2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

STONE: Shaping Terrorist Organizational Network Efficiency

Francesca Spezzano
DIMES
Università della Calabria
Rende (CS), Italy 87036
Email: fspezzano@dimes.unical.it

V.S. Subrahmanian
UMIACS & Department of Computer Science
University of Maryland
College Park, MD 20742
Email: vs@cs.umd.edu

Aaron Mannes
School of Public Policy
University of Maryland
College Park, MD 20742
Email: amannes@umd.edu

***Abstract*—This paper focuses primarily on the Person Successor Problem (PSP): when a terrorist is removed from a terrorist network, who is most likely to take his place? We leverage the solution to PSP to predict a new terrorist network after removal of a set of terrorists and to answer the question: which set of k ($k > 0$) terrorists should be removed in order to minimize the lethality of the terrorist network? We propose a theoretical model to study these questions taking into account the fact that terrorists may have different individual capabilities. We develop an algorithm for PSP in which analysts can specify the conditions an individual needs to satisfy in order to replace another person. We test the correctness of our algorithm on a real-world partial network dataset for two terrorist groups: Al-Qaeda and Lashkar-e-Taiba where we have ground truth about who replaced who, as well as a synthetic dataset where experts estimate who replaced who. Building on the solution to PSP, we develop an algorithm to identify which set of k people to remove from a terrorist network to minimize the organization's efficiency (formalized as an objective function in some different ways).**

I. Introduction

There are numerous security applications where an organizational network needs to be *reshaped*. For instance, an intelligence agency may wish to reshape a terror group's internal structure by removing certain terrorists. Such removals may include arrests, targeted killings, providing financial inducements to leave a theater of operations, and so forth.

Organizational networks are networks whose vertices are people. They are different from social networks in many ways. Individuals within a human network have "ranks" or functions within an organization — moreover, individuals also have properties. For instance, it is unlikely that when a bomb-maker in an organization is removed, that he will be replaced by a fiery clerical orator even if the latter has the highest centrality. The replacement must be a qualified bomb-maker. Thus, when a person is removed from an organizational network, the person's position in the organization is usually assumed by another person. Removing a person from a network on the basis of classical centrality measures is likely wrong for the following reasons.

1) Removing a person p from an organizational network may cause the person to be replaced by a person who is more competent and/or more lethal. We don't want this.
2) Removing a person from a network causes a cascading effect. That person's role in the organization is filled by another person whose position in the network must be filled by yet another person and so forth.
3) All individuals in an organizational network have roles, and properties. All these factors play a role in determining who should be removed, and when. For instance, a fiery orator is not likely to be replaced by a bombmaker even if the two have high ranks in the organization.

In this paper, we also answer the question. (i) *Given an organizational network $\mathcal{ON}$ and a vertex v to be removed, how does the network re-organize itself? Who assumes the position of the removed individual in the organization?* We will return the answer as a probability distribution over a set of possible resulting new networks. We then extend the answer to the case where instead of removing a single person p, we remove a set X of vertices from the network. (ii) Second, given a measure of the lethality of a network and a desire to eliminate up to $k, (k > 0)$ people from the network, which set of k people should be eliminated so as to minimize the *expected lethality* of the resulting network?

The paper is organized as follows. Section II presents a toy example, which is one of the 50 toy examples we presented to two experts on terrorism (an author of a well-known terrorism book and a retired US general) as well as to 3 students. This example is used as running example in the paper. Section III provides a formal definition of an organizational network. Section IV presents the theoretical model we developed to predict who will replace a vertex that is removed. The theoretical framework has several different parameters which users can set. Section V extends the framework to the case of multiple vertices removal, and induces a pdf on the space of all possible networks that could result. Section VI presents the STONE-Predict algorithm to compute this pdf, and presents the STONE-Reshape algorithm to re-shape a network, given a selection of the parameters in the STONE prediction model of Section IV. Section VII briefly describes our implementation of STONE-Predict and shows that our algorithm is over 80% accurate on three data sets: the synthetic data, Sageman's Al-Qaeda network [14], and a Lashkar-e-Taiba network [15].

II. Some Examples

We present one of the 50 toy examples used to test our STONE algorithms. Note that our experiments were conducted

ASONAM'13, August 25-29, 2013, Niagara, Ontario, CAN

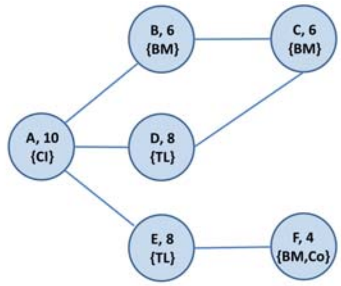

Fig. 1. Example organizational network

both on the synthetic data as well as real world network data on Al Qaeda from Sageman's [14] and Lashkar-e-Taiba [15]. In both cases, there is a wealth of data on removed individuals and who replaced them.

Each vertex has a rank (integer) in the terrorist organization. The higher the rank, the more important the vertex is. In addition, each vertex may have one or more properties (e.g. `leader, bomb-maker`, etc).

Example 1: Figure 1 shows a network with vertices A, B, C, D, E, F where each vertex has a set of properties (`top-leader (TL), chief-of-ideology (CI), bomb-maker (BM), courier (Co)`). When the leader of this network (node A) is removed, who will replace A? Our two experts said that person D would be the new leader, the reason being that D's rank is the same as E's, but D is connected to higher rank people (C, and through C to B) than E. □

III. Organizational Network: Formalization

In this section, we formally define an organizational network. Throughout this paper, we assume the existence of a finite set VP of *vertex properties*. Each vertex property $p \in \mathsf{VP}$ has an associated domain $dom(p)$ of values.

