Subcase 2.9: $\varepsilon \geq w_1$

Figure 4.14: Noordin community size and normalized conductance for subcases 2.7-2.9 with $w_1 = 3$, $w_2 = 2$, and $w_3 = 1$.

Reviewing the results from case 2 allows us to make several observations in comparison to case 1. Similar to case 1, in Figure 4.12 we observe that the $\varepsilon = v$ subcases produce many small communities with relatively low adjusted conductance. Again, subcase 1.4 performed the poorest in terms of adjusted conductance and it also has the highest misfit community. However, we observe that subcases 1.5 and 1.6 produced a higher quantity of smaller communities than in case 1. By imposing an ordering of weights on the categories we have better defined the edges that belong to each category and consequently limited the edges that are included when we build our weighted graphs and apply a threshold value. As in case 1, the adjusted conductance $W$ values typically are higher than the adjusted conducted $O$. We also observe consistency in adjusted conductance $W$ values following the same shape as the adjusted conductance $O$ values. After examining case 2, it appears that subcase 2.2 produces the best communities in terms of adjusted conductance when plotted in both $W$ and $O$.

### 4.2.3 Noordin Results: Case 3

Case 3 examines an ordered distribution of weight values including: $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$. This case is similar to case 2. However, when both the LOC and knowledge categories are included in the subcase, they are collectively still less than the trust category. Consequently, the trust category dominates individual categories and the coalition of other categories. This distribution of weights emphasizes greater importance for the trust category than case 2. We examine the subcase community plots for case 3 in Figures 4.15, 4.17, and 4.7. We follow these plots with the size and conductance per community subcase plots and summary tables in Figures 4.16, 4.18, and 4.20.

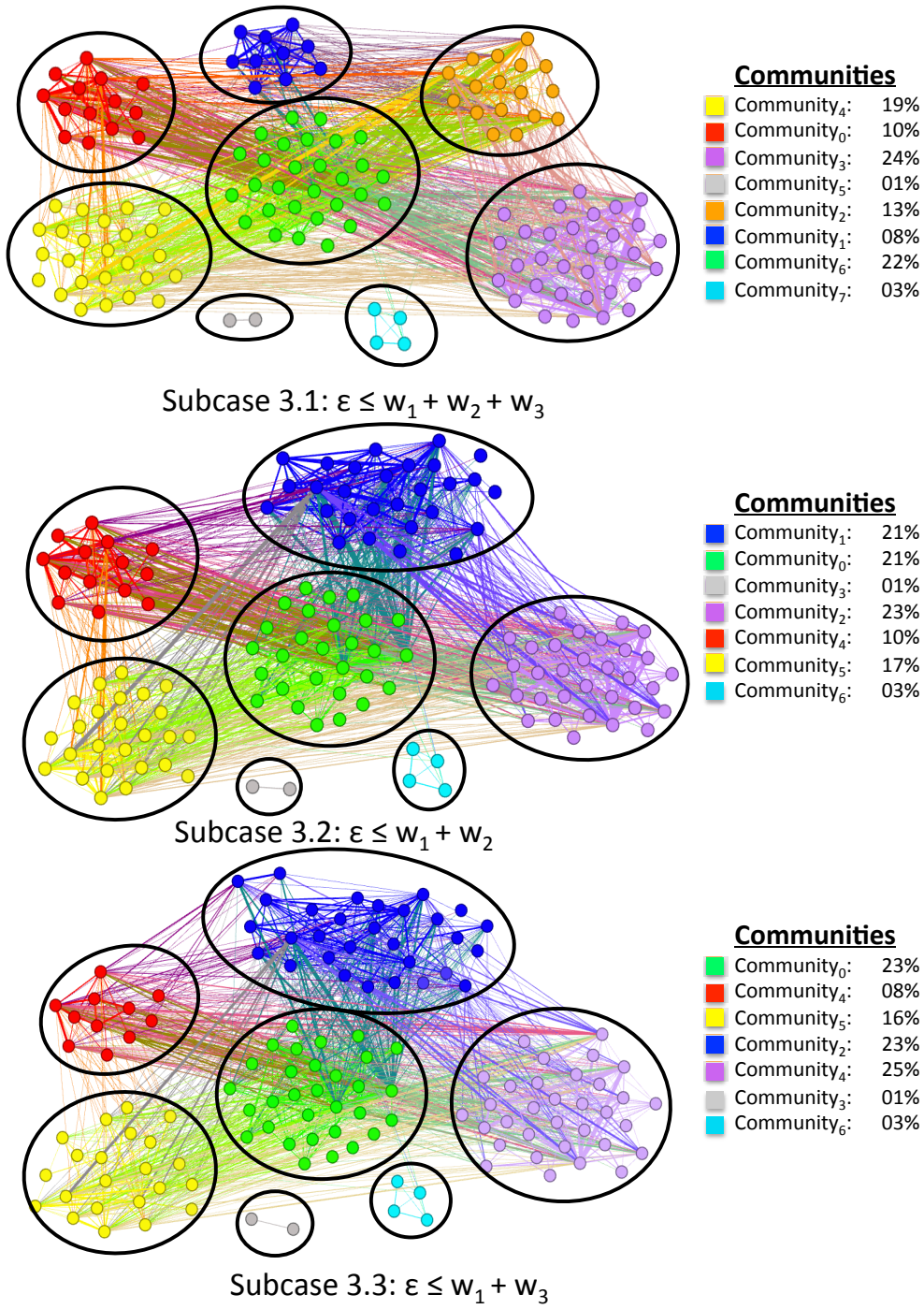
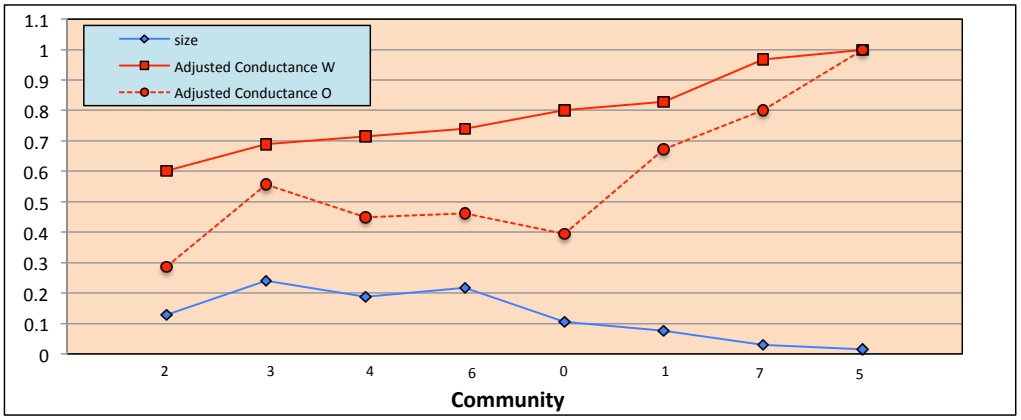
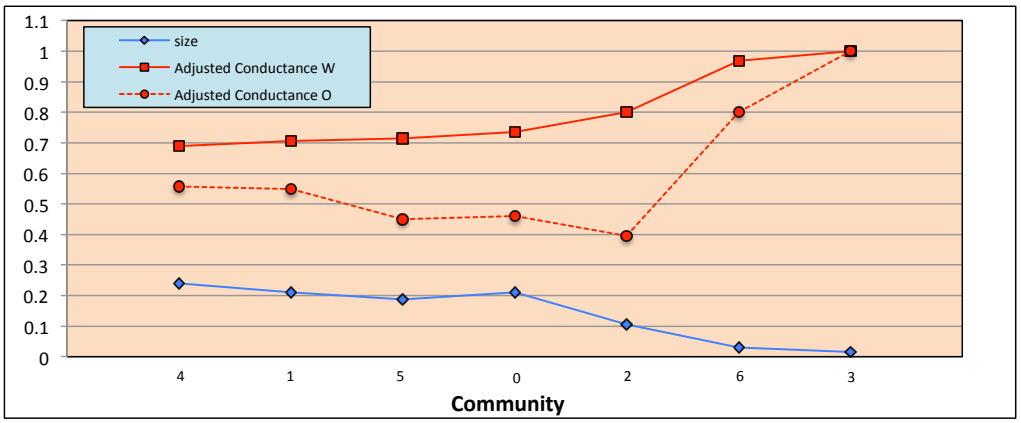


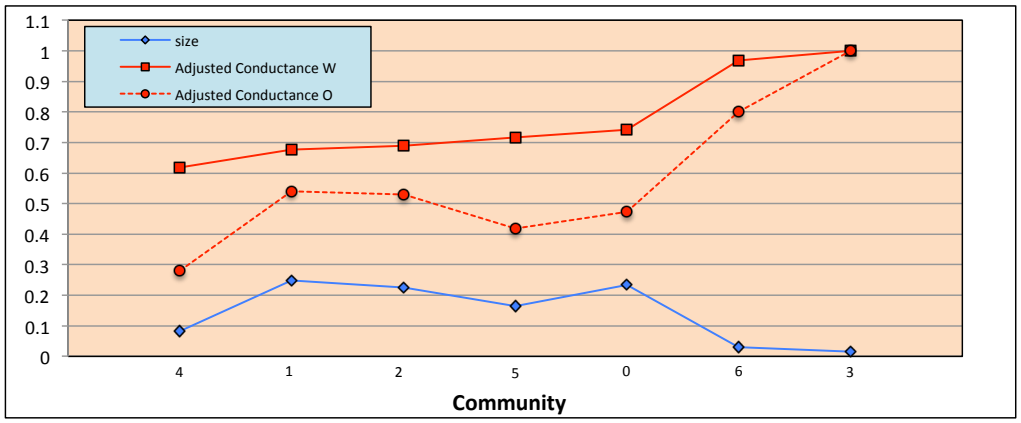

Figure 4.15: Noordin community output plot for subcases 3.1-3.3 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$.




Subcase 2.1: $\varepsilon \leq w_1 + w_2 + w_3$




Subcase 2.2: $\varepsilon \leq w_1 + w_2$




Subcase 2.3: $\varepsilon \leq w_1 + w_3$

Figure 4.16: Noordin community size and normalized conductance for subcases 3.1-3.3 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$.

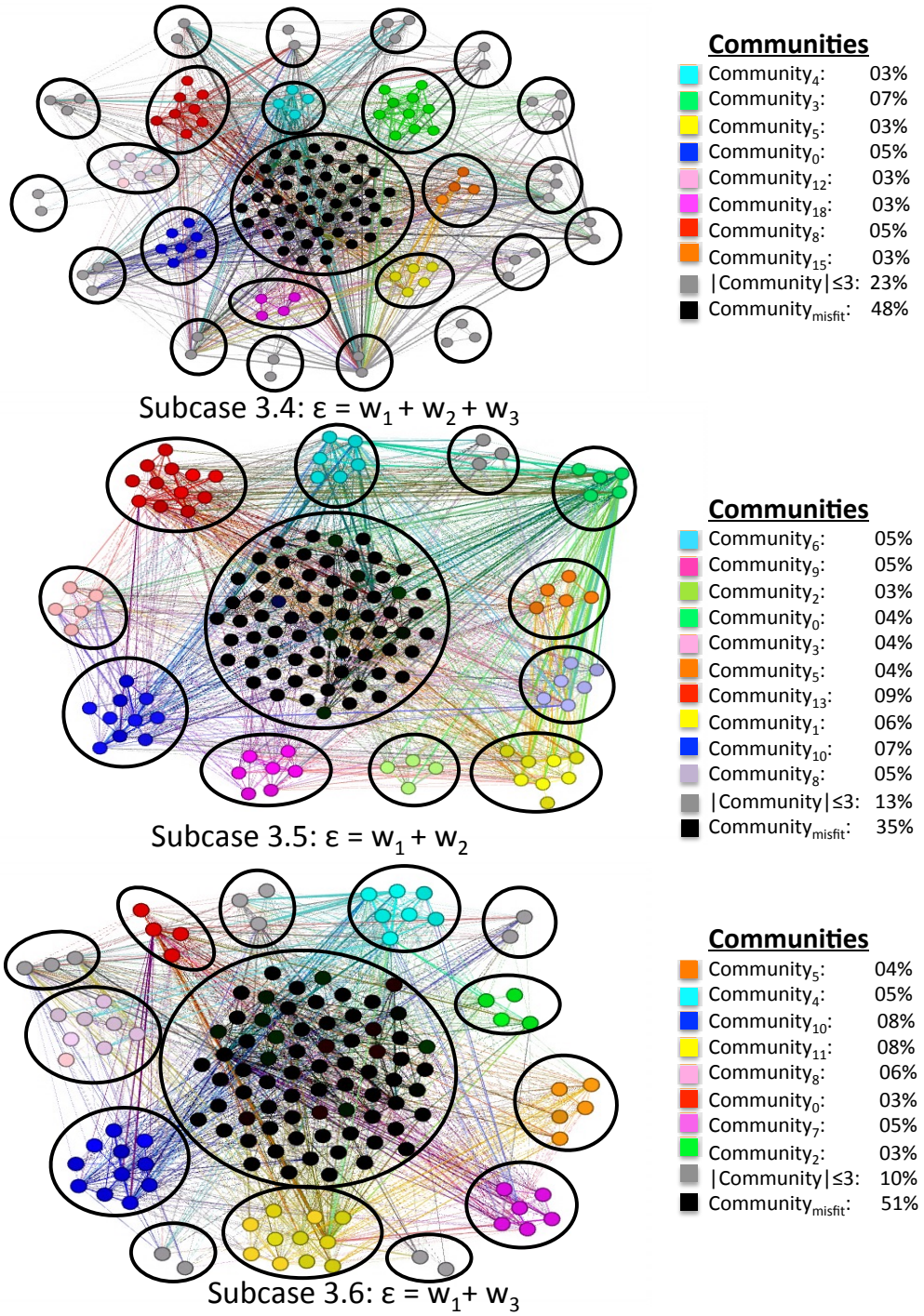
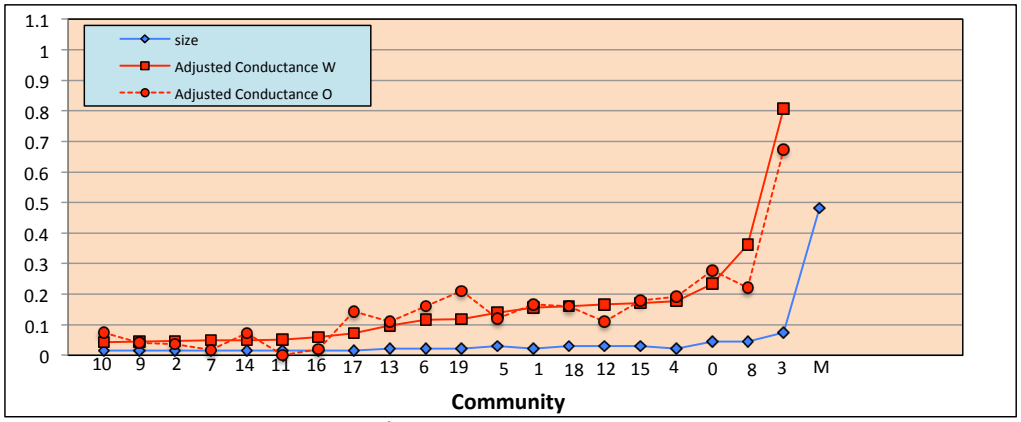
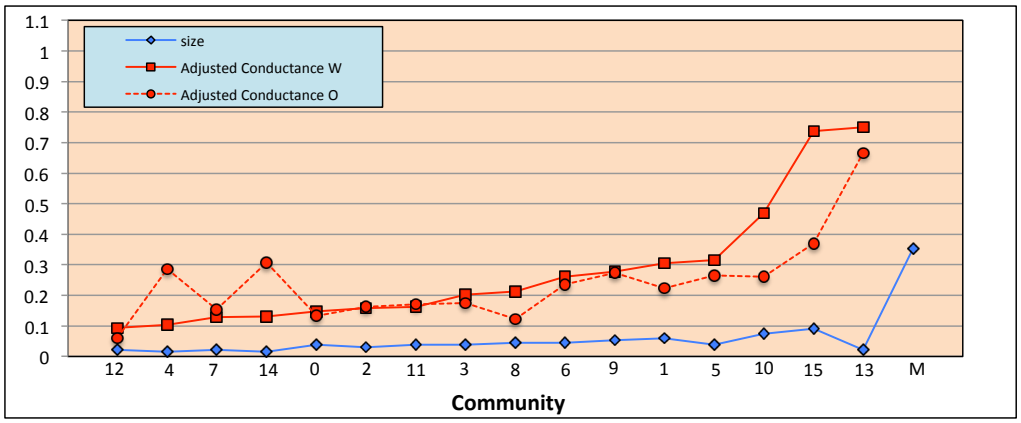


