Destabilizing Terrorist Networks

John Keane
John.F.Keane.22@Dartmouth.edu

Follow this and additional works at: https://digitalcommons.dartmouth.edu/senior_theses
Part of the Terrorism Studies Commons

Recommended Citation

This Thesis (Undergraduate) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Undergraduate Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
DESTABILIZING TERRORIST NETWORKS

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Bachelor of Arts

in

Computer Science

by

John Keane

DARTMOUTH COLLEGE

Hanover, New Hampshire

June 1, 2022

Examing Committee:

_____________________________________________
Soroush Vosoughi, Chair

_____________________________________________
V.S. Subrahmanian

_____________________________________________
Alberto Quattrini Li

_____________________________________________
Charles Palmer
Abstract

Terrorism is a threat to global security and instills fear in the lives of people across the world. Over the past decades, billions in $\text{USD}$ have been invested in counter-terrorism efforts. One approach to counter-terrorism is to destabilize terrorist organizations such that they are less effective at carrying out attacks. Previous work has investigated how to best proceed in this direction, such as which terrorists to target. Terrorist organizations have also been modeled as networks, where nodes can represent factions and/or terrorists. Research has been done to understand the network dynamics and link the structure of such networks to their lethality, and these measures have been used to determine the set of actions that minimize a network’s lethality. These actions typically consist of modifying or removing nodes from the network, but this paper considers a new action: modifying the edges in the network, that is, changing the relationship between two factions or terrorists. We use existing methods of network lethality analysis on faction-faction networks, and we assess the viability of this action against more established ones in terms of reducing the lethality. We find that adjusting relationships between factions is a viable and cost-effective method, and it calls for further investigation.
Preface

I would like to thank VS Subrahmanian, Chiara Pulice, and Youzhi Zhang for supporting me throughout the development of my thesis. I came into this project as a complete novice to terrorism and social network analysis. The work involved took me well outside of my comfort zone, and their continued guidance and input helped me continue forward.
# Contents

Abstract ................................................. ii
Preface ................................................... iii

1 Introduction ......................................... 1

2 Related Works ........................................ 4

3 Methods ................................................ 7
   3.1 Data .................................................... 7
   3.2 Problem Definitions ............................... 9
   3.3 Problem Formalization ............................ 12
   3.4 The DESTAB Algorithm ......................... 13
       3.4.1 The DESTAB Polytope ..................... 14
       3.4.2 DESTAB\textsuperscript{app} .................. 16
   3.5 Experimentation .................................. 17
   3.6 Results ............................................. 20

4 Conclusion ............................................ 23
   4.1 Discussion .......................................... 23
   4.2 Next steps .......................................... 24

References .............................................. 27
Chapter 1

Introduction

From 2001 to 2008, the world spent about US$70 billion to increase homeland security, which has reduced the number of transnational terrorism attacks by about 34%. However, during the same period, more lives have been lost to terrorism, equating to 67 more deaths each year. This is due to terrorists responding to the risks imposed by the heightened security measures, and instead, they have focused on more lethal attacks [9]. Counter-terrorism efforts have explored strategies that attempt to mitigate the causes of terrorism through strategic and diplomatic measures [12]. Defeating terrorist groups through strategic means is rare, although it has happened before, such as the Provisional IRA in Ireland and Aum Shinrikyo in Japan [5]. Thus, actions aimed at destabilizing these terrorist networks are still necessary today.

By modeling terrorist networks as graphs, we can represent the underlying structure of these organizations. This can be done on the terrorist level, where individual terrorists are represented as nodes, and the relationships held between them are represented as edges. This can be also be done at the faction level, where factions are organized groups of terrorists with similar ideology and goals. This allows us to leverage an arsenal of social network analysis tools to understand their dynamics and develop potential solutions.
This foundation can be used to understand how terrorist network structures can relate to its efficacy, such as its ability to recruit, attain materials, and carry out attacks. We ultimately want to discover how to dismantle these networks’ efficacy. For the purposes of this paper, we are concerned with the rate at which attacks that are carried out, as it is an objective measure that reflects the key goal of counter-terrorism efforts. In this sense, the rate of attacks corresponds with our notion of lethality that will be used from this point forward.

Given this notion of lethality and how it relates to network destabilization, we want to find the set of actions on a network that minimizes its lethality. In prior work, this typically involves modifying or removing nodes in the network, which corresponds to real-world actions such as capturing or assassinating terrorists.