Definition 1 (Organizational network): An *organizational network* $\mathcal{ON}$ is a 4-tuple $(\mathtt{V}, \mathtt{E}, wt, \wp)$ where:

1) $\mathtt{V}$ is a finite set whose elements are called *vertices*.
2) $\mathtt{E} \subseteq \mathtt{V} \times \mathtt{V}$ is a finite multi-set whose elements are called edges. We assume that if $(u, v) \in \mathtt{E}$, then $(v, u) \in \mathtt{E}$, i.e. edges are undirected.
3) $wt : \mathtt{V} \to \mathbb{N}$ is a *weight* function.
4) $\wp : \mathtt{V} \times \mathsf{VP} \to \bigcup_{p \in \mathsf{VP}} dom(p)$ is a function, called *property* such that $(\forall p \in \mathsf{VP})\wp(v, p) \in dom(p)$. □

Intuitively, the vertices in $\mathcal{ON}$ are people. Edges are undirected connections between people. The weight $wt(u)$ of a vertex u expresses its rank - the higher the weight, the more important the vertex. Finally, $\wp(u, p)$ returns the value of a property p for a vertex u. For instance, if the property p is `bomb-maker`, then $\wp(u, p)$ is either a 0 (not a bomb-maker) or 1 (is a bomb-maker). On the other hand, if property p were `clustering_coefficient` [16] then $\wp(u, p)$ might return numerical values.

Given an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$, vertex $u \in \mathtt{V}$, and a set $\mathtt{E}' \subseteq \mathtt{E}$ of edges, we use $nbr(u)$ to denote the set of all immediate neighbors of u, i.e. $nbr(u) = \{v \in \mathtt{V} \mid (u, v) \in \mathtt{E} \; or \; (v, u) \in \mathtt{E}\}$.

When a vertex is *removed* from $\mathcal{ON}$, many changes can occur. For instance, removing Osama Bin Laden from the Al Qaeda network led to a reassignment of ranks (e.g. Ayman Al Zawahiri's rank was changed). Thus, when a vertex is removed from an organizational network, both the topology of the network (the $\mathtt{V}, \mathtt{E}$ part) and the weights/properties of the vertices in the network can change. The term *deletion* refers just to the changes in the structure of the organizational network.

Definition 2 (Vertex deletion): Suppose $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ is an organizational network and $\mathtt{V}' \subseteq \mathtt{V}$. The *result of deleting* $\mathtt{V}'$ *from* $\mathtt{V}$ is the organizational network $\mathcal{ON}^\bullet = (\mathtt{V}^\bullet, \mathtt{E}^\bullet, wt^\bullet, \wp^\bullet)$ defined as follows. (i) $\mathtt{V}^\bullet = \mathtt{V} - \mathtt{V}'$; (ii) $\mathtt{E}^\bullet = \mathtt{E} - \{(u, v) \mid (u, v) \in \mathtt{E} \wedge (u \in \mathtt{V}' \vee v \in \mathtt{V}')\}$; (iii) $wt^\bullet(u) = wt(u)$ for all $u \in \mathtt{V}^\bullet$ and (iv) $\wp^\bullet(u, p) = \wp(u, p)$ for all $u \in \mathtt{V}^\bullet$ and $p \in \mathsf{VP}$. □

Vertex *replacement* is similarly defined.

Definition 3: Suppose $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ is an organizational network and $u, v \in \mathtt{V}$. The *result of replacing* v *with* u is the organizational network $\mathcal{ON}^\bullet = (\mathtt{V}^\bullet, \mathtt{E}^\bullet, wt^\bullet, \wp^\bullet)$ defined as follows. (i) $\mathtt{V}^\bullet = \mathtt{V} - \{v\}$; (ii) $\mathtt{E}^\bullet = (\mathtt{E} - \{(x, y) \mid (x, y) \in \mathtt{E} \wedge x = v \; or \; y = v\}) \cup \{(u, x) \mid (v, x) \in \mathtt{E} \wedge x \neq u\} \cup \{(y, u) \mid (y, v) \in \mathtt{E} \wedge y \neq u\}$; (iii) $wt^\bullet(v') = wt(v')$ for all $v' \in \mathtt{V}^\bullet$; (iv) $\wp^\bullet(v', p) = \wp(v', p)$ for all $v' \in \mathtt{V}^\bullet$ and all $p \in \mathsf{VP}$ that are intrinsic node properties. □

Intuitively, replacing v with u means that u assumes v's position in the network and adopts all of v's links (in addition to its own) but retains its own intrinsic properties (e.g. gender, date of birth), while network-related properties (like clustering coefficient) might change.[1]

Example 2: For instance, the result of deleting the vertex A from the organizational network shown in Figure 1 is shown in Figure 2. On the other hand, the result of replacing the vertex A with the vertex E in Figure 1 is shown in Figure 3.

IV. Person Successor: One Vertex Removal

In this section, we consider the case when exactly one vertex v is removed from an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$. We make four assumptions.

Assumption I. We assume that the replacement vertex u is not too far away from v in $\mathcal{ON}$.
Assumption II. We assume that the probability that v is replaced by a vertex u depends on u's rank in the network.
Assumption III. We assume that individuals with a higher weight than v will not be seeking v's position.
Assumption IV. We assume that $\mathcal{ON}$ will reshape itself to be maximally lethal and hence, when v is removed from the network, u is selected in a way that maximizes lethality (formally defined in Section VI-B).

[1] A slight alternative definition would set $wt(u) = wt(v)$ so u assumes v's rank. The results in this paper stand even if this modification is made.

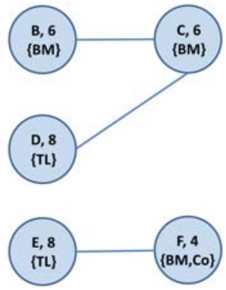

Fig. 2. Example of deletion from an organizational network

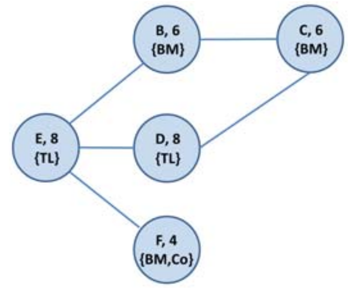

Fig. 3. Example of replacement in an organizational network

A. Replacement Value of Vertices

Suppose we wish to remove vertex r. Each vertex u in $\mathcal{ON}$ has a *replacement value* $rv(r,u)$ which quantifies whether u is an appropriate replacement for r. We define $rv(r,u)$ based on three factors. (i) Can u fulfil r's role in the organization? This is defined in terms of the properties of r and u. (ii) Does u have a weight (rank in the organization) that is lower than r? (iii) How well connected is u to the rest of the network?

To measure "connected" in (iii) above, we define the new notion of *Weighted Removal Pagerank* which considers the vertex being removed in its calculation.