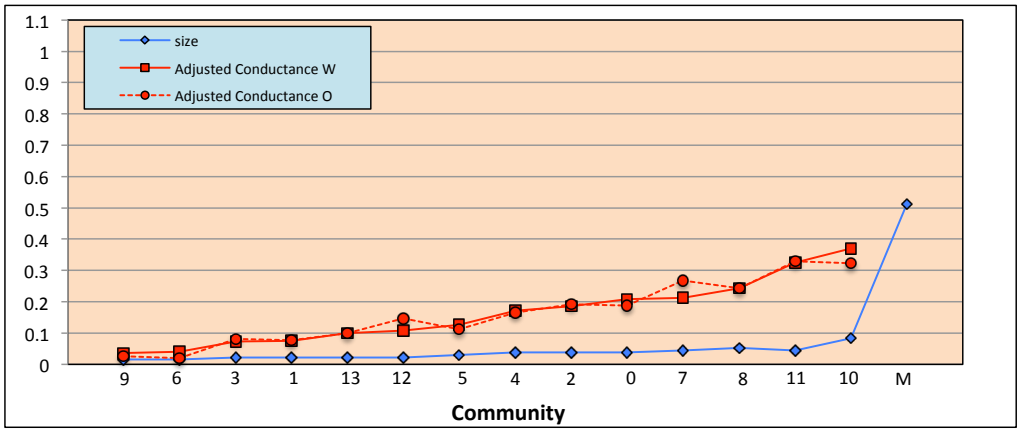

Figure 4.17: Noordin community output plot for subcases 3.4-3.6 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$.




Subcase 3.4: $\varepsilon = w_1 + w_2 + w_3$




Subcase 3.5: $\varepsilon = w_1 + w_2$




Subcase 3.6: $\varepsilon = w_1 + w_3$

Figure 4.18: Noordin community size and normalized conductance for subcases 3.4-3.6 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$.

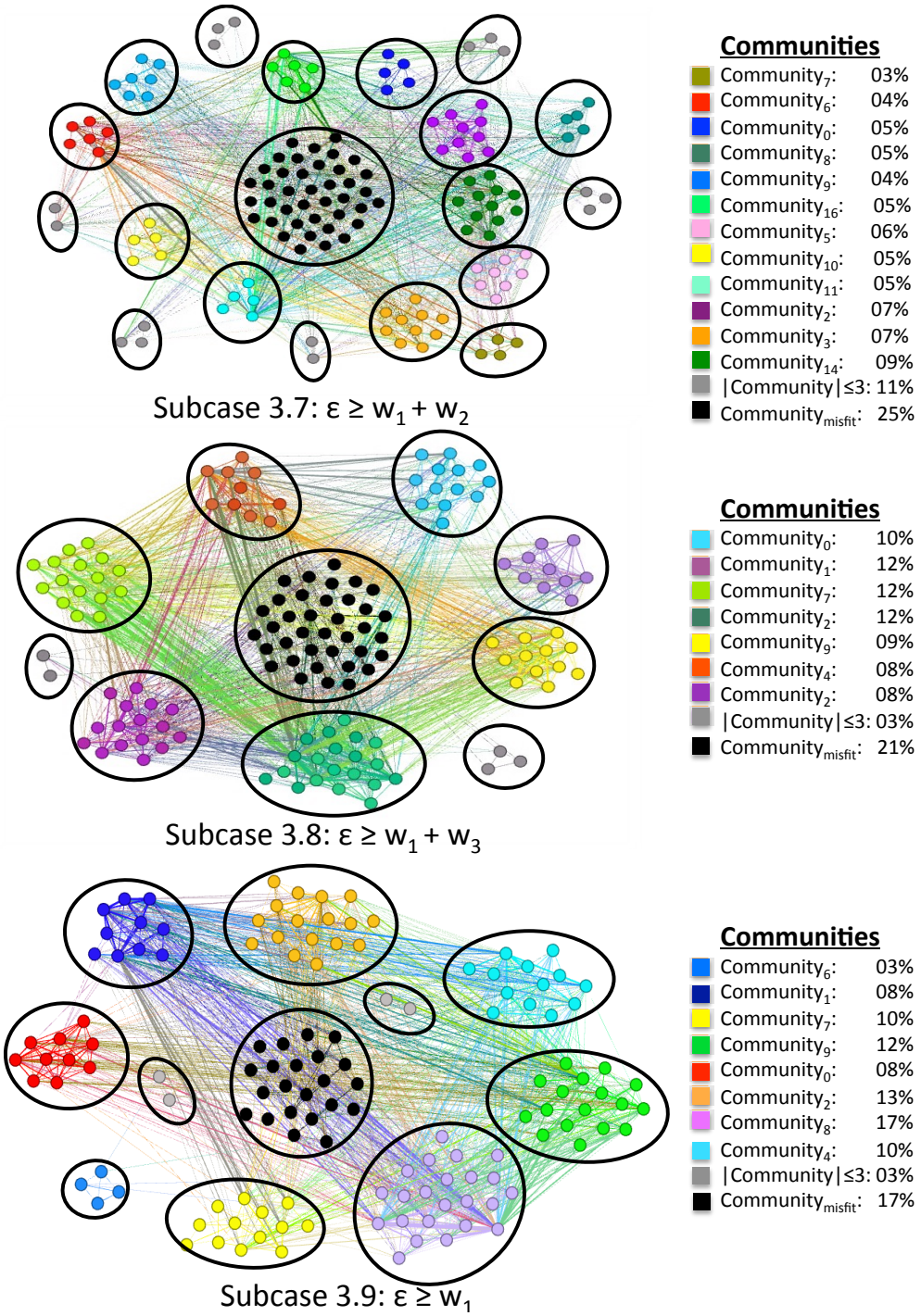
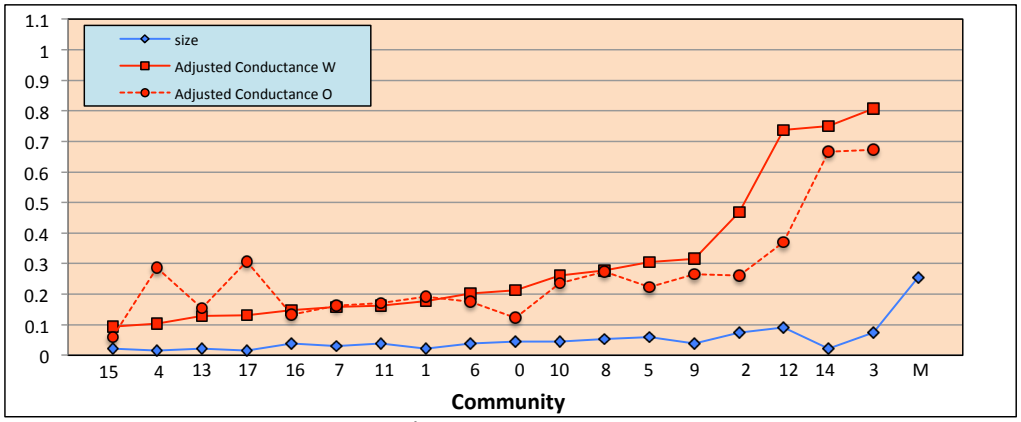
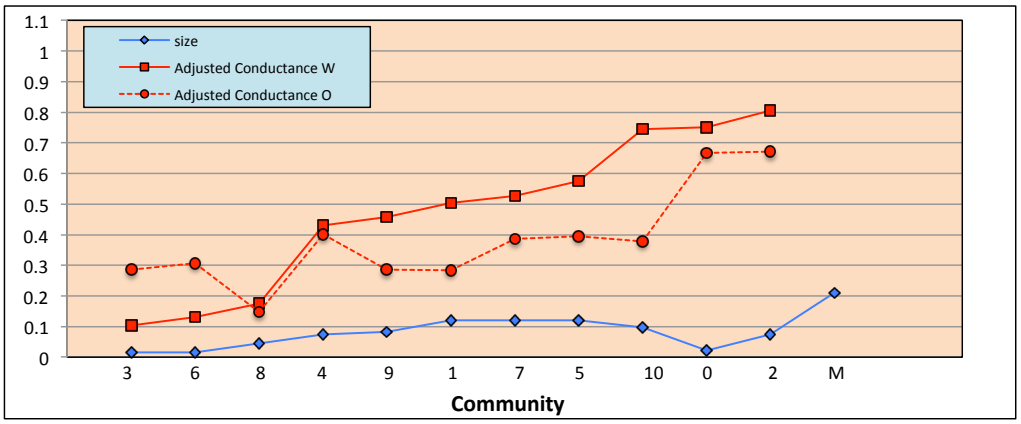


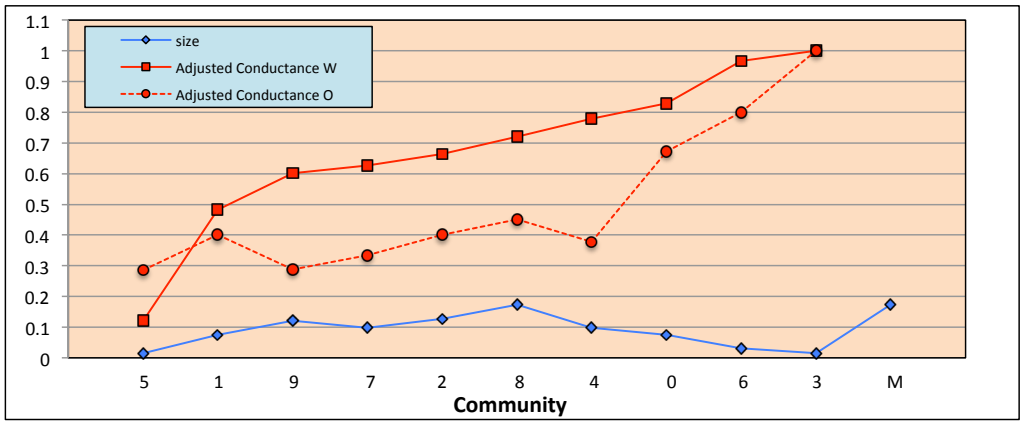

Figure 4.19: Noordin community output plot for subcases 3.7-3.9 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$.




Subcase 3.7: $\varepsilon \geq w_1 + w_2$




Subcase 3.8: $\varepsilon \geq w_1 + w_3$



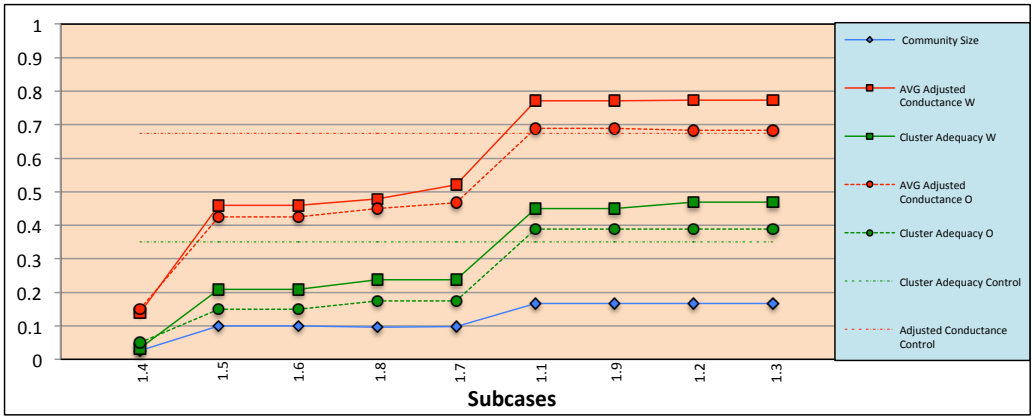

Subcase 3.9: $\varepsilon \geq w_1$

Figure 4.20: Noordin community size and normalized conductance for subcases 3.7-3.9 with $w_1 = 4$, $w_2 = 2$, and $w_3 = 1$.

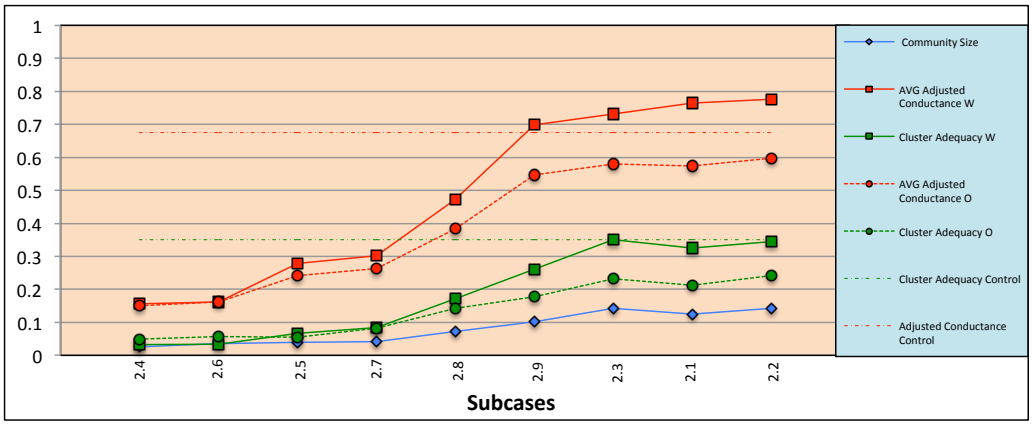
Reviewing the results from case 3 allows us to make several observations in comparison to cases 1 and 2. Case 3 closely mimics the results from case 2. Increasing the importance of the trust category caused only minor changes in the resultant community structure. However, in general, these changes resulted in better adjusted conductance values for case 3. After examining case 3, we determined that subcase 3.2 produces the best communities in terms of adjusted conductance when plotted in both $W$ and $O$. This means that the combination of trust and LOC, trust and knowledge, and each category individually produces topologically better communities than the other combinations of categories.