In this paper, however, we want to see how this can be addressed on faction-level networks. However, there are drastic costs to wiping out factions, which corresponds to a node removal, and it would be difficult to apply the results of such analysis. So, we want to consider another possible action that can be performed on the network: modifying edges. More specifically, we want to change the edge weights, which correspond to the strength of the relationship between the two underlying factions. These relationships can range from being hostile to friendly to anywhere in between. Manipulating faction relations is far more practical in comparison to removing factions altogether. However, the impact of modifying edges with respect to lethality has not been thoroughly researched. We want to see its effectiveness method in reducing lethality and if so, it can be incorporated into methods to destabilize networks.

To assess this, we developed a framework to find the set of node removals and edge modifications on a faction-faction network, with respect to constraints, that minimizes its lethality. Through this framework, we assess the effectiveness of each of these destabilization methods. We seek to find if edge modifications are viable for
destabilization, and if so, how cost-effective it is in comparison to node removals.
Chapter 2

Related Works

Methods of destabilizing terrorist networks have been explored in various ways. *Reshaping Terrorist Networks* by Spezzano et al. investigated how terrorist networks change structurally when one or many terrorists are removed. They modeled organizations such as al-Qaeda and Hamas as terrorist-terrorist networks, and they investigated how the network would respond to node removals, that is, when terrorists are captured or killed. An important dynamic considered here was the rank and role of the removed terrorist, and how they would be replaced by the remaining terrorists in the network. Here, a probability distribution is created to show potential networks that could result from said terrorist removal. This was paired with an algorithm that finds the optimal set of terrorist removals with respect to minimizing lethality. In this case, they quantified lethality with the number of attacks, and the importance of violence-prone and violence-adverse terrorists were used as predictive indicators [11].

Understanding the dynamics of terrorist networks have also been done by Bellutta et al. in *Understanding Shifting Triadic Relationships in the Al-Qaeda/ISIS Faction Ecosystem*. The focus revolved around triadic groups of factions that all had relationships with one another. These relationships could range from negative to positive. They defined a set of hypotheses for how these triadic relationships would evolve
over time. The hypotheses revolved around balance theory, where these groups of relationships would trend towards balance over time. Balanced triadic relationships were generally found to stay balanced over time, but they found that unbalanced relationships would stay unbalanced as well, which goes against the expectations from balance theory [1].

Network destabilization can be measured in ways other than attacks, as seen in Destabilizing Networks by Carley et al.. Instead, they explored the idea of information, that is, a network’s ability to disseminate information, reach consensus, and perform tasks. This notion is more broadly applicable than just terrorist networks. It could be also used for financial systems, social networks, and many other network structures [3]. However, Carley in Destabilization of covert networks drew more focus on terrorist networks. They continued this notion of sharing information and performing tasks to explore destabilization strategies. Here they focused on strategies to target nodes to be isolated from the rest of the network, and they measured the effectiveness of these strategies by measuring the network’s ability to perform tasks [2].

Work has also been done to establish a link between terrorist network structure and its lethality. In Horowitz and Potter’s Allying to kill, they measured factions’ eigenvector centrality to assess how well connected they were within a network, especially towards other well-connected factions. They used linear regressions with an assortment of controls in attempt to establish a causal link, and they found a strong connection between this centrality measure and the number of fatalities the corresponding faction would cause [7].

Chen et al. in Linking Terrorist Network Structure to Lethality also used structure in terrorist networks to establish a link toward lethality. However, this link is more predictive than causal. The paper introduced the Predictive Lethality Analysis of Ter-
terrorist Organization (PLATO) algorithm, which is capable of predicting the number of attacks carried out by a network using its structure alone. Many network structure measures were used, such as strongly connected components, clustering coefficients, and PageRank, and the predictive model was a late-fusion ensemble of simple regression models. They employed this algorithm on a series of terrorist-terrorist networks from each al-Qaeda and ISIS [4].
Chapter 3

Methods

Note: VS Subrahmanian, Chiara Pulice, and Youzhi Zhang each had significant contributions to the definitions and algorithms put forth in sections 3.2-3.4

Section 3.1

Data

The dataset used to model faction-faction networks was collected in [6]. It contains nearly every faction that has been associated with al-Qaeda or ISIS, under the qualification that there is sufficient supportive open-source information. In total, there are 254 factions and 2,729 directional relationships, which are always between two factions.