Definition 4 (Weighted Removal Pagerank): Suppose $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ is an organizational network and $r \in \mathtt{V}$ is a vertex considered for removal. The *Weighted Removal Pagerank* $WRP(v,r)$ of a vertex $v \in \mathtt{V} - \{r\}$ w.r.t. the vertex r is

$$WRP(v,r) = (1-\delta)\frac{wt(v)}{\sum_{u \in \mathtt{V}\setminus\{r\}} wt(u)} + \delta \sum_{u \in nbr(v)\setminus\{r\}} \frac{WRP(u,r)}{|nbr(u) \setminus \{r\}|}$$

where $\delta \in [0,1]$ is a damping factor.[2] □

We now define the replacement value of a vertex.

Definition 5 (Replacement value): Suppose $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ is an organizational network and $r \in \mathtt{V}$ is a vertex. The *replacement value* $rv(r,v)$ of $v \in \mathtt{V} - \{r\}$ is defined as

$$rv(r,v) = \alpha_0 \cdot WRP(v,r) + \sum_{p_i \in VP} \alpha_i \cdot \wp(v, p_i)$$

where $\sum_{i=0}^{n} \alpha_i = 1$. □

In this definition, α_i ($i > 0$) reflects the weight of each property p_i of vertex v. The relative importance of WRP with respect to the intrinsic properties of the vertex is given by the ratio $\frac{\alpha_0}{\Sigma_{p_i \in \mathsf{VP}} \alpha_i \wp(v,p_i)}$. Thus, if $\alpha_0 = 0.8$ and $\Sigma_{p_i \in \mathsf{VP}} \alpha_i \wp(v,p_i) = 0.2$, then a person's connections are much more important than his intrinsic properties. [3]

We are now ready to define the problem of who replaces a person who is being removed from the network.

Person Successor Problem. Suppose $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ is an organizational network, $r \in \mathtt{V}$ is a vertex, and C is a logical condition on VP. The *person successor problem* is the problem of finding a vertex $v \in \mathtt{V}$ such that:

1) v satisfies condition C and
2) $wt(v) < wt(r)$ and
3) there exists no $v' \in \mathtt{V} - \{r\}$ satisfying the previous two conditions such that $rv(r,v') > rv(r,v)$.

We allow analysts to specify a logical condition C limiting who can replace a removed individual r. For instance, if r is a `bomb-maker` (i.e. $\wp(r, \texttt{bomb-maker}) = 1$), then replacing him with a `courier` would not make sense. Thus, the expert may specify that the replacement node should have the `bomb-maker` property.

B. Vertex Properties

Though STONE can support any set of vertex properties, in our experiments we considered: `roles, hostility, volatility, local clustering coefficient`.[4] Roles include things like `ideolog, fund-raising, financier, militant, social-services, recruiting` etc. We allow people to have multiple roles - for instance, in the case of Al-Qaeda, Ayman Al Zawahiri's role could include virtually all of the above. Hostility refers to the hostility of the person towards the West, while Support refers to his support for and capability to execute terror actions.

We generalize the definition of local clustering coefficient in two ways — we don't just look at immediate neighbors, and we take the properties of vertices into account.

Definition 6 (Property oriented clustering coefficient (POCC)): Suppose $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ is an organizational network, $v \in \mathtt{V}$ is a vertex, $k > 0$ is an integer, and $P \subseteq \mathsf{VP}$ is a set of vertex properties. The property oriented clustering coefficient $pocc_{k,P}(v)$ of vertex v is:

$$pocc_{k,P}(v) = \frac{2 \cdot |\{(u,z) \in \mathtt{E} \ s.t. \ u, z \in nbr_{k,P}(v)\}|}{|nbr_{k,P}(v)| \cdot (|nbr_{k,P}(v)| - 1)}$$

where $nbr_{k,P}(v) = \{u \,|\, d(u,v) \le k \wedge (\forall p \in P)\wp(u,p) = 1\}$. As usual, $d(u,v)$ denotes the shortest path distance between u and v. In the case where $|nbr_{k,P}(v)| \in \{0,1\}$, we set $pocc_{k,P}(v) = 0$. □

[2] In classical Pagerank [1], δ is usually chosen to be around 0.85 and we stick with this number in our analysis.

[3] Later, we will run experiments in Section VII to determine what the α_i's should be for selected properties.

[4] We could have considered centrality notions but did not do so because Weighted Removal Pagerank is already a form of centrality measure. All our algorithms can, however, work with any set of vertex properties including ones such as classical centrality measures not listed here.

POCC identifies all vertices u within a distance k of v that have all the properties specified in the set P. It then computes the percentage of such vertices u that are directly connected to one another. Thus, it generalizes the classical local clustering coefficient[16].

Example 3: Let us return to the toy example shown in Figure 1 – suppose we wish to remove vertex A. The property oriented clustering coefficients of the vertices in Figure 1 are shown below when $k = 2$ and $P = \emptyset$.

v	$pocc_{2,\emptyset}(v)$	v	$pocc_{2,\emptyset}(v)$
B	1/2	E	1/3
C	2/3	F	1
D	1/2		

For instance, $nbr_{2,\emptyset}(D) = \{C, B, A, E\}$ and there are three edges amongst vertices in $nbr_{2,\emptyset}(D)$, i.e. $(C, B), (B, A), (A, E)$. It follows that $pocc_{2,\emptyset}(D) = \frac{2\cdot 3}{4\cdot(4-1)} = \frac{1}{2}$.

Similarly, we show $pocc_{k,P}(v)$ for $k = 2$ and $P = \{\texttt{bombmaker}\}$.

v	$pocc_{2,\{\texttt{bombmaker}\}}(v)$	v	$pocc_{2,\{\texttt{bombmaker}\}}(v)$
B	0	E	0
C	0	F	0
D	1		

□

We now define a *candidate set* of vertices one of which might replace a vertex r that we remove.

Definition 7 (Candidate set for the node r): Given an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$, a vertex $r \in \mathtt{V}$, a set $P \subseteq \mathsf{VP}$ of properties, a distance value k, and a real number $t > 0$, the set of candidates $cand(r, P, k, t)$ is the set of all vertices u in $\mathtt{V} - \{r\}$ s.t.