### 4.2.4 Noordin Observations

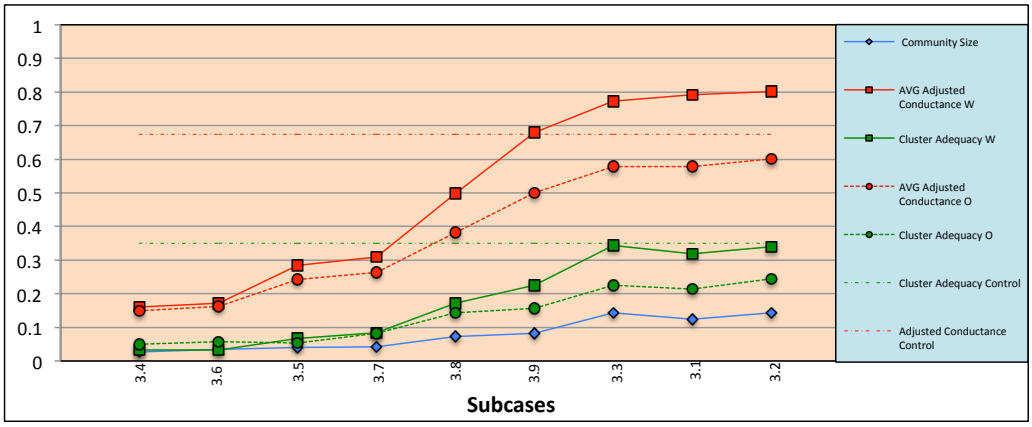
Noordin is the largest of the three terrorist networks according to edge count, with a relatively equal distribution of edges among the three categories. We summarize the results of all three Noordin cases in Figure 4.21. The evidence from the Noordin cases supports the observation that as the average community size increases, the average conductance and cluster adequacy increases. We also consistently observed that $\varepsilon = w_1 + w_2 + w_3$ produced a high volume of small and qualitatively poor communities. In general, the $\varepsilon = v$ threshold choices produced the poorest quality communities. On average, the more relaxed the threshold choice, the better quality the community according to adjjusted conductance and cluster adequacy. In terms of weight cases, we notice that case 1 reveals that an equal distribution of weights actually produces, on average, high quality communities. However, it is important to remember that many subcases from case 1 are redundant, thus we expect many subcases to follow the same trend for case 1. For subcases 1.1, 1.2, 1.3, and 1.9, the adjusted conductance and the cluster adequacy plotted in both the weighted graph and the original monoplex out-preforms the control case. Recall that the control case also represents an equal distribution in weights amongst all of the different layers. Thus, our results from case 1 indicate that we can increase community quality by employing our methodology. We also notice that case 3 slightly outperforms case 2. The highest adjusted conductance value when plotted in the weighted graph results from subcase 3.2. This suggests that heavily weighting a particular category, such as trust, potentially produces better quality communities. We continue to monitor these observed trends in the Boko Harm and FARC results to see if a pattern develops in general for dark networks. We offer an explanation for these trends in Section 4.5.




Case 1: $w_1$= 1, $w_2$ = 1, $w_3$= 1




Case 2: $w_1$= 3, $w_2$ = 2, $w_3$= 1




Case 3: $w_1$= 4, $w_2$ = 2, $w_3$= 1

Figure 4.21: Average community size, average normalized conductance, and cluster adequacy from communities plotted in $O$ and $W$ for Noordin cases 1-3.

## 4.3 Boko Haram Results and Analysis

In this section we display the results and analysis of applying our methodology to the Boko Haram Network. First we display our control case in Figure 4.22 and our results summary case plots in Figure 4.23. We follow these plots with our observations for the Boko Haram Network in Section 4.3.1.

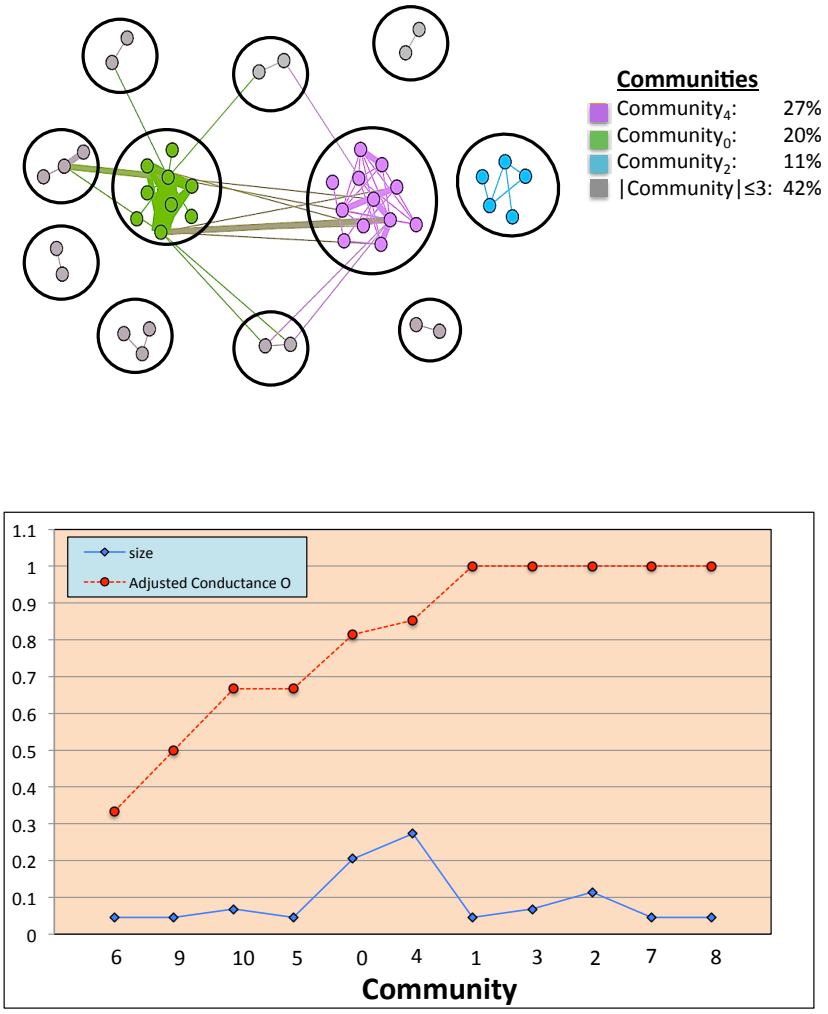
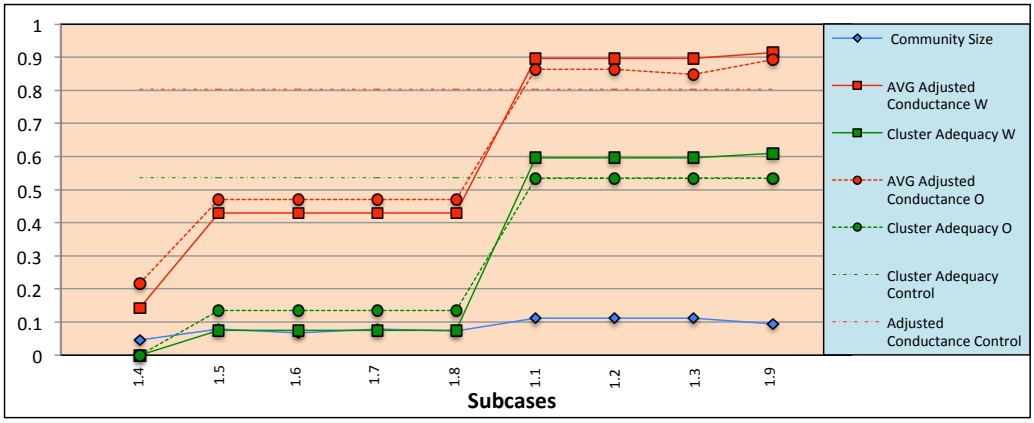
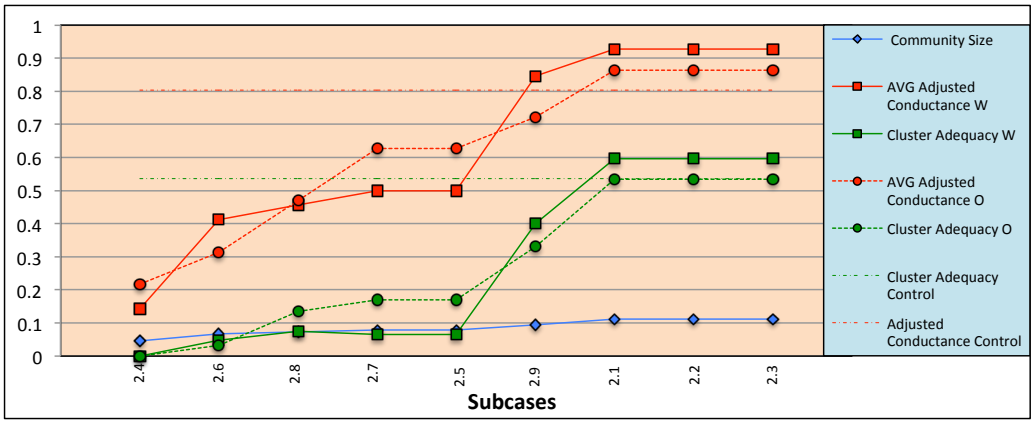


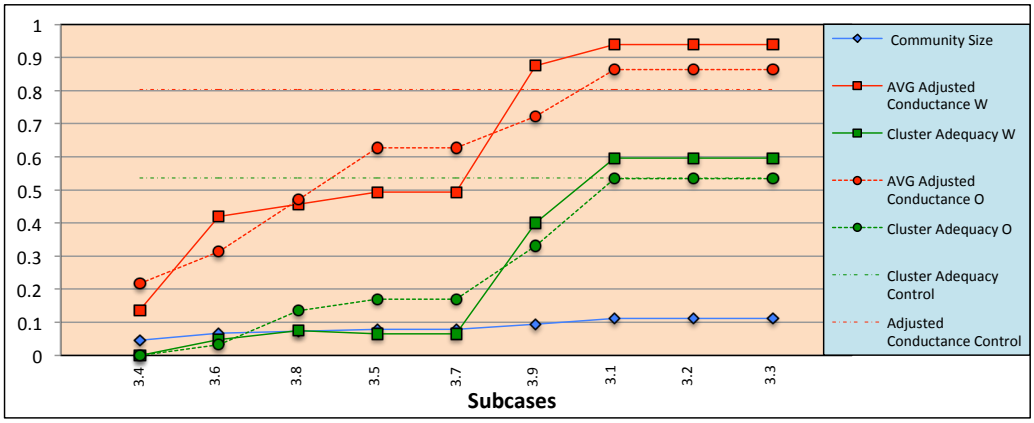

Figure 4.22: Community size and adjusted conductance for Boko Haram control case.




Case 1: $w_1$= 1, $w_2$ = 1, $w_3$= 1




Case 2: $w_1$= 3, $w_2$ = 2, $w_3$= 1




Case 3: $w_1$= 4, $w_2$ = 2, $w_3$= 1

Figure 4.23: Average community size, average adjusted conductance, and cluster adequacy for Boko Haram cases 1-3.

### 4.3.1 Boko Haram Observations

The Boko Haram Network is much sparser and more disconnected than the Noordin Network. However, each category contains a relatively equal amount of edges. Notice that the control case is already partitioned due to the high number of components and light external connectivity between communities. As a consequence, we do not expect large improvements to community quality by applying our methodology. The control case in Figure 4.22 reveals 11 communities. Three of these communities are larger and the remaining eight communities are of size three or smaller. We observe similar behavior to Noordin for communities that are also components for perfect adjusted conductance. In the largest connected component, $community_6$ and $community_9$ have the worst conductance since there have a high number of external connections relative to the internal connections within their respective communities. For example, $community_6$ has only one connection inside the community, but has four connections outside the community $community_4$ and $community_0$. Since $community_6$ has poor adjusted conductance, placing these vertices in the misfit community has the potential for increasing the average quality of the remaining communities.

In Figure 4.23 we observe similar trends as the Noordin Network. We continue to observe that $\varepsilon = w_1 + w_2 + w_3$ provides the poorest quality communities as the intersection of the communities in all three categories. The $\varepsilon \leq$ cases continue to generally produce the best quality communities. However, we observe more variance in the ordering of the subcases between the cases. The general trend of community quality increasing with average community size continues for Boko Haram.

## 4.4 FARC Results and Analysis

In this section we display the results and analysis of applying our methodology to the FARC Network. First we display our control case in Figure 4.24 and our results summary case plots in Figure 4.25. We follow these plots with our observations for the Boko Haram Network in Section 4.4.1.