These relationships between factions are directional as the two underlying factions could each view the relationship differently, and it is possible for the relationship to go in only one direction and not be reciprocated. Each relationship has a start and end date to denote the duration of the relationship. Relationships are given a label that denotes the nature of the relationship (as seen in Table 3.1) and a weight to represent its strength. Weight ranges from -2, which denotes maximum hostility, to 2, which denotes maximum friendliness.
### 3.1 Data Methods

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational</td>
<td>Factions collaborate or one faction supports another on an operational level</td>
</tr>
<tr>
<td>Ideological</td>
<td>A faction exhibits explicit or implicit ideological support to another faction</td>
</tr>
<tr>
<td>Affiliation</td>
<td>One faction aligns with, joins with, or becomes a province of another faction</td>
</tr>
<tr>
<td>Interactive</td>
<td>The two factions have communicated with each other</td>
</tr>
<tr>
<td>Adversarial</td>
<td>A negative relationship between factions that has not escalated to violent conflict</td>
</tr>
<tr>
<td>Conflict-with</td>
<td>Factions have engaged in violent conflict against each other</td>
</tr>
<tr>
<td>Pledged Bayat</td>
<td>A subordinate faction officially pledges bayat, or allegiance, to a dominant faction</td>
</tr>
<tr>
<td>Logistical</td>
<td>A faction exchanges weapons and material with another faction</td>
</tr>
<tr>
<td>Financial</td>
<td>A faction exchanges or provides funds with another faction</td>
</tr>
</tbody>
</table>

Table 3.1: Edge label descriptions provided from the dataset
To build this dataset, a variety of open-source information was collected. Ideally, these sources would be primary sources such as jihadist propaganda and social media posts, intercepted documents created by these factions, and declassified intelligence reports. Secondary sources were also used, such as books, journal articles, newspapers, and other publications. Credibility and potential bias were considered for each source, and these sources were corroborated together to establish a factual basis. The relationships were coded on variables, from timeline to strength, in accordance with guidelines from an internal codebook [6].

The dataset used to assess attacks carried out by these terrorist networks was collected from the Global Terrorism Database, an open-source database of terrorist events between 1970 and 2019 all around the world. The database has over 200 thousand attacks recorded, each containing information about the event, like date, location, the perpetrators of the attack, the nature of the attack, casualties, and more. A variety of open media sources were used to gather information about these terrorist events. However, information was not added until the corresponding sources were deemed credible [8].

Section 3.2  
Problem Definitions

We assume the existence of a set of factions $\mathcal{F}$. We leverage a network derived from the dataset in [6] of the AQIS ecosystem, which consists of 143 factions. We also assume the existence of a set of edge labels $\mathcal{L}$, which can capture virtually any type of faction-faction relationship. The specific edge labels used within the dataset are shown in Table 3.1.

We assume the existence of a discrete weight interval $\mathcal{W}_K = \{-K, -K+1, \ldots, K-1, K\}$ for some integer $K$. In our AQIS dataset, $K = 2$, i.e. $\mathcal{W} = \{-2, -1, 0, +1, +2\}$.
3.2 Problem Definitions

Definition 3.1 (Edge). An edge is a 4-tuple \((f_1, f_2, \ell, w)\) where \(f_1, f_2 \in \mathcal{F}\) are factions, \(\ell \in \mathcal{L}\) is a label, and \(w \in \mathcal{W}\) is an edge weight.

Intuitively, an edge has four parts. The relationship \(\ell\) provided by faction \(f_1\) to faction \(f_2\) and the strength of that relationship. If we look at the edge from node B to node A in Figure 3.1, the edge label “Ideological” suggests that B provided ideological support to A. The weight of 2 suggests that B is strongly positive toward A.

Definition 3.2 (Faction-Faction Network). A faction faction network (FFN for short) is a pair \((\mathcal{F}, E)\) where \(E\) is a finite set of edges such that the following restriction is
3.2 Problem Definitions

Thus, an FFN requires that there can only be one edge from faction $f_1$ to faction $f_2$ having with a given edge label. We cannot have two such edges from faction $f_1$ to faction $f_2$ with the same label but different weights.

**Definition 3.3** (Change Constraint Function). A change constraint function $cc$ takes a triple of the form $tr = (f_1, f_2, \ell)$ as input and returns an interval $[LB_e, UB_e]$. That is, the first three components of an edge determine the weight interval $[LB, UB]$.