1) $wt(u) < wt(r)$, i.e. u has a lower position in the organization than r;
2) the distance $d(r, u)$ between r and u is less than or equal to k,
3) for all $p \in P$, $\wp(u, p) = 1$, i.e. u has all required properties.
4) $WRP(u) \geq t$.

Each candidate vertex in $cand(r, P, k, t)$ has an associated probability of replacing the removed vertex r.

Definition 8 (Replacement probability): Suppose r is a vertex considered for removal. The *replacement probability* of a given vertex $u \in cand(r, P, k, t)$ is given by

$$\mathbb{P}_{cand(r,P,k,t)}(u) = \frac{rv(r, u)}{\sum_{u' \in cand(r,P,k,t)} rv(r, u')}$$

□

We provide a simple example below.

Example 4: Consider the toy example shown in Figure 1 and suppose we wish to remove A. When $k = 1$, $P = \emptyset$ and we use the scoring function $rv(A, v) = \frac{1}{3}WRP(v, A) + \frac{1}{3}wt(v) + \frac{1}{3}pocc_{2,\emptyset}(v)$, the candidate set for the node A in Figure 1 is $cand(A, P, k, t) = \{B, D, E\}$, where $\mathbb{P}(B) = 0.30$, $\mathbb{P}(D) = 0.37$, and $\mathbb{P}(E) = 0.33$.

In this paper, we assume that the candidate chosen to replace vertex r is selected according to the pdf defined above applied to the set $cand(r, P, k, t)$. The user (or application) can choose P and k as they see fit. Moreover, we envisage that in an operational system, when an analyst is considering who will replace a vertex r, he will either be shown the top m candidates or all candidates whose probability exceeds a user-set threshold so he can make an informed choice.

V. Person Successor: Multi-Vertex Removal

In this section, we extend the results of the previous section to handle the removal of multiple vertices.

Definition 9 (Candidate replacement set for a set R of vertices): Suppose $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$, $P \subseteq \mathsf{VP}$ is a set of properties, k is a distance value, and $t > 0$ is a real number. Suppose $R \subseteq \mathtt{V}$ is a set of vertices to be removed. A *candidate replacement set* X_R is any set of pairs of the form (r, c_r) such that:

1) For each $r \in R$, there is a pair $(r, c_r) \in X_R$, i.e. each removed node has a replacement, and
2) $\{c_r \mid (r, c_r) \in X_R\} \subseteq (\mathtt{V} - R)$, i.e. a replacement node cannot be removed at the same time, and
3) for each pair $(r, c_r) \in X_R$, $c_r \in cand(r, P, k, t)$.

We use $cand(R, P, k, t)$ to denote the set of all candidate replacement sets. □

Given a candidate replacement set X_R for the set of vertices R, we use $vertices(X_R)$ to denote the set of vertices $\{c_r \mid (r, c_r) \in X_R\}$. [5]

Another requirement might be that the set $cand(R, P, k, t)$ be as small as possible (i.e. be minimal w.r.t. inclusion or be minimal w.r.t. the set $\{c_r \mid (r, c_r) \in cand(R, P, k, t)\}$). This will be enforced through the notion of an optimal candidate set defined shortly below - but before doing that, we present a simple example.

Example 5: Consider the example in Figure 1, and suppose the vertices to be removed are $R = \{A, D\}$. When $k = 1$ and $P = \emptyset$, $cand(A, P, k, t) = \{B, D, E\}$, while $cand(D, P, k, t) = \{C\}$. Then, $\{(A, B), (D, C)\}$ and $\{(A, E), (D, C)\}$ are possible candidate replacement sets for R, while $\{(A, D), (D, C)\}$ is not admissible since the removed node D cannot be used to replace the node A. □

Definition 10 (Probability of candidate replacement set): Suppose we are given an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$, a set $P \subseteq \mathsf{VP}$ of properties, a distance value k, and a real number $t > 0$. Suppose $R \subseteq \mathtt{V}$ is a set of vertices to be removed and X_R is a candidate replacement set. For each $r \in R$, let c_r be its replacement in X_R and let $\mathbb{P}(c_r)$ denote the probability of a replacement vertex c_r w.r.t. $cand(r, P, k, t) \setminus R$. Then the probability of X_R is given by:

$$\mathbb{P}(X_R) = \begin{cases} 0, & if\ \exists (r, c_r), (r', c_r) \in X_R : r \neq r'; \\ \Pi_{(r,c_r) \in X_R} \mathbb{P}(c_r), & otherwise. \end{cases}$$

[5]The above definition says that a candidate set is any set of replacement vertices, one for each vertex to be removed, which satisfy the obvious requirements. It makes the simplifying assumption for now that P, k are fixed to be the same for all vertices. In the real world, k might be the same for all vertices, but P might different. For instance, we might wish to replace some vertices with vertices having one set of properties, and other vertices with replacements having other properties. The framework can be easily extended to handle this but is not done here due to space constraints.

□

We assume that replacements of vertices in R are independently chosen.

Example 6: Consider the candidate replacements sets $X_R = \{(A, B), (D, C)\}$ and $X'_R = \{(A, E), (D, C)\}$ from Example 5 for the vertices $R = \{A, D\}$ in Figure 1. For the scoring function given in Example 4, $\mathbb{P}(B) = 0.45$ and $\mathbb{P}(E) = 0.55$, where $\mathbb{P}(B)$ (resp. $\mathbb{P}(E)$) denotes the probability of the replacement vertex B (resp. E) w.r.t. $cand(A, P, k, t) \setminus R$. Since $\mathbb{P}(C) = 1$ (w.r.t. $cand(B, P, k, t) \setminus R$), $\mathbb{P}(X_R) = 0.45 \cdot 1 = 0.45$ and $\mathbb{P}(X'_R) = 0.55 \cdot 1 = 0.55$. □

Multi-vertex replacement (MVR) problem. Suppose we are given an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$, a set $P \subseteq \mathsf{VP}$ of properties, a distance value k, a real number $t > 0$, and a $\delta \in [0, 1]$. The MVR problem is that of finding the set $\{X_R \mid X_R$ is a candidate replacement set and $\mathbb{P}(X) - \mathbb{P}(X_R) \leq \delta\}$ where X is the candidate replacement set s.t. $\mathbb{P}(X)$ is maximal.