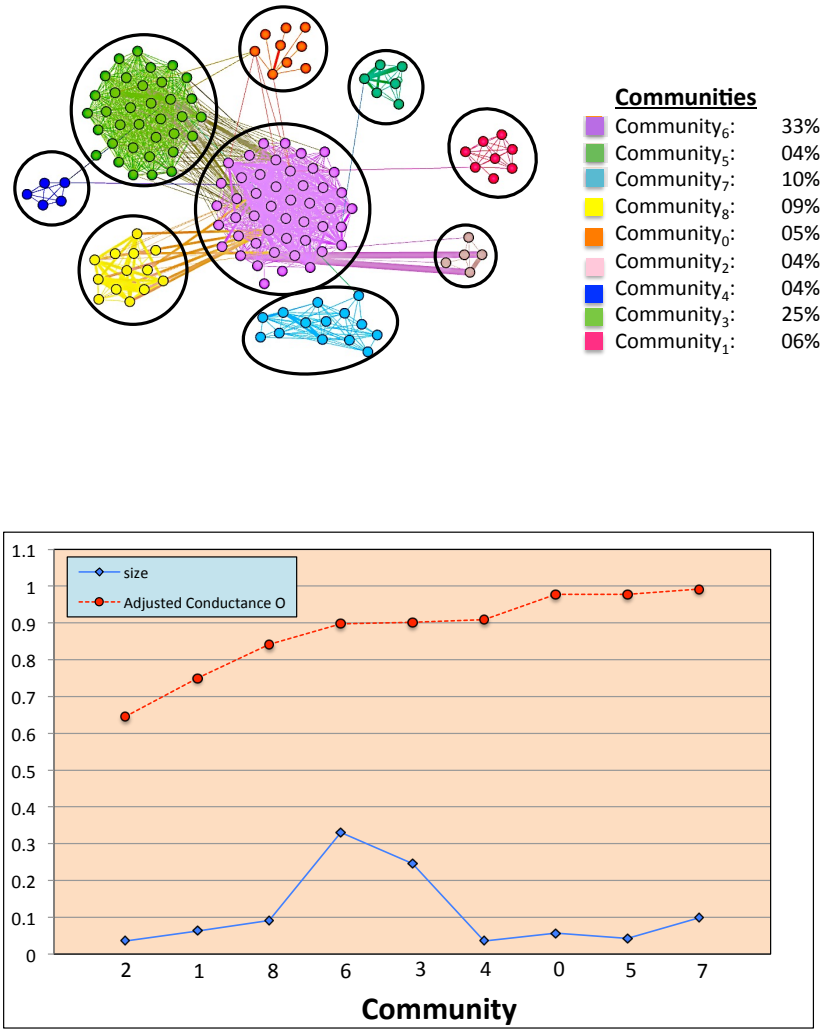
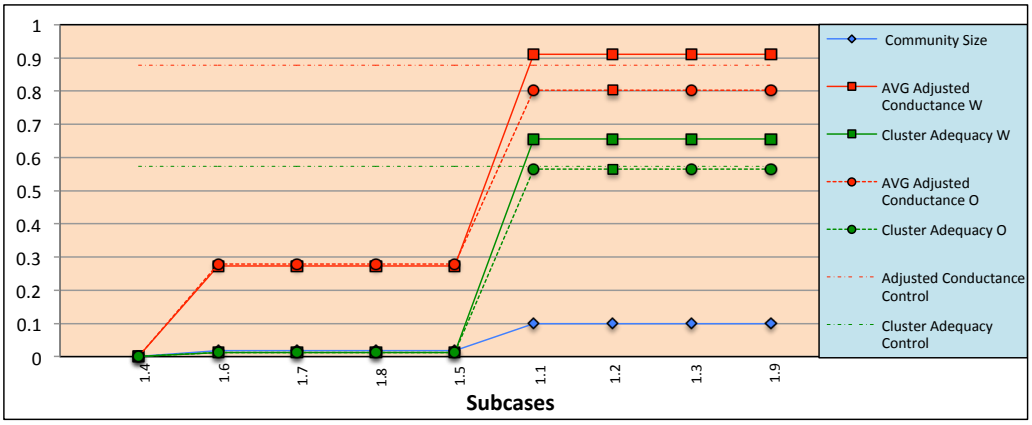
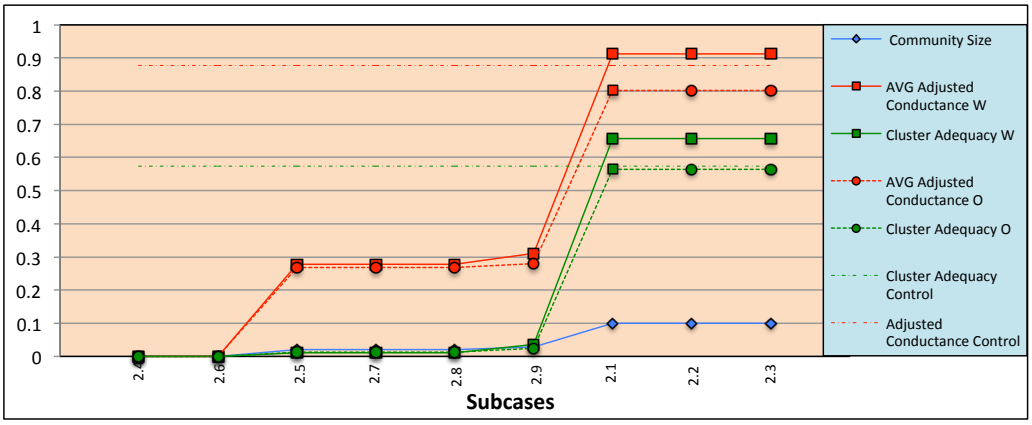


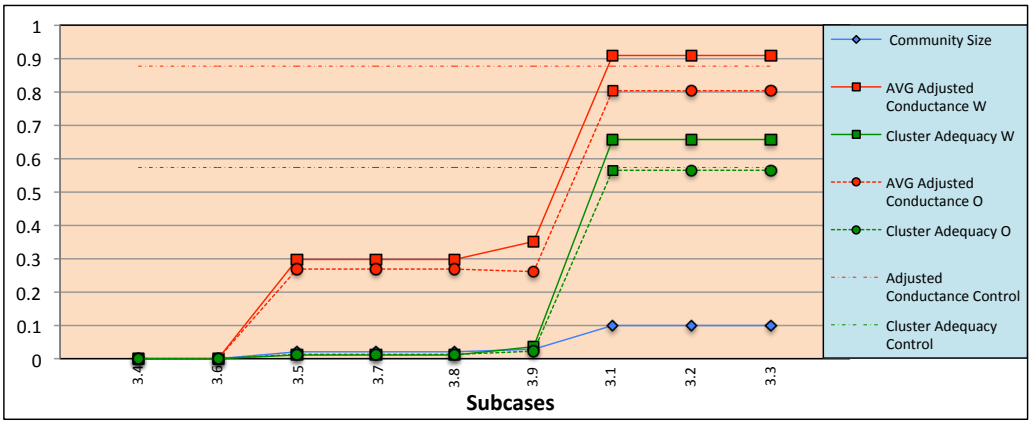

Figure 4.24: Community size and adjusted conductance for FARC control case.




Case 1: $w_1$= 1, $w_2$ = 1, $w_3$= 1




Case 2: $w_1$= 3, $w_2$ = 2, $w_3$= 1




Case 3: $w_1$= 4, $w_2$ = 2, $w_3$= 1

Figure 4.25: Average community size, average adjusted conductance, and cluster adequacy for FARC cases 1-3.

### 4.4.1 FARC Observations

The FARC Network is dominated in edge distribution by the LOC category. This category accounts for more than 98% of the edges. The clear hierarchy between organizations is visible in the control case in Figure 4.24. Notice that the control case produces very high quality communities. There are a total of nine communities. Notice that $community_7$ is dense internally and only has one external connection. This structure allows $community_7$ to have excellent adjusted conductance. These near component communities follow the similar trends by component communities found in Noordin and Boko Haram.

In Figure 4.25 we observe that some threshold cases did not result in communities. This is a product of the edge distribution in the three categories. Since there are no communities in $\varepsilon = w_1 + w_2 + w_3$, this means that the same edge relationship does not exist in all three categories. We also observe a much more dramatic shift in the quality of communities as we transition from $\varepsilon \geq$ subcases to $\varepsilon \leq$ subcases. The domination of the LOC category makes it difficult to produce quality communities when LOC is not included.

## 4.5 General Observations

Through our analysis of the three terrorist networks we identified the following common observations. The two different community quality metrics were relatively consistent in their evaluation of each subcase. This consistency indicates that the metrics could provide substantial evidence in determining the quality of the communities in absence of ground truth. For average adjusted conductance, we identified the community strength relative to the remainder of the network. For modularity, we evaluated community strength relative to the expected connections in a random graph. We further refined modularity using cluster adequacy to determine the strength of the graph relative to the best possible modularity based on partitioning the network into $m$ communities. The consistency in these metrics is critical to our analysis of identifying the best quality communities from the subcases.

The control case cluster adequacy and adjusted conductance values were typically very high when compared to the other subcases, yet several subcases still performed better. Recall that the control case represents aggregating all of the information into a single weighted graph, which results in the loss of detailed information that is inherent to each layer. Consequently, a connection in the control graph means two vertices are related, but we no longer have

the available information to distinguish how they are related. Our subcases methodically aggregate layers that are similar in meaning to manage the information loss problem of aggregating the entire network. Choosing a weight case and threshold subcase allows the user to determine which categories to include, and how important they are to the analytical goals. For example, small communities may not be the best quality, but they may be easier to target. While not the focus here, Chapter 5 and 6 discuss this possibility in greater depth.

Generally, as the average size of the communities increased, the adjusted average conductance and the cluster adequacy values increased as well. This indicates that fewer communities of larger size is more optimal for the dark networks we have studied. However, as discussed previously in section 4.2.2, the size of the community becomes irrelevant if the community is a component of the network. Communities that are also characterized as components have no external edges to the community, which results in perfect adjusted conductance.

Our hypothesis that $\varepsilon = v$ would produce many small and poor quality communities was supported by the results of all three networks based on the two metrics used. Placing this high restriction on the community development forced the communities to remain small. Embedding these small communities in the weighted graph and the original monoplex revealed their poor quality. The external connections that were ignored during the community development process of applying thresholds become very important in determining community quality. For example, notice that $community_0$ from Noordin subcase 3.6 in Figure 4.17 is a small 4-vertex community. At most, this community can have six internal connections. Yet, when these community members are plotted back into the monoplex, we notice a large number of external connections to other communities, including the misfits. As a result, both adjusted conductance and cluster adequacy rank this community as poor in quality in Figure 4.18.

In most cases, the subcases with the highest average adjusted conductance and cluster adequacy came from the threshold choice of $\varepsilon \leq v$, which was the most relaxed of all of the thresholds and did not contain any misfit vertices. This is surprising considering our initial claim that threshold choices of $\varepsilon \geq v$ would produce the best quality communities. Community quality instead increased as the restriction of the threshold cases was relaxed. It is important to note that under these relaxed conditions, every vertex was assigned to a

community and no vertices were labelled misfits. We assumed that by placing vertices in a misfit group, the overall quality of the community would increase. These misfits are still connected in the graph when our cluster adequacy and adjusted conductance are calculated. Boko Haram case 1.9 demonstrated that physically removing misfit vertices from the graph does have the potential to increase the quality of the communities. In this subcase, the vertices in $community_6$ from Figure 4.22 were re-designated as misfits, which removed it from consideration in averaging the adjusted conductance values of the communities.

In this chapter, we have identified the best quality communities based on average adjusted conductance and cluster adequacy, as we had no ground truth community. Based on these metrics, case 1 generally performed the best, yet the best overall subcase for Noordin was subcase 3.2. However, we return to our definition of KSC to verify that we have successfully identified the best possible communities for network disruption. Our intuition still points to subcase 3.9 because it provides the necessary bias for the trust foundation category to dominant the remaining community information from the other categories. At this point, we are unable to determine if the best quality community is indicative of a meaningful community as described by the KSC. To explore this, we model the Noordin Network in Chapter 5 as a network flow problem and examine the community properties in detail for subcase 3.9 to build community targeting profiles to disrupt the network.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 5:
# Modeling and Application

In this chapter we develop an optimization model for the Noordin Network using Pyomo [56], which is a python software package from Sandia Labs. We begin with some background information and an explanation of our network flow model formulation. Next, we present the optimal attack results for network disruption. We then examine the community properties from subcase 3.9 to refine our disruption strategy and compare performance results against the control attack plan.

## 5.1 Model Formulation

According to Ahuja et al. [57], minimum cost network flow problems involve some type of resource or commodity that exists as a supply for some set of vertices and as a demand for another set of vertices. The objective of these problems is to transport commodities from supply to demand in the most efficient manner possible without violating a given set of constraints. There are a myriad of different approaches to transforming a network into a network flow problem. According to Carlyle [58], one technique for optimizing network flow involves building and interdicting the shortest path algorithm. He explains that the shortest path algorithm calculates the shortest distance between a source or set of source vertices and a destination or set of destination vertices. Alderson et al. [59] represent the shortest path formulation between a designated initial start vertex, $s$, and a terminal vertex, $t$, as a linear program using the following objective function and constraint equations:

$$\min_{x} \sum_{(i,j)\in A} c_{ij} X_{ij}, \tag{5.1}$$

where $c_{ij}$ is the cost assigned to the edge between vertex $i$ and $j$ and $X_{ij}$ is a binary variable, which represents flow along edge $(i, j)$. $X_{ij}$ is equal to one if the edge $(i, j)$ is on the shortest path and is equal to zero otherwise.

Equation 5.1 essentially means we are adding up the length or cost of all of the edges along the shortest path to produce a total cost amount. The objective of this linear program is to

minimize the total cost to flow or transport a commodity or resource from $s$ to $t$. Alderson et al. [59] invoke the following constraint equations on the objective function:

$$\sum_{j:(i,j)\in A} X_{ij} - \sum_{j:(j,i)\in A} X_{ji} = \begin{cases} 1 & \text{if } i = s \\ 0 & \text{if } i \neq s,t \; \forall i \in N \\ -1 & \text{if } i = t, \end{cases} \tag{5.2}$$

$$X_{ij} \geq 0, \tag{5.3}$$

where $X_{ij}$ is the flow into the vertex $j$ and $X_{ji}$ is the flow out of the vertex $j$.

These constraints ensure the flow at each vertex in the network is properly balanced and that all flow is non-negative. To interdict the network, Alderson et al. [59] define a new set of data and variables to represent the attacker problem as:

$$\max_{Y} \min_{X} \sum_{(i,j)\in A} (c_{ij} + q_{ij} Y_{ij}) X_{ij}, \tag{5.4}$$

where $q_{ij}$ is the associated penalty for attacking the edge $(i, j)$ and $Y_{ij}$ is a binary variable that equals one when edge $(i, j)$ is attacked, and zero otherwise.

Alderson et al. [59] point out that there are two competing objectives represented in Equation 5.4. The network defender continues to desire the shortest possible path in order to minimize cost. Conversely, the network attacker's goal is to maximize the length of the shortest path to maximize the cost and consequently, the damage to the network. Alderson et al. [59] define the following additional constraints for the attacker problem:

$$\sum_{(i,j)\in A} Y_{ij} \leq max_num_attacks, \tag{5.5}$$

$$Y_{ij} \in \{0, 1\}. \tag{5.6}$$

These constraints ensure the optimal number of attacks is chosen to be equal to or less than a specified number by the attacker. Alderson et al. reveal that a problem that

simultaneously maximizes and minimizes the objective function cannot be directly solved as a linear program. To remedy this difficulty, they explains that for a fixed attack plan, the minimization problem can be transformed into the dual maximization problem.

According to Ahuja et al. [57], the max-flow min-cut theorem states that:

> the maximum value of the flow from $s$ to $t$ equals the minimum capacity of all $s - t$ cuts.

Since an optimal solution exists for our original primal problem, the theorem of strong duality states that an optimal solution for the dual problem must exist as well. Alderson et al. exploit this property by transforming the primal attacker problem into a dual integer linear program where the objective function is represented as:

$$\max_{\pi, Y} \pi_s - \pi_t, \tag{5.7}$$

where $\pi_s$ and $\pi_t$ represent a relative distance between $s$ and $t$.