When destabilizing a network by changing the relationships between factions, we need to be bound by prevailing political or other realities. For instance, counter-terrorism analysts may deem it unlikely that the adversarial relationship between A and B can be changed. Or they may believe it can be changed from a -2 (current) to a -1 but not to anything better than that. The definition of change constraint functions above captures this intuition.

**Definition 3.4** (Edge Cost Function). An edge cost function is a partial mapping $\chi$ from a pair of edges $e, e'$ to the non-negative real numbers. The function is defined if and only if $e, e'$’s first 3 components are all the same, i.e. $e.f_1 = e'.f_1$ and $e.f_2 = e'.f_2$ and $e.\ell = e'.\ell$. In addition, we require that:

(a) $e'.w \notin cc(e.f_1, e.f_2, e.\ell) \rightarrow \chi(e, e')$ is not defined. This ensures that we cannot change edge $e$ to edge $e'$ in those cases when $e'$ assigns a weight in violation of the change constraint function.

(b) $e.w \leq e'.w \leq e''.w \rightarrow \chi(e, e') \leq \chi(e, e'')$. This ensures that the more we increase the weight, the greater the cost.
3.3 Problem Formalization

(c) \( e.w \leq e'.w \leq e''w \Rightarrow \chi(e'', e') \leq \chi(e'', e) \). This ensures that the more we decrease the weight, the greater the cost.

The intuition in the last 2 clauses in the definition of edge cost functions is that a larger change has a higher cost (regardless of whether the change is to increase or decrease friendships) than a smaller change. For example, changing the adversarial relationship between C and D in Figure 3.1 from a −2 to −1 is cheaper than changing it to 0 which in turn is cheaper than changing it to +1.

Definition 3.5 (Node Removal Constraint). A node removal constraint \(NRC\) is a finite set of nodes (which cannot be removed from the network).

We can always target a network by effectively neutralizing a node. While neutralizing an entire terror group such as Al-Qaeda or ISIS may be impossible, it may be possible to neutralize a small faction — at least for all practical purposes. This can be captured as a node removal operation. But not all nodes are going to be neutralizable in such a fashion and our node removal constraint identifies such nodes.

Definition 3.6 (Node Cost Function). We assume the existence of a node cost function \(\eta\) which is a mapping from nodes in FFN to real numbers.

Section 3.3

Problem Formalization

We are now ready to define the basic problem we wish to solve.

Problem 1 (FFN Reshaping Problem). Given a faction to faction network \( FFN(\mathcal{F}, E) \), a change constraint function \( cc \), a change cost function \( \chi \), a node removal constraint \( NRC \), a node cost function \( \eta \), and a bound \( C \) on the total cost allowed, the FFN Reshaping problem is one that finds a new faction faction network \( FFN' = (\mathcal{F}', E') \) such that:
(a) \( F' \subseteq F \), i.e. no new factions are introduced, and

(b) \( NRC \subseteq F' \), i.e. no factions whose removal is forbidden are removed, and

(c) For all \( e' \in E' \), if \( e' = (f_1, f_2, \ell, w') \), then there exists an edge \( e = (f_1, f_2, \ell, w) \) such that \( w' \in cc(f_1, f_2, \ell) \). This says that each edge \( e' \) in the reshaped network has a near identical edge (i.e. the first 3 components of an edge) in the original network with a different weight.

(d) \( \left( \sum_{e \in E \land e' \in E' \land e.f_1 = e'.f_1 \land e.f_2 = e'.f_2 \land e.\ell = e'.\ell} \chi(e, e') \right) + \left( \sum_{f \in \mathcal{F} - \mathcal{F'}} \eta(f) \right) \leq C \). The sum of the costs of removing nodes and changing edges must lie within the budget.

In this case, \( FFN' \) is called a solution to the FFN Reshaping Problem.

**Problem 2 (Optimal FFN Reshaping Problem, OptFFN).** Given an FFN Reshaping problem and a lethality function \( \text{lf} \), the optimal FFN Reshaping decision problem is to check if there exists a solution to the FFN Reshaping Problem whose lethality \( \text{lf}(FFN') < \text{lf}(FFN) \).

The Optimal FFN Reshaping Search Problem is to find a solution to the FFN Reshaping Problem whose lethality is minimized.

Throughout this paper, we assume the existence of a lethality function \( \text{lf} \) that can take any terror network as input and that returns a non-negative number as output. Such lethality functions have been previously proposed in [10, 11]. When reshaping a faction-faction network, we want a reshaped network whose lethality is at least smaller than that of the original, and preferably the minimal lethality possible.