Note that X is NOT an input to the above problem. The idea behind MVR is that we want to present the counter-terror analyst a set of options that he can choose from, all having close to the highest probability of replacing vertex r. What we mean by "close" is captured by a user-specified δ. When $\delta = 0$ the MVR problem returns all candidate replacement sets having highest probability. We report on experiments where we vary δ to be $2, 3, 4, 5$.

The MVR problem is closely related to a matching problem over a complete weighted bipartite graph. Let $G = (V_1 + V_2, E, c)$ be a complete weighted bipartite graph where $V_1 = R$ and $V_2 = \bigcup_{r \in R} cand(r, P, k, t) \setminus R$ are two disjoint sets of vertices, $E = V_1 \times V_2$ is the set of edges, and $c : V_1 \times V_2 \to \mathbb{R}$ is a cost function assigning a weight to each edge. In particular, $c(\langle r, s\rangle) = \mathbb{P}(s)$ if $s \in cand(r, P, k, t)$, and $c(\langle r, s\rangle) = 0$, otherwise. Then, if $\delta = 0$, the MVR problem corresponds to the problem of finding a matching $M \subseteq E$ in G whose corresponding weight product of the edges in M is maximum. Moreover, if we assume we are given as input, for each edge $\langle r, s\rangle \in E$, a weight $c'(\langle r, s\rangle) = log(c(\langle r, s\rangle))$, then the MVR problem with $\delta = 0$ can be solved by computing the maximal matching over the graph $G' = (V_1 + V_2, E, c')$, since maximizing the weight product corresponds to maximizing the sum of the logarithm of the weights. It is well known that the maximal matching over a complete weighted bipartite graph is polynomial, and can be computed by using the Hungarian algorithm [5] in time $O(n^3)$, where $n = |V_1| + |V_2|$.

The following result shows that the MVR problem can be solved in polynomial time.

Theorem 1: The Multi-vertex Replacement Problem can be solved in polynomial time in the size of the number of candidate replacement sets X_R s.t. $\mathbb{P}(X) - \mathbb{P}(X_R) \leq \delta$

Proof: [10] describes an efficient algorithm for ranking, in increasing order of cost, all solutions to the minimal matching in a complete weighted bipartite graph. Indeed, the same algorithm can be used to rank the solutions for the maximal matching. Given the i^{th}-solution, $1 \leq i \leq m - 1$ where m is the number of solutions, the computational cost of generating the next solution in the sequence is that of solving at most $(n-1)$ different maximal matching problems, one each of size $2, 3, \ldots, n$. Recall that a maximal matching problem can be solved in time $O(n^3)$. Then, the solution to the MVR problem can be found by using the algorithm given in [10] and by computing in decreasing order of cost, i.e. by starting from the most probable solution, all the candidate replacement sets, and stopping the computation when the current solution X_R is such that $\mathbb{P}(X) - \mathbb{P}(X_R) > \delta$. It follows that, the computational cost is $O(p(n-1)n^3)$, where p is the number of the number of candidate replacement sets X_R s.t. $\mathbb{P}(X) - \mathbb{P}(X_R) \leq \delta$. □

VI. Network Re-shaping Prediction

In STONE, when a set R of vertices R is removed, the network re-shapes itself according to a cascading process as follows: the set of nodes R is replaced according to a candidate replacement set $X_R \in cand(R, P, k, t)$, the set of nodes $R_1 = vertices(X_R)$ is replaced according to a candidate replacement set $X_{R_1} \in cand(R_1, P, k, t)$, the set of nodes $R_2 = vertices(X_{R_1})$ is replaced according to a $X_{R_2} \in cand(R_2, P, k, t)$, and so on, until $cand(R_m, P, k, t)$ is empty, or $m = h$, for some user-specified value h. Of course, vertices in $vertices(X_{R_i})$ do not include vertices already used to replace other vertices during the re-shaping process.

A. Possible Worlds

Since for each set of removed nodes $R, R_1, \ldots,$ we have many candidate replacement sets $X_R, X_{R_1}, \ldots$ in $cand(R, P, k, t), cand(R_1, P, k, t), \ldots$ distributed according to probability distribution functions $\mathbb{P}(X_R), \mathbb{P}(X_{R_1}), \ldots,$ there is a set of possible networks (or *possible worlds*) of $\mathcal{ON}$, each showing a possible re-shaping of $\mathcal{ON}$.

Definition 11 (Substitution diagram tree): Let $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ be an organizational network and let $R \subseteq \mathtt{V}$ be a set of vertices to be removed from $\mathcal{ON}$. A substitution diagram tree is a tree $\mathtt{T}_{\mathcal{ON}} = (\mathtt{V_T}, \mathtt{E_T}, \mathcal{P}_{\mathtt{E_T}})$ where:

1) the set of nodes $\mathtt{V_T}$ and edges $\mathtt{E_T}$ is defined recursively as follows: (i) the replacement set $X_\emptyset = \{(\_, r) \mid r \in R\}$ is the root node; (ii) for each node $X_V \in \mathtt{V_T}$, the set of its children is given by $cand(vertices(X_V), P, k, t)$; 2) $\mathcal{P}_{\mathtt{E_T}} : \mathtt{E_T} \to [0, 1]$ is a function assigning probabilities to edges s.t.

$$\mathcal{P}_{\mathtt{E_T}}(\langle X_V, X_{V'}\rangle) = \frac{p(X'_V)}{\sum_{X_{V''} \in cand(vertices(X_V), P, k, t)} p(X_{V''})}$$

where $\langle X_V, X_{V'}\rangle$ is an edge in $\mathtt{E_T}$, U is the union of all $vertices(X)$ for all X that are in the path from $X_\emptyset$ to X_V in $\mathtt{T}_{\mathcal{ON}}$ ($X_\emptyset$, X_V included), and

$$p(X_{V'}) = \begin{cases} 0, & \text{if } vertices(X_{V'}) \cap U \neq \emptyset; \\ \mathbb{P}(X_{V'}), & \text{otherwise.} \end{cases}$$

□

We now define a possible world.