The dual objective function allows us to maximize the distance between $s$ and $t$. This relative distance increases when edges are attacked. For a detailed explanation on how to build the dual problem from the existing primal problem see Brown et al. [60]. After some simplification, Brown et al. represent the attacker dual constraints as:

$$\pi_i - \pi_j - q_{ij} Y_{ij} \leq c_{ij} \; \forall (i, j) \in A, \tag{5.8}$$

$$\sum_{(i,j) \in A} Y_{ij} \leq max_num_attacks, \tag{5.9}$$

$$Y_{ij} \in \{0, 1\}, \tag{5.10}$$

$$\pi_i \text{ unrestricted}, \tag{5.11}$$

$$\pi_s \equiv 0. \tag{5.12}$$

Alderson et al. [59] explain that formulating the shortest path interdiction problem results in a linear optimization problem that seeks to increase the length of the shortest path from the supply vertices to the demand vertices. In the next section, we apply the shortest path interdiction dual formulation to the Noordin Network.

## 5.2 Noordin Formulation

We developed a scenario that exploits Noordin's documented success in coordinating between the five major terrorist organization in Indonesia. Under this scenario, Noordin is planning a joint attack with the support of the major terrorist organizations in Indonesia. To model this coordination, Noordin represents our start vertex with a commodity supply of information. His objective is to optimize the flow of information to the key tactical leaders from Darul Islam, KOMPAK, Jemaah Islamiyah, and Ring Banten Group. The corresponding set of destination vertices, $D$, from these terrorist groups is:

$$D = \{\text{'Kang Jaja', 'Aris Munandar', 'Ali Imron', 'Iwan Dharmawan'}\}. \tag{5.13}$$

The attacker in this case is represented by our established user in Chapter 3, JIEDDO. The attacker objective is interrupt the flow of information by attacking the optimal combination of edges that maximize the sum of the lengths of the shortest paths from Noordin to the elements in the set $D$. Here, the cost of each edge is represented as increments of time in hours. For example, a path that is 24 units long corresponds to a message delay of 24 hours or one day. Attacking an edge corresponds to less invasive actions such as jamming cell phone capabilities, which, given time, can be overcome by actions such physically meeting with the person. However, this time delay has the potential to disrupt the coordination of a planned attack. The edges used to build this model are extracted from the LOC and Trust categories as described in Table 3.4.

It is important to note that the LOC category only includes 120 of the total 133 terrorists and 318 of the 2451 total connections. Here we extend Krebs idea of the trust relationship importance to improve the realism of our model and increase the number of terrorists and available connections to 128 and 598 respectively. Another aspect we must consider is that network flow problems require directed networks. To model this requirement in the Noordin undirected data set, we create parallel edges, in the opposite direction, for each

pair of vertices. This results in doubling the number of available edges to 1196.

Given the importance of trust, it is feasible to infer additional communication edges using existing trust relationships. For example, two terrorists who were friends in school, may not have any documented terrorist activity communication between them. However, if an existing point of contact is unable to be reached, the terrorist can potentially activate a dormant relationship, such as trust, to reestablish communication. Using this theory, we can enhance our network model's resilience and robustness capability using the trust category edges. If an edge between two vertices is attacked, then the vertices have the option of formulating new communication channels using the trust category edges. By rewarding redundant connections between terrorists, our model reflects choosing the path with more familiar relationships over a path between vertices with a single acquaintance.

To model this capability, we added some of the edges from the trust category to the model, but incorporated the desired secondary availability of trust edges by establishing a cost hierarchy. Under this hierarchy, costs for edges in both LOC and trust categories were calculated by subtracting the total number of LOC edges, $w_{2_{ij}}$, and trust edges, $w_{1_{ij}}$, between $i$ and $j$ from the value 13. The value 13 was chosen because the $max(w_{2_{ij}} + w_{1_{ij}}) = 12$, which results in a range of edge costs between 1 and 12. This construct rewards two terrorists who have multiple connections for trust and LOC by lowering the cost value of communicating with each other.

The next case in our cost hierarchy is LOC connections only. Similar to LOC and trust, LOC connections only simply subtracts the total number of LOC edges, $w_{2_{ij}}$ from the values 13, which results in cost values between 5 and 12. In an effort to delay the use of trust only edges, the total number of trust only edges, $w_{1_{ij}}$, between $i$ and $j$ was subtracted from the value 24, which results in a range of edge costs between 21 and 23. The value 24 was chosen since it results in a range of cost values that is higher than LOC only and LOC and trust, yet lower than the penalty cost. We established a uniform arbitrarily high penalty cost of $q_{ij} = 50$, when edges are attacked. As a result, this model always favors the edges according to the established cost hierarchy in calculating the shortest path. For comparison, we also created a uniform cost dictionary that weighted each edge as one regardless of the category or redundant edges within a category.

We conducted 16 attack scenarios, each corresponding to an increase in the available

number of attacks by one from 0 to 15. Figure 5.1 depicts the resultant parametric curves from executing the attack scenarios. The number of attacks is displayed on the x-axis and the corresponding shortest path distance is displayed on the y-axis for the uniform and hierarchical cost values. To solve this linear program, we used a commercial solver called Gurobi [61]. According to Meindl et al. [62], commercial optimization solvers such as CPLEX and Gurobi outperform open source solvers in both speed and accuracy.

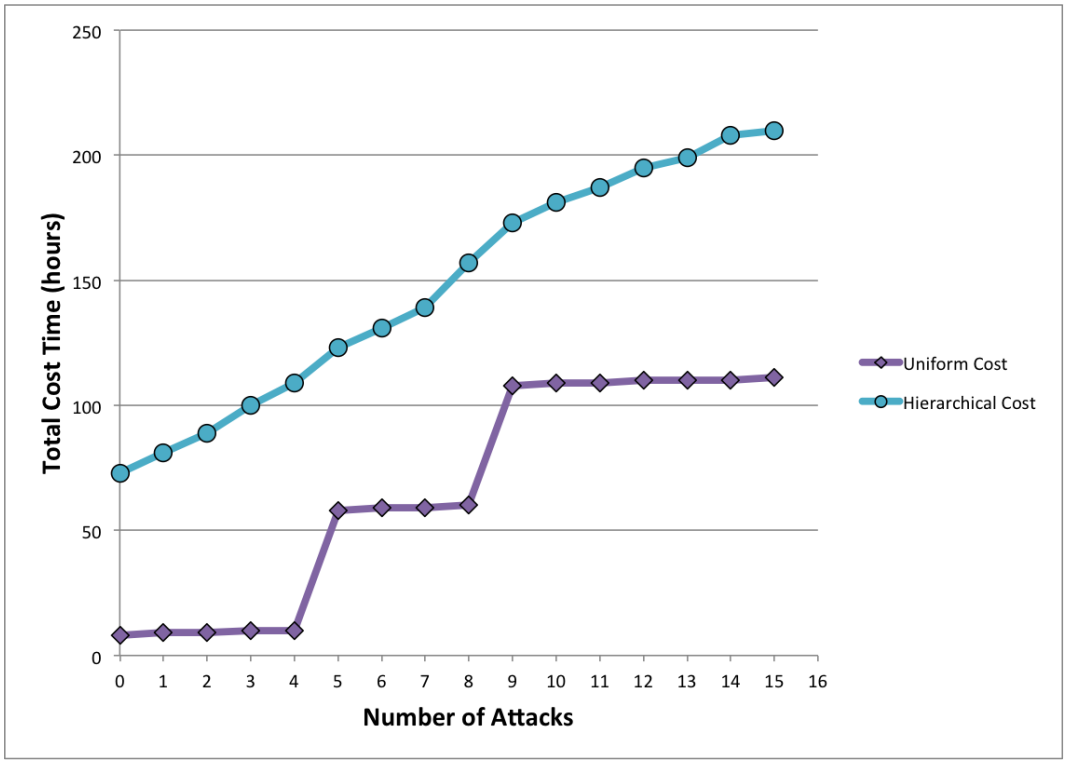



Figure 5.1: Cost in hours for attack plans in uniform and hierarchical cost models.

Notice in Figure 5.1 that the hierarchical cost scenario results in a relatively linear increase in the cost hours as the number of attacks increase. However, the rate of increase slows slightly after nine attacks. Conversely, the uniform cost scenario results in three noticeable plateaus at zero, five, and nine attacks. Since all edges are equally weighted for cost, the only noticeable increase occurs when vertices become completely disconnected. Under this scenario, the shortest path interdiction algorithm simply attacks the edges connected to the demand vertices in order of smallest to largest degree. Thus at attack number five, all edges

connecting to Ali Imron have been targeted resulting in the maximum penalty of 50. The algorithm then begins targeting all of the connections to Aris Munandar until he becomes disconnected at nine attacks. In the next section, we analyze the community properties of subcase 3.9 to enhance the attack strategy.

## 5.3 Community Properties and Attack Strategy

Subcase 3.9 was chosen to examine in depth due to its inclusion of (i) trust and (ii) trust and LOC edges. However, this subcase does not completely reflect the edges chosen to model the network flow problem. Subcase 3.9 does not include LOC only edges and includes extra edges from (i) trust and knowledge and (ii) trust, LOC, and knowledge combined. Yet, subcase 3.9 provides the best approximation for the network flows model out of the available threshold cases while still maintaining a relatively high average community quality value.

For each community, we examined the following properties: Size, Density (De), Total External Edge Count (TEEC), Total Internal Edge Count (TIEC), Number of Demand Vertices (NDV), Adjusted Conductance $O$ ($\phi^1$), Influence (I), Community Influence (CI), and Total Influence (TI), which are summarized in Table 5.1. We also examined Between Community Edge Count ($BCE_{ij}$), which is summarized in Table 5.2.

The community density is calculated by dividing the actual number of internal edges in the community by the maximum possible number of edges in the community. The maximum possible number of undirected edges for a clique is $n \cdot \frac{n-1}{2}$. To model this number for our converted directed network, we multiplied this value by two. It is important to note that our modification to the density equation is only possible because for every edge (i,j) there is a corresponding edge (j,i) in the network. For a community with $n$ vertices, and $|E|$ edges, we represent community density as:

$$De = \frac{|E|}{n \cdot (n-1)}. \tag{5.14}$$

The TEEC refers to the total number of edges that begin with a vertex in the community and terminate with a vertex outside the community. TIEC refers to the total number of edges that begin and end within vertices inside the community. The NDV is a count of the total number of vertices within the community that have a demand for the information commodity.

Between Community Edge Count calculates the total number of external edges from the source community to each destination community. For example, for $communit\,y_0$, the value for $BCE_{09}$ is calculated by counting the number of edges that originate in $communit\,y_0$ and terminate in $communit\,y_9$.

The Influence is calculated by summing the ratio of $BCE_{ij}$ of the source community to the $TEEC_j$ of each terminal community. For a fixed source community, $i$ and a terminal community $j$ influence is defined as:

$$I = \sum_j \frac{BCE_{ij}}{TEEC_j}, \forall i. \tag{5.15}$$

Influence allows us to determine how important the terminal community views the source community relative to the total number of external connections from the terminal community to the whole network. For example, if person A is friends with person B and person B only has three friends total, then person A's influence is $\frac{1}{3}$. However, if person B only has one friend, then person A increases their influence to $\frac{1}{1}$. We calculate CI by totaling the number of communities connected to the source community. The TI is calculated as their sum:

$$TI = I + CI. \tag{5.16}$$

Figure 5.2 summarizes four of the community properties including Size, TI, De, and $\phi^1$. The community names are displayed in order of increasing TI on the x-axis. Community size is represented as blue diamonds and TI is represented as green triangles on the primary left y-axis. Density is represented as purple squares, and $\phi^1$ is represented as red circles on the secondary right y-axis. We observe that the community density and adjusted conductance generally follow the same shape. Additionally, the total community influence generally increases as the size of the community increases.

Table 5.1: Subcase 3.9 community properties.

| Name | Size | De | TEEC | TIEC | NDV | $\phi^1$ | I | CI | TI |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 2 | 0.50 | 0 | 2 | 0 | 1.000 | 0.000 | 0 | 0.000 |
| 6 | 4 | 1.00 | 2 | 12 | 0 | 0.800 | 0.014 | 2 | 2.014 |
| 0 | 10 | 0.56 | 7 | 90 | 0 | 0.672 | 0.045 | 2 | 2.045 |
| 5 | 2 | 0.50 | 5 | 2 | 0 | 0.286 | 0.064 | 2 | 2.064 |
| 7 | 13 | 0.32 | 31 | 50 | 1 | 0.334 | 0.312 | 4 | 4.312 |
| 4 | 13 | 0.40 | 69 | 62 | 0 | 0.378 | 0.778 | 5 | 5.778 |
| 1 | 10 | 0.40 | 77 | 36 | 1 | 0.401 | 1.109 | 5 | 6.109 |
| 2 | 17 | 0.21 | 58 | 58 | 1 | 0.401 | 1.313 | 5 | 6.313 |
| 9 | 17 | 0.35 | 122 | 96 | 0 | 0.287 | 2.010 | 8 | 10.010 |
| 8 | 23 | 0.41 | 164 | 206 | 1 | 0.451 | 4.018 | 9 | 13.018 |
| misfit | 20 | 0.04 | 29 | 16 | 0 | NA | 0.349 | 5 | 5.349 |

Table 5.2: Subcase 3.9 community total influence summary.

| Name | $BCE_{i0}$ | $BCE_{i1}$ | $BCE_{i2}$ | $BCE_{i3}$ | $BCE_{i4}$ | $BCE_{i5}$ | $BCE_{i6}$ | $BCE_{i7}$ | $BCE_{i8}$ | $BCE_{i9}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 1 |
| 5 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 7 | 0 | 12 | 0 | 0 | 3 | 0 | 0 | 0 | 9 | 7 |
| 4 | 0 | 8 | 8 | 0 | 0 | 0 | 0 | 3 | 36 | 10 |
| 1 | 0 | 0 | 5 | 0 | 8 | 0 | 0 | 12 | 18 | 29 |
| 2 | 0 | 5 | 0 | 0 | 8 | 3 | 0 | 0 | 23 | 10 |
| 9 | 1 | 29 | 10 | 0 | 10 | 0 | 1 | 7 | 62 | 0 |
| 8 | 6 | 18 | 23 | 0 | 36 | 2 | 1 | 9 | 0 | 62 |
| Misfit | 0 | 5 | 9 | 0 | 4 | 0 | 0 | 0 | 9 | 2 |

The least influential and smallest community is $community_3$. This makes sense since $community_3$ is a component community and thus is isolated and incapable of influencing other communities. Not surprisingly, Noordin Top belongs to the largest community with the greatest total influence, $community_8$. The tactical commanders he is sending his orders to are each in separate demand communities including: $community_1$, $community_2$, $community_7$, and $community_8$. As we examine the properties of these demand communities, we notice that $community_7$ is the least influential, which means it is more isolated than the other communities. We hypothesize that the more isolated the community, the more vulnerable it will be to edge attacks. As a result, we can prioritize targeting edges that connect the source community to the demand communities in increasing order of influence. We discuss modeling this attack plan in more depth in Chapter 6.