### Section 3.4

**The DESTAB Algorithm**

In this section, we propose \( \text{DESTAB} \), an architecture to address the OptFFN problem. We then propose the \( \text{DESTAB} \) algorithms to address the OptFFN problem. \( \text{DESTAB} \)
has 2 major phases. The first phase creates what we call a DESTAB Polytope which uses integer programming to find the feasible space of solutions to a relaxation of the FFN Reshaping Problem. Each point in this polytope is a possible new (reshaped) network $FFN'$. However, selecting a solution from this space which minimizes the lethality of the resulting network is challenging because a regression model is used to estimate lethality. We propose generating an approximate solution, $DESTAB^{app}$, by using a random walk amongst the integer points inside the DESTAB Polytope, applying the regression function learned by [10, 11], and choosing the best one. We start by describing the DESTAB Polytope, followed by the algorithms.

### 3.4.1. The DESTAB Polytope

In this section, we define a set of constraints that jointly define a convex polytope from which the DESTAB algorithms will find a solution to the OptFFN problem. We use a set of variables shown in Table 3.4.1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Ranges Over</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_i$</td>
<td>{0, 1}</td>
<td>Should faction $f_i$ be removed or not? One such variables for each $f_i \in F$</td>
</tr>
<tr>
<td>$c_{i,j,\ell,w,w'}$</td>
<td>{0, 1}</td>
<td>Should the edge $e = (f_i, f_j, \ell, w) \in E$ be changed to have weight $w'$ where $w' \in cc(e)$.</td>
</tr>
</tbody>
</table>

From these variables, we can write down the following constraints.

\begin{align*}
    r_i &= 0 \text{ for each } f_i \in NRC \quad (3.1) \\
    \sum_{w' \in cc(e)} c_{i,j,\ell,w,w'} &= 1 \text{ for each edge } e = (f_i, f_j, \ell, w) \in E \quad (3.2)
\end{align*}

Constraint 3.1 ensures that the non-removability constraints are preserved, and con-
3.4 The DESTAB Algorithm

Methods

Constraint 3.2 ensures that the change constraints are preserved. This constraint additionally ensures that only one edge of the form \((f_i, f_j, \ell, -)\) is present in the reshaped network.

\[
C_r = \sum_{f_i \in F} r_i \times \eta(f_i) \tag{3.3}
\]

\[
C_c = \sum_{e=(f_i, f_j, \ell, w) \in E} c_{i,j,\ell,w,w'} \times \chi((f_i, f_j, \ell, w), (f_i, f_j, \ell, w')) \tag{3.4}
\]

\[
C_r + C_c \leq C \tag{3.5}
\]

Constraint 3.5 ensures that the total cost \(C\) is not exceeded by the proposed reshaping actions. In addition, we have the following two obvious constraints which say that the node removal and edge change variables are all in the \([0, 1]\) interval.

\[
r_i \in [0, 1] \text{ for all } f_i \in F \tag{3.6}
\]

\[
c_{i,j,\ell,w,w'} \in [0, 1] \text{ for all edges } (f_i, f_j, \ell, w) \in E \wedge w' \in cc(e). \tag{3.7}
\]

We remove at most \(k_r\) factions and change at most \(k_c\) edges:

\[
\sum_{f_i \in F} r_i \leq k_r \tag{3.8}
\]

\[
\sum_{e=(f_i, f_j, \ell, w) \in E} c_{i,j,\ell,w,w'} \leq k_c \tag{3.9}
\]

We use \(\text{CONS(FFN)}\) to denote the set of all constraints having the forms shown in equations 3.1, 3.2, 3.5, 3.6, 3.7, 3.8, and 3.9 above. It is important to note that even though it is our intention that the \(r_i\) and \(c_{i,j,\ell,w,w'}\) variables are all binary \(\{0, 1\}\) variables, we use \(\text{CONS(FFN)}\) to denote, for now, their linear relaxation. As a consequence, \(\text{CONS(FFN)}\) generates a convex polytope which contains all possible
(integer) solutions to our OptFFN problem as well as other non-integer solutions.

3.4.2. DESTAB\textsuperscript{app}

To find an approximate solution to the Optimal FFN Reshaping Problem, a random walk algorithm was employed, as seen in Algorithm 1. It receives a binary integer array as input, which represents the variables defined in DESTAB Polytope. The values of the input array $x$ correspond to the initial FFN, and to complement the array we have the constraint function $\text{CONS}$, which enforces all of the aforementioned constraints, and a lethality function $Leth$. Given the input array $x$, we use it to initialize a 2-dimensional array $X$, which will contain each instance of DESTAB Polytope over the course of the random walk. Alongside $X$, an array $L$ is initialized, where the element $L_i$ corresponds to the lethality of $X_i$.