Definition 12 (Possible world): Let $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ be an organizational network and let $R \subseteq \mathtt{V}$ be a set of vertices to be removed. A *possible world* (or possible network re-shaping) N_π is a network obtained from $\mathcal{ON}$ by (1) taking a path $\pi = (X_0 = X_\emptyset, X_1 = X_R, \ldots, X_{m+1} = X_{R_m})$ in $\mathtt{T}_{\mathcal{ON}} = (\mathtt{V_T}, \mathtt{E_T}, \mathcal{P}_{\mathtt{E_T}})$ from the root to a leaf, and (2) by replacing $vertices(X_i)$ with $vertices(X_{i+1})$, for $i = 0, \ldots, m$. □

Algorithm 1 STONE-Predict Network prediction algorithm

Input: Organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$, set of nodes $R \subseteq \mathtt{V}$ to be removed from $\mathcal{ON}$, and the maximum candidate distance k.
Output: Set $\mathcal{N}$ of possible worlds for $\mathcal{ON}$ after the removal of R.
function POSSIBLEWORLD($\mathcal{ON}$, R, k)
  $\mathcal{N} = \emptyset$
  VISIT(R,$\langle \mathcal{ON}, 1\rangle$,$R$,$\mathcal{N}$)
  return $\mathcal{N}$
end function

Input: node set to be removed R, possible world $\langle N, p\rangle$, set of already used nodes U, set $\mathcal{N}$ of possible worlds
function VISIT(R, $\langle N, p\rangle$, U, $\mathcal{N}$)
  If($cand(R, P, k, t) = \emptyset \wedge condition$[1])
    $N' = $ REMOVE(R,N)
    $\mathcal{N} = \mathcal{N} \cup \langle N', p\rangle$
  Else
    $allLowProb = true$
    For each $X_R \in cand(R, P, k, t)$
      $p' = $ PROBABILITY(R, X_R, U)
      If($p' \geq T_{Pr}$)
        $allLowProb = false$
        $N' = $ REPLACE(R,X_R,N)
        $p' = p \cdot p'$
        $U' = U \cup R$
        VISIT(R, $\langle N', p'\rangle$, U', $\mathcal{N}$)
    End For
    If($allLowProb$)
      $N' = $ REMOVE(R,N)
      $\mathcal{N} = \mathcal{N} \cup \langle N', p\rangle$
  End If
end function

function PROBABILITY(R, X_R, U)
  Return the probability $\mathcal{P}_{\mathtt{E}'}(X_\emptyset, X_R\rangle)$ computed as in Definition 11
end function

function REMOVE(D, $(\mathtt{V}, \mathtt{E}, wt, \wp)$)
  $\mathtt{V}' = \mathtt{V} \setminus D$
  $\mathtt{E}' = \mathtt{E} \setminus \bigcup_{d \in D} \{(x, y) \mid (x, y) \in \mathtt{E} \wedge x = d \, or \, y = d\}$
  return $(\mathtt{V}', \mathtt{E}', wt, \wp)$
end function

function REPLACE($\{r_1, \ldots, r_n\}, \{(r_1, c_1), \ldots, (r_n, c_n)\}, (\mathtt{V}, \mathtt{E}, wt, \wp)$)
  For each $i \in \{1, \ldots, n\}$
    $\mathtt{E}' = \mathtt{E} - \{(x, y) \mid (x, y) \in \mathtt{E} \wedge x = r_i \, or \, y = r_i\}$
    $\mathtt{E}' = \mathtt{E}' \cup \{(c_i, x) \mid (r_i, x) \in \mathtt{E} \wedge x \neq c_i\} \cup \{(y, c_i) \mid (y, r_i) \in \mathtt{E} \wedge y \neq c_i\}$
    $\mathtt{E}' = \mathtt{E}' \cup \{(r_i, x) \mid (c_i, x) \in \mathtt{E} \wedge x \neq r_i\} \cup \{(y, r_i) \mid (y, c_i) \in \mathtt{E} \wedge y \neq r_i\}$
    $wt'(v) = wt(v), \wp'(v) = \wp(v) \forall v \in \mathtt{V} \setminus \{r_1, \ldots, r_n\}$
    $wt'(r_i) = wt(c_i)$, $\wp'(r_i) = \wp(c_i)$, $1 \leq i \leq n$
  End For
  return $(\mathtt{V}, \mathtt{E}', wt', \wp')$
end function

[1] $condition$ expresses a condition like $prob(N) \geq T$, or $L(N) \geq T'$, etc.

Given an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ and a set of vertices $R \subseteq \mathtt{V}$ to be removed from $\mathcal{ON}$, we denote by $\mathcal{N}(\mathcal{ON}, R)$ the set of all possible worlds for $\mathcal{ON}$ w.r.t. R. Given a possible world $N_\pi \in \mathcal{N}(\mathcal{ON}, R)$, the probability of N_π is given by

$$p(N_\pi) = \prod_{i=0}^{m} \mathcal{P}_{\mathtt{E}_\mathtt{T}}(\langle X_i, X_{i+1}\rangle)$$

where $\langle X_i, X_{i+1}\rangle$, $0 \leq i \leq m$, are the edges in the path π.

B. Network Lethality

Each possible world above has an associated lethality. Lethality can be defined in many ways and we provide a few here. In our real-world network data for Al-Qaeda and Lashkar-e-Taiba, we have a `hostility` (towards the West) and `support` (for violent acts) property that we use here.

Let α, β be two real values in the interval $[1, MAX]$ for some number MAX (in our data, $MAX = 5$). Given an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ and vertex $u \in \mathtt{V}$, $u \in \mathtt{V}$ is *violence-prone* iff $hostility(u) \geq \alpha \wedge support(u) \geq \beta$, otherwise it is *violence-averse*. We denote by $VP(\mathcal{ON}) \subseteq \mathtt{V}$ the set of all network nodes that are violence-prone, and by $VA(\mathcal{ON}) = \mathtt{V} \setminus VP(\mathcal{ON})$ the set of nodes that are violence-averse. Three measures of network lethality are given below.

$$L_1(\mathcal{ON}) = \textstyle\sum_{v \in VP(\mathcal{ON})} wt(v) - \sum_{v' \in VA(\mathcal{ON})} wt(v')$$
$$L_2(\mathcal{ON}) = \textstyle\sum_{v \in VP(\mathcal{ON})} deg(v) \cdot wt(v) - \sum_{v' \in VA(\mathcal{ON})} deg(v') \cdot wt(v')$$
$$L_3(\mathcal{ON}) = \textstyle\sum_{v \in VP(\mathcal{ON})} WRP(v, \emptyset) \cdot wt(v) - \sum_{v' \in VA(\mathcal{ON})} WRP(v, \emptyset) \cdot wt(v')$$

The first measure is simple: it says network lethality is simply the sum of the weights of violence-prone vertices minus the sum of the weights of the violence-averse vertices. The second and third measures say that these sums should be weighted by a measure of the influence of the vertex - in the second lethality measure, that measure of influence is the degree of the vertex, while in the third, the measure of influence is the Weighted Removal Pagerank defined earlier.