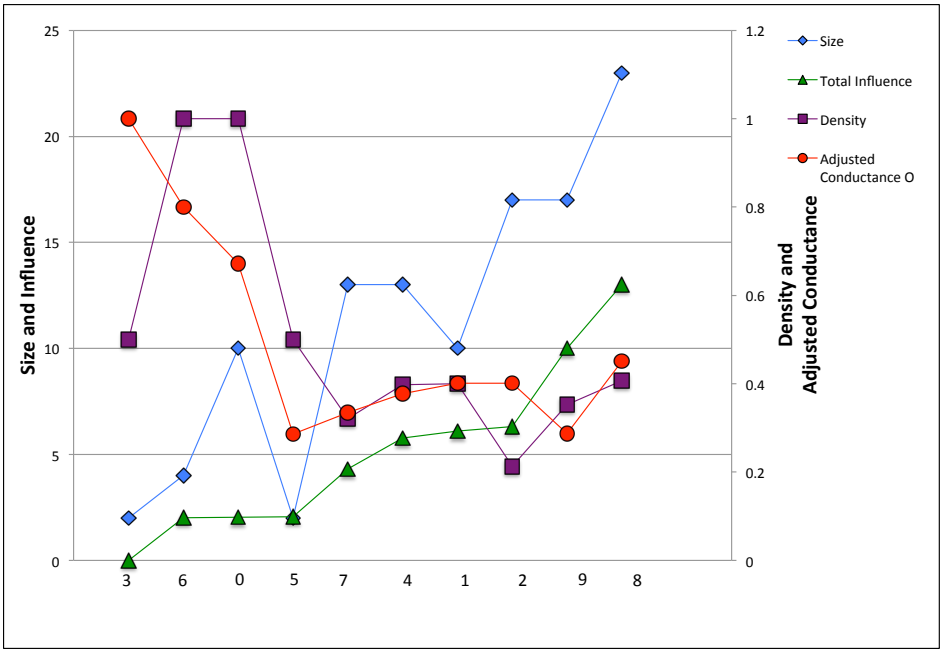



Figure 5.2: Subcase 3.9 community properties.

Another approach to using community information in the network is to reduce the complexity of the current attack plan. As seen in Equation 5.8, every edge represents an additional constraint, and consequently, more computations. The current Noordin Model has a total of 1196 edges. We believe that attacks should be primarily focused on edges inside communities that contain demand vertices and on edges that connect these communities to the remainder of the network. Given this logic, we can reduce the communities without a demand vertex to representation as a single vertex. Under this construct, the internal edges of $communit\,y_0$, $communit\,y_3$, $communit\,y_4$, $communit\,y_5$, $communit\,y_6$, $communit\,y_9$ and $communit\,y_{misfit}$ are considered defended and not available for attack. This reduces the number of edges subject to attack calculations to 628 from the original 1196. We consider the communities with demand as priority for analysis, and the remainder of the network as noise that can be ignored. As a result, we implement the shortest path interdiction algorithm on the simplified representation of the network. The goal of this simplification is to achieve similar damage, but to reduce the computation time. According to Lundh [63], the python

time module can be used to benchmark the run time of an algorithm. For our purposes, the time module recorded the elapsed time required for Gurobi to solve the shortest path interdiction dual algorithm as a function of the number of attacks.

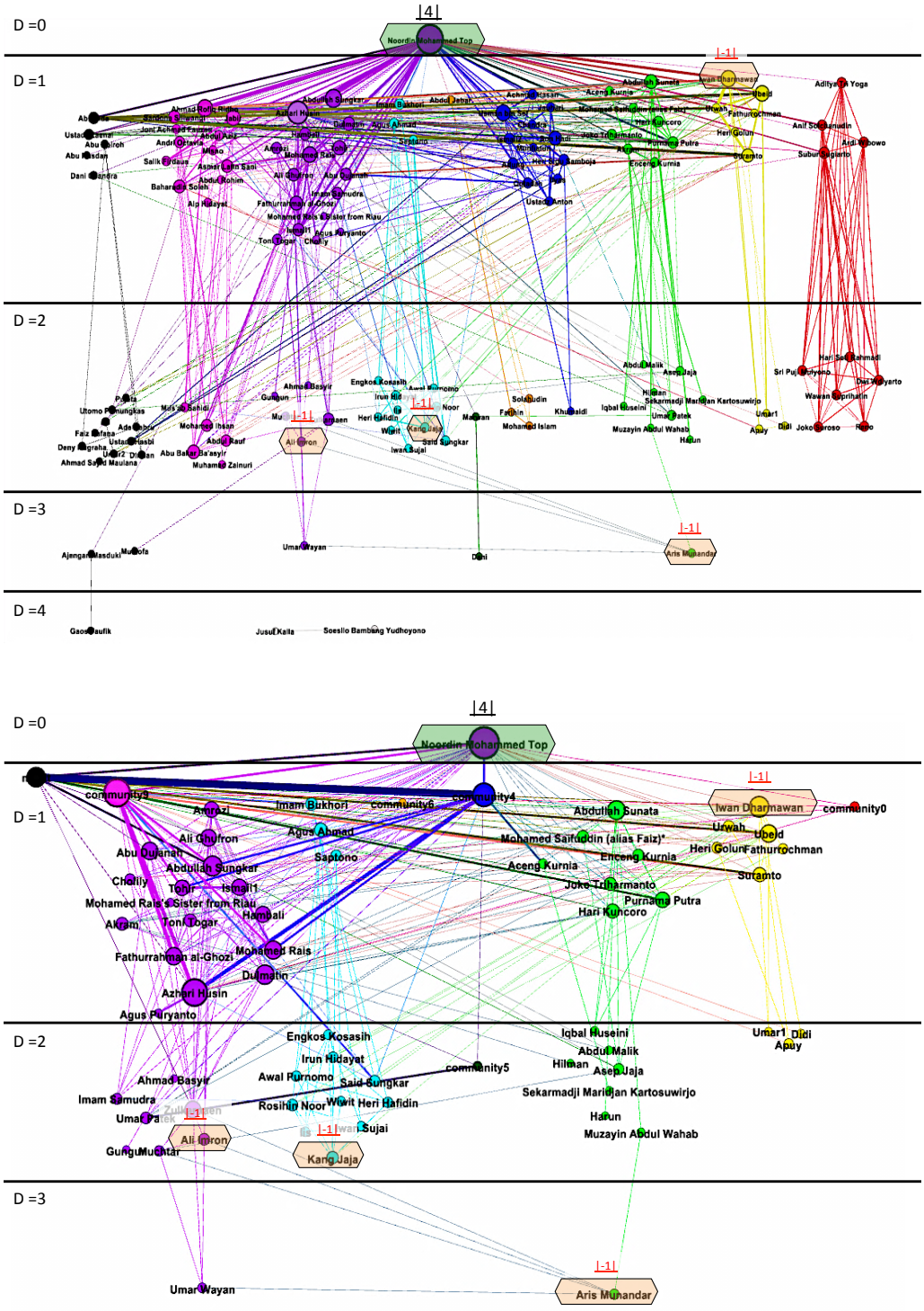



Figure 5.3: Noordin original and simplified attack models.

To model removing these edges from attack consideration, we place a large cost of 100 on the internal edges of the designated communities, and place a penalty cost equal to zero. Consequently, these edges will never be considered for the shortest path, and they will never be attacked.

In the top diagram in Figure 5.3 we arranged the vertices in the Noordin Network according to community and distance away from Noordin. The red color corresponds to $community_0$, the yellow color corresponds to $community_1$, the light green color corresponds to $community_2$, the gray color corresponds to $community_3$, the dark blue color corresponds to $community_4$, the dark green color corresponds to $community_5$, the orange color corresponds to $community_6$, the light blue color corresponds to $community_7$, the purple color corresponds to $community_8$, the pink color corresponds to $community_9$, and the black color corresponds to $community_{misfit}$. Using the same color scheme, the bottom diagram in Figure 5.3 illustrates our method of collapsing communities into single vertices that do not contain a demand vertex. Demand vertices are represented by a red negative one and an orange hexagon. The supply vertex, Noordin Top, is represented by a black four and a green hexagon. Vertices are further organized into groups according to the number of hops, $D$, that they are away from Noordin Top, where $D \in \{1, 2, 3, 4\}$.

The results of attacking the Noordin Network using the optimal attack strategy with uniform costs on all edges are displayed in Figure 5.4. We refer to the optimal attack as the results from directly applying the shortest path interdiction algorithm to the Noordin Network flow model. The community guided attack refers to the modified attack we implemented by simplifying the Noordin Network flow model based on community partitions. On the primary left y-axis, the optimal uniform cost is represented as purple squares with a solid line and the community guided uniform cost is represented as a red x with a dotted black line. The cost is represented in units of hours. On the secondary right y-axis, the optimal uniform cost time and community guided cost times to execute the algorithm are represented as blue circles and orange diamonds respectively. The execution time is represented in units of seconds.

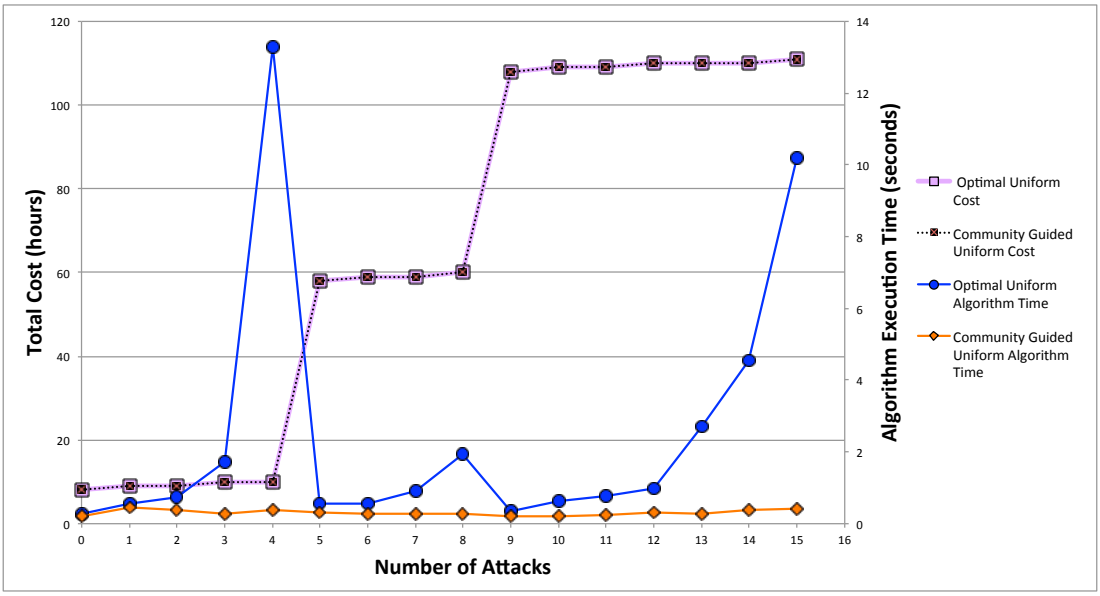



Figure 5.4: Uniform cost results.

We observe that the uniform cost for the optimal and community guided attacks are identical. Notice the spike in the amount of time to calculate the optimal attack strategy for four attacks. Another spike, though considerably smaller, is also visible for the optimal attack strategy for four attacks as well. As the attack number is increased from nine to 15, we observe a seemingly exponential increase in the time to execute the algorithm. As we increased the attack number beyond 15 to 16 we noticed a dramatic increase in the run time for the optimal attack from 10.19 to 113.20 seconds. However, 16 community guided attacks resulted in a nominal change from 0.40 to 0.39.

We attempted to execute a 50 attack scenario for the optimal attack, but the solver timed out after 30 minutes without determining an optimal solution. However, the community guided attack did determine an an optimal solution after only 0.39 seconds. Generally, we observe an increase in the run time of the algorithm as the cost value plateaus with a spike occurring right before a large increase in cost. We also observe that the community guided uniform cost time is virtually horizontal. This supports our assessment that communities can be used to reduce the complexity of the problem while achieving the same results.

The results of attacking the Noordin Network using the optimal attack strategy with hierarchical costs on all edges are displayed in Figure 5.5. The optimal hierarchical cost and

community guided cost produce perform similarly. However, the community guided cost does slightly out perform the optimal cost at several attacks including the final attack case. Yet, there are a few instances where the community guided cost is lower than the optimal. To understand these discrepancies, the Gurobi reference manual [61] reveals that Gurobi has a default optimality gap of $1 \times 10^{-4}$. This tolerance is also known as the relative optimality criteria gap.

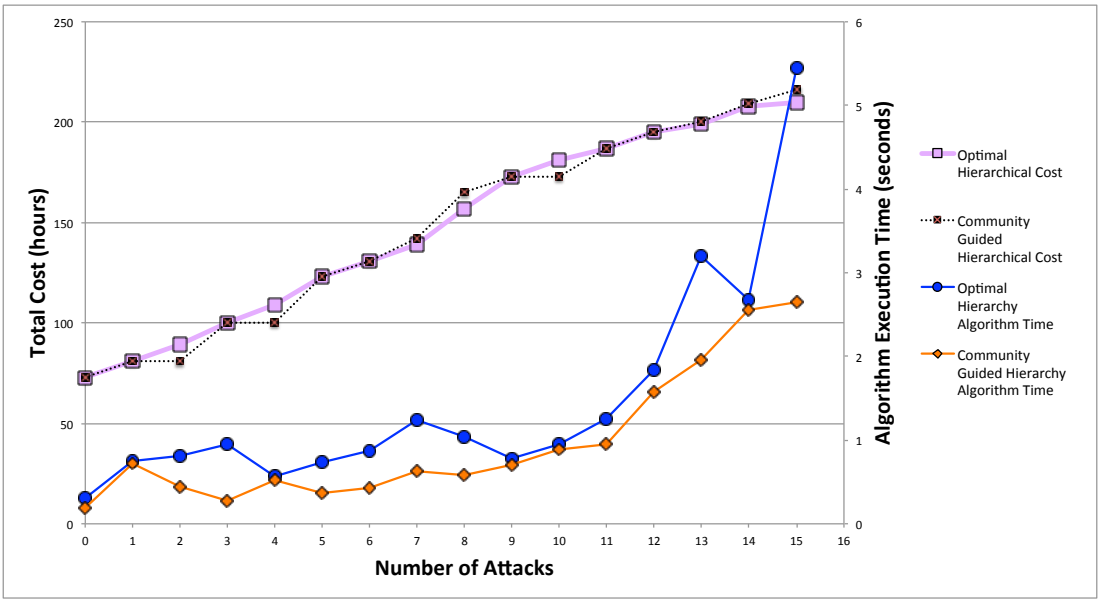



Figure 5.5: Hierarchical cost results.

Brown et al. [60] state that the optimal solution occurs when the difference gap between upper and lower bounds is ideally zero. However, this could prove to take an inordinate amount of time. Relative optimality criteria allows the user to choose the level of accuracy of the optimal solution at the expense of time. It is possible that this default optimality criteria gap caused the solutions to vary slightly between optimal and community guided approaches.