After the initialization step, the walk begins. The walk will last for $T$ steps, and each step starts with an integer variable $x_i$ being selected at random. Once a variable is selected, it is then flipped from 0 to 1 or vice versa. However, if it is an edge variable $c_{i,j,t,w,w'}$, then the corresponding active variable $c_{i,j,t,-,-}$ is also flipped to satisfy constraint 3.2.

After the variable flip, we check if $x$ violates the constraints defined by CONS. If so, the flip is reversed and we continue to the next iteration. If not, then the lethality of the network induced by $x$, $Leth(x)$, is calculated and compared to the lethality of the previously defined network, $Leth(X_t)$. The probability that this new network is accepted is $\min\{1, e^{-(Leth(x) - Leth(X_t))/m}\}$, where the value of $m$ is based on the magnitude of lethality. If the lethality of the candidate is lower, it is always accepted. However, if the candidate increases lethality, the probability of acceptance exponentially depends on the magnitude of the increase. If the candidate $x$ is accepted, then it is appended to $X$, and if not, a copy of the previous array $X_t$ is appended instead. Then the lethality of the appended array is appended to $L$. After all this, $t$ is
incremented, and the loop starts over.

After the while loop concludes, the binary integer array in $X$ with the lowest lethality is returned alongside its lethality.

**input**: $n$ binary integer variables, $x_i \in \{0, 1\}$, $1 \leq i \leq n$, the constraint function $\text{CONS}$, the lethality function $\text{Leth}$

1. Initialize $X_0$ to represent the initial FFN. Note that $X_{0,i}$ is the value of $i$-th integer variable;
2. Initialize array $L_0$ with $\text{Leth}(X_0)$;
3. $t \leftarrow 0$;
4. while $t < T$ or other conditions do
   1. Randomly choose an integer variable $x_i$;
   2. $X'_{t+1,j} = X_{t,j}$ if $j \neq i$; otherwise, $X'_{t+1,i} = |1 - X_{t,i}|$;
   3. if $X'_{t+1}$ satisfies $\text{CONS}(\text{FFN})$ then
      1. Accept or reject the candidate point $X'_{t+1}$ with probability
         $\min\{1, e^{-({\text{Leth}}(X'_{t+1})-\text{Leth}(X_t))/T_t}\}$;
      2. $X_{t+1} \leftarrow X'_{t+1}$; if accepted, otherwise, $X_{t+1} \leftarrow X_t$;
      3. $L_{t+1} \leftarrow \text{Leth}(X'_{t+1})$; if accepted, otherwise, $L_{t+1} \leftarrow \text{Leth}(X_t)$;
      4. $t \leftarrow t + 1$;
   4. else
      1. $X_{t+1} \leftarrow X_t$;
   5. end
5. end
6. return $X_{\text{argmin}(L)}, \min(L)$.

**Algorithm 1**: Random Walk

### Section 3.5

#### Experimentation

The first step was to take the AQIS dataset from [6] and develop a temporal sequence of networks. These networks will be used to train a lethality function, and the last network in this sequence will be used in the FFN Reshaping Problem. For every month in the time range of the dataset, January 1990 to December 2017, the active relationships were extracted, and those relationships were used to define the edges in the network. The nodes in the network were derived from the defined edges. This
3.5 Experimentation

Figure 3.2: Prediction results from the PLATO algorithm on our network data. Note that the model performed best predicting the number of attacks per month.

resulted in a sequence of 336 networks, where the start and end date were a month apart, but many of them, especially early on, did not change from month to month. So, when two temporally adjacent networks were identical, they would be merged together and the start/end date would be expanded. This resulted in 174 networks spanning between January 1990 and December 2017, averaging a size of 84.58 factions and 61.68 relationships.

In order to develop the lethality function, we gathered data on the attacks carried out by the sequence of networks. We used the dataset from [8] to find attacks that occurred during the time frame of each network and was carried out by a faction in the network. In total, we found 23,384 attacks that were carried out by these networks, averaging 131.16 attacks per month across all networks.