We now define the expected lethality value of the resulting network when a set R of nodes is removed.

Definition 13 (Expected lethality): Let $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ be an organizational network, and $R \subseteq \mathtt{V}$ be a set of nodes to be removed from $\mathcal{ON}$. Then, the lethality expected value $L_{\mathrm{EV}}(\mathcal{ON}, R)$ after the removal of the nodes in R from $\mathcal{ON}$ is

$$L_{\mathrm{EV}}(\mathcal{ON}, R) = \textstyle\sum_{N \in \mathcal{N}(\mathcal{ON}, R)} p(N) \cdot L(N)$$

where $p(N)$ is the probability of the world N, and $L(N)$ its lethality. $\square$

Network Reshaping Problem. Given an organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$ and a set EX of vertices that cannot be removed, find a set of vertices $R \subseteq \mathtt{V}$ to remove from $\mathcal{ON}$ s.t. $|R| \leq k$ and the expected lethality $L_{\mathrm{EV}}(\mathcal{ON}, R)$ is minimized.

We now present in Algorithm 2 a heuristic (greedy) algorithm to solve the Network Reshaping Problem. Algorithm 2 starts with an empty set R of nodes to be removed, and, at each iteration, proceeds by selecting the node r such that, by adding it to the nodes to be removed, the resulting lethality expected value is minimum. If by adding r to R the resulting lethality expected value decreases w.r.t. the lethality expected value obtained by removing only R, then R is updated to $R = R \cup \{r\}$ and if $|R| \leq k$ the iterations continue, otherwise the algorithm terminates.

VII. Implementation and Experiments

We implemented all algorithms described here in 1000 lines of Java code and ran them on a AMD Opteron Quad-Core 2354 @ 2.2 GHz, 8GB RAM Linux machine (i) Our Toy dataset containing 50 toy graphs each with a vertex r

Algorithm 2 STONE-Reshape network reshaping algorithm

```
Input: Organizational network $\mathcal{ON} = (\mathtt{V}, \mathtt{E}, wt, \wp)$, maximum set size $k$, set $EX$ of non-removable vertices.
Output: Set $R$ of nodes to remove.
function RESHAPE($\mathcal{ON}$, $k$, $EX$)
    $R = \emptyset$
    $\ell = L(\mathcal{ON})$
    $continue = true$
    do
        $r = \arg\min_{v \in \mathtt{V} \setminus (R \cup EX)} L_{\mathrm{EV}}(\mathcal{ON}, R \cup \{v\})$
        $\ell_r = L_{\mathrm{EV}}(\mathcal{ON}, R \cup \{r\})$
        If($\ell_r \leq \ell$)
            $R = R \cup \{r\}$
            $\ell = \ell_r$
        Else $continue = false$
        EndIf
    while($continue \wedge |R| \leq k$)
    return $R$
end function
```

Setting	Dataset	$\delta = 2\%$	$\delta = 3\%$	$\delta = 4\%$	$\delta = 5\%$
1	Toy	0.905(2)	0.925(3)	0.95(3)	0.99(5)
	AQ	0.7(6)	0.8(7)	0.8(9)	0.8(10)
	LeT	0.667(4)	0.667(6)	0.833(7)	0.833(7)
2	Toy	0.56(2)	0.56(2)	0.605(2)	0.635(2)
	AQ	0.8(9)	0.8(12)	0.8(13)	0.8(14)
	LeT	0.5(5)	0.667(7)	0.833(7)	0.833(8)
3	Toy	0.53(2)	0.80(3)	0.865(4)	0.935(5)
	AQ	0.7(6)	0.8(8)	0.8(9)	0.8(12)
	LeT	0.833(5)	0.833(7)	0.833(7)	0.833(8)
4	Toy	0.835(3)	0.93(4)	0.97(4)	0.985(5)
	AQ	0.7(6)	0.8(8)	0.8(9)	0.8(12)
	LeT	0.833(5)	0.833(7)	0.833(7)	0.833(8)
5	Toy	0.765(3)	0.865(4)	0.92(5)	0.94(5)
	AQ	0.8(7)	0.8(8)	0.8(11)	0.8(13)
	LeT	0.833(6)	0.833(7)	0.833(7)	0.833(8)
6	Toy	0.16(2)	0.26(3)	0.32(3)	0.37(3)
	AQ	0.2(3)	0.3(5)	0.8(9)	0.8(12)
	LeT	0.667(4)	0.833(6)	0.833(7)	0.833(8)

Fig. 4. Accuracy results for Experiments 1

being removed. Each vertex's rank within the organization was scored from 1 to 10. The successor of r was clearly identified by 2 counter-terrorism experts (one an author of two books in CT, another a retired US general) and 3 students. (ii) The Al-Qaeda dataset (39 nodes, 63 edges) was taken from Sageman [14]. (iii) Our Lashkar-e-Taiba (LeT) dataset (166 nodes, 240 edges) was constructed during the writing of a book by two of the authors on LeT. Both the Al-Qaeda and LeT datasets contained real information on people who were removed from the network and who replaced them. Thus, the last two data sets represent real ground truth. As the STONE algorithm does not require training, all the data was used directly for testing.

Experiment 1. We ran experiments to determine which settings yielded the highest prediction accuracy for the PSP problem. We allowed δ to be $2, 3, 4, 5\%$. For space reasons, we just show six of the settings we used below. $\alpha_{WRP}, \alpha_{pocc}$ denote the weights of WRP and POCC, α_{wt} denotes the rank of a vertex, $\alpha_{hostility}$ denotes the hostility of a vertex to the West, while $\alpha_{support}$ denotes the vertex's support for terrorism.