We observe for the hierarchical algorithm times, that both optimal and community guided run times follow a gradual increase until attack nine where the slope increase sharply for the optimal strategy attack strategy. We also attempted a 50 attack scenario for the optimal cost, but the solver timed out after 30 minutes without determining an optimal solution. However, the community guided cost determined an optimal solution after only

10.31 seconds. Similar to the uniform cost results, the hierarchical cost results support simplifying the network model using the community partitions for similar cost values and reduced algorithm performance times. Additionally, simplifying the model potentially allows the user to reduce the optimality criteria for a more precise optimal solution.

In this chapter we demonstrated one potential use for exploiting community properties to disrupt terrorist networks. In the final chapter, Chapter 6, we discuss improvements to both the community detection algorithm and modeling the network as a network flow problem. Additionally, we discuss some general observations and recommendations for continuing this research in the future.

# CHAPTER 6:
# Future Work and Recommendations

In this thesis we have presented a community detection algorithm for multiplex dark networks. We have also demonstrated the utility of partitioning a network into communities to disrupt network functionality. Given our results, we provide some future direction in this chapter for improving our community detection algorithm, enhancing our network flow model, and alternative strategies for disrupting networks using communities. Finally, we summarize the key findings from this thesis and offer some general conclusions.

## 6.1 Community Detection Algorithm Improvements

In this section we discuss improving the community detection algorithm based upon our current results. We suggest a procedure for selecting layers, determining category weights in Subsection 6.1.1. We also discuss the effect of removing misfit vertices from the graph on community quality in Subsection 6.1.2, and some recommendations for applying the algorithm to multiplex networks with complete information in Subsection 6.1.3.

### 6.1.1 Layer and Category Importance

Our community detection algorithm requires user input for selecting layers from an available data set. For the dark networks we examined, we selected layers based on similar meaning. However, some networks may not be as easily sorted, and user intuition on layer selection may not be as obvious. This problem is particularly difficult for large data sets with many layers.

Sharma et al. [64] suggest that one technique for selecting layers is to determine the relative importance of each layer. His approach focuses on the impact of missing data with respect to network modeling. They asserts that missing information or data has a more profound effect upon the analysis conducted on multiplex networks than other types of network models. One of the metrics he uses is called exclusive relevance. This metric determines the importance of a particular layer, $L$, based upon the fraction of connections from a node, $n$, to the nodes adjacent to $n$ in $L$. Crawford et al. [65] explored the concept of network

layer importance by examining the contribution of a layer with respect to final community structure of a graph.

Recall from Section 2.3, Taylor et al. [29] proclaim that layer aggregation is extremely beneficial for network analysis if conducted appropriately. They asserts that one of the fundamental problems of layer aggregation is determining which layers to aggregate. They believe that using all of the available layers of a network can actually over-model a network. Over-modeling refers to threshold at which the amount of information used to model the system hinders the analysis. Taylor et al. explain that over-modeling leads to computational and memory storage difficulties. They suggests that repetitive layers should be aggregated to more concisely represent the network. Their work supports the idea of increased detectability of communities in a network when layers are aggregated.

In [65] the authors establish additional criteria for layer aggregation using community evolution to determine which layers or set of layers are dominant in the network for producing communities. They examined the same three dark terrorist networks as this thesis, as well as two transportation networks. They used Normalized Mutual Information (NMI), purity, density, and modularity as metrics for comparing our resultant community evolution cases to the established ground truth communities from the layer aggregation.

The key findings in [65] were that layer uniqueness and edge density were the most important factors in assigning importance to layers and categories. The knowledge category was the dominant category for the Noordin Network. Recall that the knowledge category accounts for more than 50% of the edges in the Noordin data set, thus it is expected to be the most dominant category. However, combining knowledge category layers with trust category layers resulted in the most accurate approximation of ground truth communities. The trust category represents many of the social relationships, which suggests that the community structure of social relationships is unique.

The research of [65] implies that uniqueness and edge density are important factors when determining which layers or categories are dominant in the data set. Their research requires further verification on other types of networks, but identifying layers that are both dense and unique in structure is a promising method for layer selection and inclusion for analytical purposes. However, while density is easily calculated, uniqueness is a more difficult quality to evaluate prior to conducting the layer dominance procedure.

This concept could be applied to category importance to determine an appropriate weight for each category in step 5 of our community detection algorithm. To extend this thesis, the dominant set of layers from each category could be selected to model the Noordin Network. Additionally, the categories could be weighted according order of dominance explained in [65]: knowledge, trust, and then LOC. For more details on the layer and category dominance results of the three dark networks see [65].

### 6.1.2 Misfit Community Elimination

Another modification to our community detection algorithm involves removing the misfit vertices from the network. In Chapter 4, our results revealed that the networks with larger misfit communities tended to produce poorer quality communities according to adjusted conductance and cluster adequacy. We initially believed removing vertices with high external connections would increase the community quality value. However, since the misfit vertices are still part of the graph, the communities external connections remain as high or higher that reduces the adjusted conductance score. Also, by removing vertices from the community, the community loses some of the valuable internal connections that cluster adequacy favors. Simply labeling a vertex as a misfit is not enough to benefit increased community quality. As an extension to this thesis, we recommend physically eliminating misfits and all of their associated connection from the graph. The resultant graph would be considerably less complex and produce higher quality communities according to our established metrics. We performed an initial localized experiment on the Noordin Network case 3 to demonstrate the potential benefits of pursuing this extension.

We explored misfit elimination for the Noordin Network case 3 and compared it to our original community quality results for average adjusted conductance, $\phi^1$, and cluster adequacy for plotting communities in the original graph. We subtracted the values produced by eliminating the misfits from the results where misfits remained present to produce the change in values, $\Delta$. These results are summarized in Table 6.1.

Notice that some of the subcase average adjusted conductance and cluster adequacy values did not change. This is because there are no identified misfits in these subcases. A preliminary analysis of the results suggests that physically removing misfit vertices from the network always improves the quality of the community according to cluster adequacy and

Table 6.1: Noordin case 3 misfit elimination Δ.

| | Misfits Present | | Misfits Eliminated | | Changes in Values (Δ) | |
|---|---|---|---|---|---|---|
| **Subcase** | **Average $\phi^1$** | **Cluster Adequacy** | **Average $\phi^1$** | **Cluster Adequacy** | **Δ Average $\phi^1$** | **Δ Cluster Adequacy** |
| 1 | 0.578 | 0.214 | 0.578 | 0.214 | 0.000 | 0.000 |
| 2 | 0.602 | 0.244 | 0.602 | 0.244 | 0.000 | 0.000 |
| 3 | 0.578 | 0.226 | 0.578 | 0.226 | 0.000 | 0.000 |
| 4 | 0.149 | 0.049 | 0.191 | 0.096 | 0.047 | 0.042 |
| 5 | 0.242 | 0.054 | 0.295 | 0.085 | 0.031 | 0.053 |
| 6 | 0.162 | 0.057 | 0.213 | 0.118 | 0.060 | 0.051 |
| 7 | 0.263 | 0.082 | 0.303 | 0.111 | 0.029 | 0.040 |
| 8 | 0.383 | 0.143 | 0.419 | 0.173 | 0.031 | 0.036 |
| 9 | 0.501 | 0.157 | 0.520 | 0.188 | 0.032 | 0.019 |

average adjusted conductance. The greatest improvements generally increased for subcases 4, 5, and 6, which had the largest populations of misfits with 64, 47, and 68 respectively. These results need to be compared with other cases and networks to verify the trends we identified with our preliminary analysis.

Throughout this thesis, the Noordin Network was the primary focus for in-depth analysis. We can continue dissecting the Noordin data set using different thresholds, weights, categories, and layer combinations. All of the recommendations and applications should also be applied to the Boko Haram, and FARC networks. Additionally, the results need to be confirmed with other classified dark network data sets. To demonstrate the flexibility of the community detection algorithm, we need to apply our methodology to a variety of networks.

### 6.1.3 Complete Information Multiplex Networks

This thesis has focused on dark networks where incomplete information is an inherent and challenging property. However, partitioning non dark networks into communities is also useful for analysis. For example, a group of NPS students in a class may be represented as a multiplex network with layers representing different skill sets. In this case, the PDC would be equivalent to a group project team. By sorting the students into different communities based upon skills, the professor can easily identify and build modular project teams that are balanced in terms of a wide spectrum of capabilities. In this example, all of the information used to construct the network is already available. We can apply our algorithm to networks,

such as this example, which contain complete information. For these networks, the only recommended change is to remove the clique conversion instruction from step 4 of the algorithm. Converting the communities into cliques serves no logical purpose for inferring edges when all of the edges are already known. In the next section, we explore some alternative modeling practices that could enhance the model described in Chapter 5.

## 6.2 Network Flow Model Enhancements

In this section we discuss two improvements to our network flow model focused more realistically modeling the terrorist network. These improvements include modeling invasive attacks, and defender capabilities.

### 6.2.1 Invasive Attacks

In Chapter5 we focused on attacking edges between vertices. These attacks equated to non-invasive actions such as jamming cell phone communication or indirectly affecting the terrorist's ability to communicate by causing a route to be blocked or closed. It is conceivable that non-invasive attacks can be implemented at a higher volume and low risk to the attacker. However, if more direct action is desired, we can model an invasive attack that would mean physically removing or capturing an individual in the network. To model the removal of a vertex, Alderson et al. [59] suggest vertex splitting. Vertex splitting is a method for representing each vertex as a additional edge in the data set. Of course, this increases the number of edges, $|E|$, in the model by the number of vertices, $n$. Figure 6.1 demonstrates how a green vertex $i$ can be replaced by a pair of vertices, $i'$, which is red, and $i''$, which is yellow, and an edge, $(i', i'')$, between $i'$ and $i''$.

Notice how all of the edges flowing into $i$ are now attached to vertex $i'$, and that the edges flowing out of vertex $i$ are now connected to $i''$. To model attacking vertex $i$, we can now attack edge $(i', i'')$. This modification to the data set allows the attacker to attack the best combination of edges and vertices in the graph. Attacking a vertex can be extremely effective by disrupting multiple connections with one attack. However, direct attacks, such as removing vertices, should be more restrictive and costly for the attacker to execute. For example, the number of attacks could be constrained as a function of invasive, $y$ and non-invasive, $y'$ attacks such that:

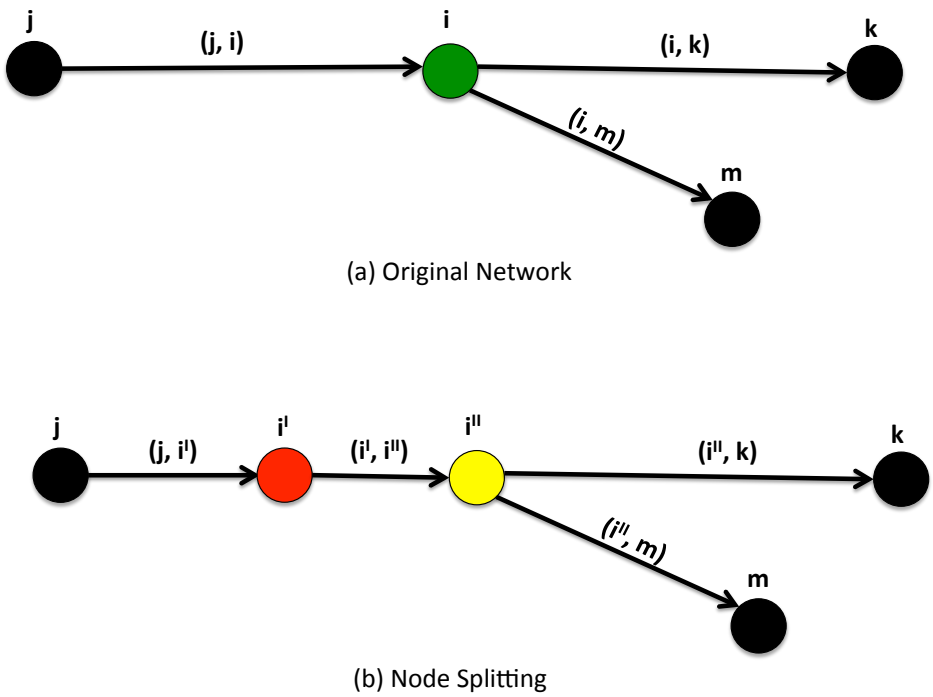



Figure 6.1: Vertex splitting example.

$$num_attacks = 2y + y'. \tag{6.1}$$

Equation 6.1 allows the attacker to choose an optimal combination of invasive and non-invasive attacks for a given number of available attacks with the understanding that invasive attacks are twice as resource depleting as non-invasive attacks. Constraining the attack resources in this manner adds realism to the difficulty and risk associated with direct action verses indirect actions.

The knowledge of which vertices are involved in the shortest paths is also valuable for analysis. For example, if invasive action is not possible, and the message cannot be blocked by less invasive measures, it is still possible to gather valuable intelligence by monitoring the vertices along the shortest path. Another modification to the network flow model involves the defender capability.

### 6.2.2 Defender Capabilities

Our model primarily focused on the attacker's resources and capabilities to interdict the network. However, in reality, an intelligent defender understands their own network vulnerabilities and, given a finite amount of resources, strengthens these weaknesses accordingly. Alderson et al. [59] recommend formulating the defender model to strengthen existing connections or potentially build new connections in the network. In the Noordin network, forging new relationships could represent sending more individuals to planning meetings or forcing individuals to train more together to potentially build friendship and stronger working relationships. However, Krebs [39] reveals that building too many new connections could come at the costly expense of secrecy.

Defending an individual could have a more physical representation such as placing him in a safe house with armed guards or constantly moving his location. Alderson et al. [59] believe that an intelligent defender can mitigate the effects of a network attack by incorporating a new binary decision variable into our existing model, $w_{ij}$, which is equal to one if a new connection between terrorists is forged, and zero otherwise. Alderson et al. [59] explain that the objective function is modified to the following equation:

$$\max_{w} \min_{x} \max_{y} y_{ts} - \sum_{(i,j)\in E} 2(y_{ij} + y_{ji})x_{ij}, \tag{6.2}$$

where $x_{ij}$ is the attacker's decision to attack edge $(i, j)$, $y_{ts}$ is an artificial flow variable that connects the source, $s$, to the demand vertices, $t$, $y_{ij}$ is the flow of information from vertex $i$ to $j$, and $E$ are all of the undirected edges between $i$ and $j$; where $i < j, \forall(i, j) \in E$.

This new objective function is subject to additional constraints involving new data that includes a defense budget as well as a defense cost. For more information on how to formulate the defender model, see [59].

Alderson et al. [66] reveal that the practice of defending a vertex also provides valuable attack alternatives. By protecting or hardening a link in the optimal attack plan, the attacker is able to develop multiple attack strategies. These plans are less optimal than the original attack, but provide alternative options for decision makers to use in case the optimal attack cannot be physically carried out or is too costly.

## 6.3 Alternative Disruption Strategies

Our model focused on interdicting the shortest path from a source vertex to a set of destination vertices. We used the knowledge of the communities to reduce the complexity of our model in order to enhance the optimal solution time. In this section, we discuss some additional ideas for using community properties to disrupt the network according to shortest path interdiction and methods for measuring network resilience.