Using the PLATO algorithm, we trained a model on our sequence of networks to predict the number of attacks a future network would carry out. We also considered the number of kills and number of casualties as our target metric, but the PLATO algorithm performed best predicting the number of attacks, where it achieved a Pearson correlation coefficient of 0.946.

While in pursuit of finding an approximate solution to the problem put forth, we also wanted to compare the effectiveness of different destabilization methods with respect to reducing the lethality. That is, alongside the method of node removals, which was observed on terrorist-terrorist networks in [10], the method of modifying
edge weights is also considered, and we can observe whether this new method is effective, both on its own and in conjunction with removing nodes.

Thus, multiple iterations of the random walk solver were run. For all runs, we always used the most recent network from the AQIS dataset so that the lethality model could train on all prior networks. The only parameters that changed between each run were the constraints $k_r$ and $k_c$, which allowed us to assess the relative effectiveness of the two methods. The other parameters that were held constant are as follows:

(a) The number of random walk steps $T$ was set to 500.

(b) The budget $C$ was set to $\infty$. This is to allow the constraints $k_r$ and $k_c$ to be the only limiting factors for modifying the network.

(c) The change constraint function $cc$, which provides a weight interval based on an edge’s properties, returns either $[0, 2]$ or $[-2, 0]$ based on the edge label. More precisely, it returns $[0, 2]$ if the label denotes a positive relationship and returns $[-2, 0]$ if it denotes a negative relationship, as described in Table 3.1.

(d) The node removal constraint $NRC$ was an empty set, which means that any node in FFN can be removed.

(e) The edge cost function $\chi$ and node cost function $\eta$, for the purposes of the optimization problem, always returned 0. Similarly to the budget, this is to allow $k_r$ and $k_c$ to be the only limiting factors.

For the constraints $k_r$ and $k_c$, the values $\{0, 1, 2, 3\}$ and $\{0, 1, 2, 3, 4, 5\}$ were used respectively, and every combination of these constraints were used in one run of the random walk except $(0, 0)$, i.e. $(0, 1), (0, 2), \ldots, (3, 4), (3, 5)$.
From these iterations of the random walk solver, we got the provided constraints $k_r$ and $k_c$ as well as the lowest lethality obtained from the walk. In addition, the network modifications corresponding to the lowest obtained lethality were also provided. We also receive the lethality of the initial network before any modifications are performed, which was 292.81 attacks per month.

First, we observe the isolated effect of each node removals and edge modifications. When only performing edge modifications, we find that each one reduces the lethality of the network by 1.26 attacks per month. We also find that edge modifications have diminishing returns as more are performed. In fact, the best result occurred when only 2 edges were modified, which led to a reduction in the lethality by 3.45 attacks per month. For runs that had more than 2 edge modifications, all but one proved to be slightly less effective in reducing lethality, ranging from 2.32 to 2.94 fewer attacks per month. Surprisingly the worst result occurred when attempting to modify 5 edge weights, which was 0.71 fewer attacks per month.

When only performing node removals, we find that each one reduces the lethality

![Figure 3.3: The results of the random walk when only node removals or edge modifications were performed. Note the difference in scale between the two graphs](image-url)

20
of the network by 34.20 attacks per month on average. Unlike the method of edge modifications, we do not observe diminishing returns when more nodes are removed, the effect is nearly linear. When 3 nodes are removed from the network, the lethality of the network is reduced by 90.34 attacks per month.

When a mix of edge modifications and node removals are performed, we find the results were slightly worse relative to when only node removals were performed. For each node removed, regardless of the number of edge modifications, the lethality of the network would decrease by 25.70 attacks per month on average. However, it is difficult to assess the effect of edge modifications in this situation, as it is inherently mixed with the effects of node removals and there is variability due to the random nature of the solver. It seems that it is more difficult for the solver to find an optimal solution when the modification constraints increase, which resulted in diminishing returns. This is likely due to the increased search space, so it is more difficult to converge on an optimal network. This was likely a significant factor when a mix of edge modifications and node removals are allowed. In the construction of the problem, where the network is represented as an array of binary integers, the vast majority of the variables correspond to edge modifications, 85.13% to be precise, but the most impactful modifications are node removals. So, given the same number of steps with a random walk, it will be more difficult for a mixed search to find the best set of nodes to remove as opposed to a pure node removal search.

Another important thing to note is that the node removal constraint, $NRC$, was an empty set, so any faction could be removed from the network. This means that significant factions such as al-Qaeda or ISIL were subject to removal, and in fact, some of these significant factions were removed in some of the solutions returned from the random walks.

That being said, our results show that node removals are on average 27.14 times
more effective than edge modifications with respect to reducing the lethality. While it is intuitive that removing a faction would have more impact, this means that if the cost of modifying a relationship between factions is at least 27.14 times cheaper than removing a faction, then it is a more cost-effective option.
Chapter 4

Conclusion

Section 4.1

Discussion

It is evident that modifying relationships between terrorist factions is effective in reducing the lethality of a terrorist network. This is a significant realization, and it calls for further investigation to better understand how these dynamics operate, in theory, and in reality.

While the impact is a magnitude smaller than removing a terrorist faction, it is necessary to consider in cost implications of these two actions. In practice, removing a terrorist faction would mandate billions in $USD of financial investment, risk countless lives, and there is no guarantee of success. In fact, the chances of successfully removing a terrorist organization are slim in practice. Furthermore, the notion of removing a faction in the DESTAB framework does not effectively capture the externalities of removing a faction.

On the other hand, modifying a relationship between two factions is far more practical. Costs could be in the magnitude of millions of $USD, and there is minimal human risk involved, as these sorts of actions could be performed remotely in a safe
location. In this sense, modifying relationships between factions is *thousands of times* more cost-effective than removing factions. This could be a quality counter-terrorism option in the future, barring a more precise understanding of its dynamics.

Another note of consideration is that we did not restrict the nodes that could be removed during the random walk, so large and significant factions such as al-Qaeda were subject to removal. In many cases, this tier of factions was in fact removed in the random walk, so when we consider the comparison between the two destabilization methods, removing factions and manipulating faction relations, we are comparing the relative efficacy of modifying relationships to the removal of the most powerful terrorist factions. This suggests that the relative cost-effectiveness may be even greater than what the results reveal.

However, it is necessary to understand that these comparisons of cost-effectiveness should be taken with a grain of salt. We assumed a uniform cost for both node removals and edge modifications, but in reality, these costs would undoubtedly vary. Furthermore, we do not have constraints that limit the actions that can be taken on the network, which would reflect their feasibility, so we may be comparing actions that are not possible or prohibitively expensive. This does not take away from the fact that manipulating relationships between factions is a viable destabilization strategy, but the relative cost-effectiveness should not be taken at face value.

### Section 4.2

**Next steps**

Given the findings from the experimentation, the most important step is to further investigate this notion of modifying faction relationships. First, it is necessary to develop a lethality function that is better adjusted to faction-faction networks and modifying edge weights. In the above experimentation, we used the PLATO algo-
4.2 Next steps

Although it did perform well on the networks in our dataset, yielding a PCC of 0.946, the feature set was not built to leverage varying edge weight values, and the only way edge weights impact feature extraction was through the PageRank algorithm, which is performed on the unrestricted network.

Therefore, work needs to be done to develop features that leverage edge weights, so their impact can be better captured. Furthermore, factions could be attributed characteristics beyond those derived from edge labels, which could then be leveraged for feature extraction. For example, when PLATO was being developed for terrorist-terrorist networks, the rank and status (i.e. alive, imprisoned, or dead) of terrorists were utilized. Adding more information to characterize factions, such as their size or resources at hand, could be helpful to enrich the feature set.

Work with respect to the reshaping problem also needs to be done. In our experimentation, we did not utilize the cost constraints because we did not have well-defined cost functions for edge modifications and node removals. During preliminary experimentation, we used very simplistic functions and did not consider the underlying factions. Focused work on each of these functions, so that they can adequately estimate the cost behind these destabilization methods, will allow for the DESTAB framework to be more practical in usage.

In a similar sense, we did not leverage the node removal constraint nor the edge change constraint to limit the actions that can be taken against the network. In reality, many of these actions are not feasible, or at least prohibitively expensive. Work needs to be done to systematically or manually define these constraints.

One limitation within the current framework is that it assumes that all of the actions would be performed simultaneously, that is, within a single time step, factions would be removed and relationships would be manipulated. This assumption is a bit idealistic, as it takes time to carry out these actions, and each action may impact the
cost and feasibility of others. It may be useful to consider a framework that finds the optimal sequence of actions, where the ordering of the actions would affect the final network lethality, even if the final network is identical.

While there is still a lot of work that needs to be done before the DESTAB framework is applied to the real world. We can find appreciation in the fact that it found viability in manipulating relationships between factions. It is now necessary to determine the extent in which this method can be employed.
Bibliography