### 6.3.1 Community Isolation

Community isolation is another potential option for exploiting community properties for disruption purposes. We recommend targeting the external edges of the communities with demand vertices in order of increasing influence in the network. By this logic, we would push the weaker communities away first, since they will require fewer external edge attacks. To conduct this experiment, we recommend isolating each community for attack by only leaving the isolated communities external edges available for attack. For each community, record the number of external attacks required to fully isolate the community from the network, as well as the total resultant cost damage to the network.

A modification of this option is to extend the concept of total influence to the the individual vertex level. We can then target the vertices that are the most influential in the network. In essence, this approach is similar to determining vertex centrality with respect to community influence. Establishing the total influence of a vertex enhances our ability to identify the key brokers in the network. targeting these brokers assists in the process of isolating the communities they belong to from the rest of the network and consequently disrupts the flow of information. This experimental idea can also be applied to the other dark networks as well for additional trials and verification of results.

### 6.3.2 Optimal Community Size

In Chapter5, we examined only one subcase threshold, subcase 3.9. All of our threshold cases need to be examined to determine if there is an optimal community size for applying our strategy of collapsing communities that do not contain supply or demand vertices. If the number of communities is too high, then we risk protecting edges from attack that would normally provide the shortest path from supply to demand vertices. Conversely,

if the number of communities is too low then collapsing the demand free communities may not significantly decrease the solution evaluation time. Thus, an optimal community size would reduce the complexity of the linear program while still determining an optimal solution within an acceptable tolerance. We recommend testing the remaining thresholds and comparing the results to the established community quality metrics.

Community size optimality will also depend on the nature of the attack model scenario. For example, if one threshold subcase placed all of the demand vertices into one small community, then all attack resources could be focused on isolating one community. However, if the defender objective was to deliver a message to every vertex in the network, then an attacker strategy based on separating the network into multiple components might be more appropriate. Many small size communities might be more beneficial for fracturing the entire network. We can guide our attack based on the aforementioned community isolation technique described in the previous subsection. Now that we have identified some other methods for attacking the network, in the next subsection we discuss some additional performance metrics for measuring the quality of the attack with respect to the resilience of the network.

### 6.3.3 Network Resilience

Alderson et al. [66] recommends measuring the operational resilience, which is the adaptability of the network to maintain functionality after attacks. Alderson et al. [67] propose that one method to quantitatively measure operational resilience is to build the resilience curve, which measures the post-attack cost growth as a function of the number of lost components. This is similar to the parametric curves in Section 5.2, but also incorporates the defender capabilities described in Section 6.2.2. Alderson et al. explains that the shape of the parametric curve indicates the resilience of the network. Curves that begin with a steeper slope and then gradually level (high elbow) represent less resilient networks. Conversely, Alderson et al. points out that curves with a more gradual slope followed by a steep slope (low elbow) after many attacks represent a more resistant system.

Bhatia et al. [68] discuss an alternative method for measuring network resilience using percolation theory to define the State of Critical Functionality (SCF). They define Total Functionality (TF) to be the number of demand vertices in the giant or largest component

of the graph after zero attacks. They then record the Fragmented Functionality (FF), which is the number of demand vertices that are still part of the giant component as a function of the number of attacks on the network. Using the definitions of TF and FF, Bhatia et al. define SCF as:

$$SCF = \frac{FF}{TF}. \tag{6.3}$$

Bhatia et al. [68] explain that SCF values range between zero and one, with values closer to one representing higher functionality and resilience for the network. The metrics introduced by Alderson et al. and Bhatia et al. for network resilience could be applied to our network data sets to determine the adaptability of the networks to attack and defensive operations.

## 6.4 Conclusions

This thesis has presented an alternative method for conducting community detection in multiplex networks. The algorithm was specifically tailored for dark network data sets, and was tested on three data sets. However, the user engagement in our algorithm allows it to be flexible for other networks. The thresholding option in our algorithm produces different numbers of communities of different sizes according to the user's purpose.

We noticed that the community quality generally increased with the size of the community. The larger communities were developed under the provisions of the most relaxed threshold values. The observation of threshold relaxation will most likely depend on the network, but we do believe an optimal value does exist for most networks. However, optimality depends on the goal the communities will be used for. Some networks have poor community structure in general due to high connectivity amongst all vertices. In these cases, the graph would prefer to remain one large community. The degree distribution may provide an initial indicator on the potential for good quality communities, but the optimal number of communities depends more specifically on the arrangement of the connectivity of the data set. The speculation of the generality of our observations to other types of networks provides a lot of potential for continuing this research.

The main purpose of our community development was to disrupt a terrorist network. With this goal in mind, we formulated a community guided shortest path interdiction network

flow model. Subcase 3.9 provided the necessary community compositions to guide the shortest path interdiction model towards a faster solution, and it was the only subcase tested for validation. More trials using other subcases may reveal a more optimal community composition, but subcase 3.9 demonstrated the utility in reducing solution time by using our community guided approach. Our focus on first defining a purpose for community detection helped guide our algorithm development into a working procedure with tangible results. We believe that detecting purpose-driven communities in multiplex networks by thresholding user-engaged layer aggregation is a promising area of research that should be continued and examined with more data sets in the future.

THIS PAGE INTENTIONALLY LEFT BLANK

# List of References


[1] G. Bianconi, "Statistical mechanics of multiplex networks: Entropy and overlap," *Physical Review E*, vol. 87, no. 6, pp. 1–15, 2013.

[2] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.

[3] R. M. Bakker, J. Raab, and H. B. Milward, "A preliminary theory of dark network resilience," *Journal of Policy Analysis and Management*, vol. 31, no. 1, pp. 33–62, 2012.

[4] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, vol. 2, no. 3, pp. 203–271, 2014.

[5] M. Newman, *Networks: An introduction*. Oxford, England: Oxford University Press, 2010.

[6] B. Bollobás, *Modern graph theory*. New York, New York: Springer Science & Business Media, 1998, vol. 184.

[7] M. De Domenico, M. A. Porter, and A. Arenas, "Muxviz: A tool for multilayer analysis and visualization of networks," *Journal of Complex Networks*, pp. 1–18, 2014.

[8] M. De Domenico, A. Lima, P. Mougel, and M. Musolesi, "The anatomy of a scientific rumor," *Scientific Reports*, vol. 3, 2013.

[9] G. K. Orman, V. Labatut, and H. Cherifi, "Towards realistic artificial benchmark for community detection algorithms evaluation," *International Journal of Web Based Communities*, vol. 9, no. 3, pp. 349–370, 2013.

[10] S. Fortunato and C. Castellano, "Community structure in graphs," in *Computational Complexity*. Springer, 2012, pp. 490–512.

[11] L. Eroh and R. Gera, "Global alliance partition in trees," *J. Combin. Math. Combin. Comput*, vol. 66, pp. 161–169, 2008.

[12] D.-Z. Du, K.-I. Ko, and X. Hu, *Design and analysis of approximation algorithms*. New York, New York: Springer Science & Business Media, 2011, vol. 62.

[13] G. K. Orman, V. Labatut, and H. Cherifi, "On accuracy of community structure discovery algorithms," *Journal of Convergence Information Technology*, 2011.

[14] L. Ana and A. K. Jain, "Robust data clustering," in *Proceedings of the 2003 IEEE Conference on Computer Society*, vol. 2, 2003, pp. II–128.

[15] L. G. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney, "Think locally, act locally: Detection of small, medium-sized, and large communities in large networks," *Physical Review E*, vol. 91, no. 1, pp. 1–29, 2015.

[16] F. R. Chung, *Spectral graph theory*. Providence, Rhode Island: American Mathematical Soc., 1997, vol. 92.

[17] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 631–640.

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

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. 1–13, 2008.

[20] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, pp. 1–5, 2004.

[21] A.-L. Barabási, *Network science*. Cambridge, England: Cambridge university press, 2016.

[22] G. K. Orman and V. Labatut, "A comparison of community detection algorithms on artificial networks," in *Discovery science*. Springer, 2009, pp. 242–256.

[23] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[24] S. F. Everton, *Disrupting dark networks*. Cambridge, England: Cambridge University Press, 2012, no. 34.

[25] M. M. Siems, "A network-based taxonomy of the world's legal systems," 2014.

[26] S. P. Borgatti, M. G. Everett, and L. C. Freeman, "Ucinet for windows: Software for social network analysis," 2002.

[27] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, pp. 1–6, 2004.

[28] H. Zhang, X. Chen, J. Li, and B. Zhou, "Fuzzy community detection via modularity guided membership-degree propagation," *Pattern Recognition Letters*, vol. 70, pp. 66–72, 2016.

[29] D. Taylor, S. Shai, N. Stanley, and P. J. Mucha, "Enhanced detectability of community structure in multilayer networks through layer aggregation," *arXiv preprint arXiv:1511.05271*, 2015.

[30] V. Batagelj, "Notes on blockmodeling," *Social Networks*, vol. 19, no. 2, pp. 143–155, 1997.

[31] T. P. Prescott and A. Papachristodoulou, "Layering in networks: the case of biochemical systems," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 4544–4549.

[32] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *science*, vol. 328, no. 5980, pp. 876–878, 2010.

[33] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, 2012.

[34] G. Didier, C. Brun, and A. Baudot, "Identifying communities from multiplex biological networks," *PeerJ*, vol. 3, pp. 1–20, 2015.

[35] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Community mining from multi-relational networks," in *Knowledge Discovery in Databases: PKDD 2005*. Springer, 2005, pp. 445–452.

[36] M. Rocklin and A. Pinar, "On clustering on graphs with multiple edge types," *Internet Mathematics*, vol. 9, no. 1, pp. 82–112, 2013.

[37] S. Howison, M. Porter, M. Bazzi, S. Williams, M. McDonald, and D. Fenn, "Community detection in temporal multilayer networks, with an application to correlation networks," *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 2015.

[38] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *Artificial neural networks–ICANN 2009*. Springer, 2009, pp. 175–184.

[39] V. E. Krebs, "Mapping networks of terrorist cells," *Connections*, vol. 24, no. 3, pp. 43–52, 2002.

[40] M. K. Sparrow, "The application of network analysis to criminal intelligence: An assessment of the prospects," *Social networks*, vol. 13, no. 3, pp. 251–274, 1991.

[41] B. H. Erickson, "Secret societies and social structure," *Social Forces*, vol. 60, no. 1, pp. 188–210, 1981.

[42] S. F. Everton, "Network topography, key players and terrorist networks," in *annual conference of the Association for the Study of Economics, Religion and Culture in Washington, DC*, 2009.

[43] L. M. Gerdes, *Illuminating Dark Networks: The Study of Clandestine Groups and Organizations*. Cambridge, England: Cambridge University Press, 2015, vol. 39.

[44] N. Roberts and S. F. Everton., "Terrorist data: Noordin top terrorist network," https://sites.google.com/site/sfeverton18/research/appendix-1, June 2011.

[45] M. Bastian, S. Heymann, M. Jacomy *et al.*, "Gephi: An open source software for exploring and manipulating networks." *ICWSM*, vol. 8, pp. 361–362, 2009.

[46] J. Sall, A. Lehman, M. L. Stephens, and L. Creighton, *JMP start statistics: A guide to statistics and data analysis using JMP*. SAS Institute, 2012.

[47] K. Cherven, *Network graph analysis and visualization with Gephi*. Birmingham, England: Packt Publishing Ltd, 2013.

[48] T. G. Lewis, *Network science: Theory and applications*. Hoboken, New Jersey: John Wiley & Sons, 2011.

[49] A. Walker, *What is Boko Haram?* Washington, DC: US Institute of Peace, 2012.

[50] D. Cunningham, "The boko haram network. [machine-readable data file]," https://sites.google.com/site/sfeverton18/research/appendix-1, June 2014.

[51] G. Weimann, *Terror on the Internet: The new arena, the new challenges*. Washington, DC: US Institute of Peace Press, 2006.

[52] D. Cunningham, S. Everton, G. Wilson, C. Padilla, and D. Zimmerman, "Brokers and key players in the internationalization of the farc," *Studies in Conflict & Terrorism*, vol. 36, no. 6, pp. 477–502, 2013.

[53] M. A. Cheever, J. P. Allison, A. S. Ferris, O. J. Finn, B. M. Hastings, T. T. Hecht, I. Mellman, S. A. Prindiville, J. L. Viner, L. M. Weiner *et al.*, "The prioritization of cancer antigens: a national cancer institute pilot project for the acceleration of translational research," *Clinical Cancer Research*, vol. 15, no. 17, pp. 5323–5337, 2009.

[54] B. Martin, T. Manacapilli, J. C. Crowley, J. Adams, M. G. Shanley, P. Steinberg, and D. Stebbins, "Assessment of joint improvised explosive device defeat organization (jieddo) training activity," DTIC Document, Tech. Rep., 2013.

[55] U. J. F. Command, "Commander's handbook for attack the network," 2011.

[56] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, *Pyomo–optimization modeling in python*. Springer Science & Business Media, 2012, vol. 67.

[57] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows," DTIC Document, Tech. Rep., 1988.

[58] "Shortest path network interdiction," class notes for OA4202, Department of Operations Analysis, Naval Postgraduate School at Monterey, CA, Fall 2015.

[59] D. L. Alderson, G. G. Brown, and W. M. Carlyle, "Assessing and improving operational resilience of critical infrastructures and other systems," *Stat*, vol. 745, p. 70, 2014.

[60] G. Brown, M. Carlyle, A. Abdul-Ghaffar, and J. Kline, "A defender-attacker optimization of port radar surveillance," *Naval Research Logistics (NRL)*, vol. 58, no. 3, pp. 223–235, 2011.

[61] *Gurobi Optimizer Reference Manual*, Gurobi Optimization, Inc., Houston, TX, 2016, release 6.5.1.

[62] B. Meindl and M. Templ, "Analysis of commercial and free and open source solvers for linear optimization problems," *Eurostat and Statistics Netherlands within the project ESSnet on common tools and harmonised methodology for SDC in the ESS*, 2012.

[63] F. Lundh, *Python standard library*. Sebastopol, CA: " O'Reilly Media, Inc.", 2001.

[64] R. Sharma, M. Magnani, and D. Montesi, "Missing data in multiplex networks: a preliminary study," in *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*. IEEE, 2014, pp. 401–407.

[65] C. Brian, R. Gera, R. Miller, and B. Shrestha, "Community evolution in multiplex layer aggregation," submitted (2016).

[66] D. L. Alderson, G. G. Brown, and W. M. Carlyle, "Operational models of infrastructure resilience," *Risk Analysis*, vol. 35, no. 4, pp. 562–586, 2015.

[67] D. L. Alderson, G. G. Brown, W. M. Carlyle, and L. A. Cox Jr, "Sometimes there is no most-vital arc: assessing and improving the operational resilience of systems," DTIC Document, Tech. Rep., 2013.

[68] U. Bhatia, D. Kumar, E. Kodra, and A. R. Ganguly, "Network science based quantification of resilience demonstrated on the indian railways network," *PLOS one*, vol. 10, no. 11, pp. 1–17, 2015.

# Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California