Setting	α_{WRP}	α_{pocc}	α_{wt}	$\alpha_{hostility}$	$\alpha_{support}$
1	0.1	0.1	0.6	0.1	0.1
2	0.8	0	0	0.1	0.1
3	0.267	0.266	0.267	0.1	0.1
4	0.2	0.2	0.4	0.1	0.1
5	0.4	0.2	0.2	0.1	0.1
6	0.4	0.4	0	0.1	0.1

The accuracy results are summarized in Figure 4. The numbers in parentheses denote the cardinality of the answer returned. An answer was deemed correct if one of the returned solutions was in fact the true successor of the vertex being removed. We see that for $\delta = 2\%$, settings 4 and 5 give the best results (highest accuracy and small numbers of returned candidates). When $\delta = 3\%$, setting 4 gives the best results though setting 1 also has some advantages (fewer number of answers given with an almost equivalent accuracy). *In short, the results show that we can use setting 4* with either $\delta = 2\%$ or 3%. The results on real-world AQ and LeT data shows that the rank of a vertex is the most significant predictor of the vertex being a successor to a removed vertex and that the Weighted Removal Pagerank and Property Oriented Clustering Coefficient also play a significant role.

Experiment 2. As measuring the accuracy of the STONE-Reshape algorithm by removing terrorists from networks is infeasible and no ground truth exists from the past to test it (as terrorists removed in the past may not have been the optimal ones to remove), we tested the accuracy of STONE-Reshape against opinion of counter-terror experts who rated the results on a 1-5 scale (with 5 being the best). We tested all three lethality functions and all six settings above against the Al-Qaeda and Lashkar-e-Taiba datasets for a total of 18 combinations. Space reasons prevent us from presenting the full results, but the table below shows the score of each of the lethality functions L_1, L_2, L_3 averaged across the six settings above.

Lethality function $\rightarrow$	1	2	3
AQ	3.46	-75.83	-1.27
LeT	-470	-1435	-3.57

We see that lethality function L_3 got high degrees of satisfaction from the counter-terror experts with close to a 4 out of 5 for reshaping both Al-Qaeda and LeT. Moreover, for just lethality function L_3, we show the results on each of the six settings.

Setting	Al-Qaeda	Lashkar-e-Taiba
1	4	3.5
2	4	4
3	4	4
4	4	4
5	4	4
6	4	4

Thus, we see that lethality function L_3 works well with all of the six settings with a mild dip for the first setting. In particular, this validates Experiment 1's result that setting 4 is a good one.

VIII. Related Work and Conclusions

Identifying key actors in networks has been studied extensively using traditional social network measures such as degree, between-ness, eigenvector, and closeness centrality measures [16]. Memon [8] studies traditional centrality measures to identify key actors in terror networks. Memon and Larsen [9] use classical centrality measures and define efficiency of a network using prior work by Latora and Marchiori [6]. They propose a position role index (PRI) to distinguish between gatekeepers and followers in such networks and define "dependence centrality" (DC). They show the values of DC on Krebs' [4] $9/11$ network. Carley [3] proposes the CONSTRUCT-O model in which we can measure destabilization of a network via three measures: ability to diffuse information through the network, ability to reach consensus, and reducing the ability of the network to perform tasks.

Ozgul et al. [12] have studied the problem of detecting terror cells in terror networks and proposed a variety of algorithms such as the GDM and OGDM methods. Similarly, Lindelauf [7] et al. have studied the structure of terrorist networks and how they need to maintain sufficient connectivity in order to connectivity as well as maintain sufficient disconnectivity in order to stay hidden - they model this tension between communication and covertness via a game-theoretic model. This same intuition led to the concept of covertness centrality [11] in social networks where a statistical (rather than game-theoretic) method is used to predict covert vertices in a network.

Callahan et. al. [2] reshape networks by identifying a set of nodes to remove that jointly maximize the network wide degree centrality which extends degree centrality of nodes to a network as a whole. Network wide degree centrality could possibly be one of the many notions of lethality defined in this paper. The authors also do not take properties of nodes into account. Though they do run tests on real data, they do not compare against real ground truth as we do. Petersen et al. [13] develop methods to remove nodes in criminal networks by taking into accounts, the new links generated by the removal (similar to Definition 3 in this paper).

In contrast, STONE develops a mathematical model of who replaces a "removed" individual in a terrorist network taking into account, not only network structures, but also the properties of the vertices in the network (such as operational roles) as well as the need for the terrorist network to achieve operational efficiency. We show that our STONE-Predict algorithm has over 80% accuracy. We also identify a pdf over the space of possible networks that might result after a set of vertices is "removed" and use this to determine which set of vertices in a network to remove. We test our results on both synthetic networks and 2 real-world networks of AQ derived from Sageman [14] as well as Lashkar-e-Taiba [15]. Our results are validated either by what happened in real life (person successor problem) or by experts.

IX. Conclusion

In this paper, we define STONE, a framework to identify a set of k nodes whose removal will optimally destabilize a terrorist network. In order to achieve this, we note that unlike past work, terror networks are not just a bunch of nodes and edges. Nodes have semantic properties like a rank, a role or roles in the organization, levels of capability and hostility, and so forth, which must be taken into account when deciding what nodes to remove from a network. In order to achieve this, we start by defining organizational networks that have such semantic properties (and in fact, our Al-Qaeda and Lashkar-e-Taiba data includes these semantic properties). STONE solves three problems in order to destabilize an organization: first, we predict who will replace a given vertex that is removed. We show using real data on replacements that have occurred in AQ and LeT, that in over 80% of the cases, the vertex that replaces a removed vertex has a probability within 2% of the most probable replacement – thus, operationally, STONE can present analysts with the most likely replacement for a possibly removed vertex along with those whose probability of being a replacement is within around 2-3% of the most likely replacement. Second, STONE defines the set of possible networks that results when a set of nodes is removed and associates a probability distribution over this set of possible networks. Using multiple possible definitions of the lethality of a terrorist network based on both network and semantic properties, STONE is able to identify a set of k individuals whose removal would minimize the expected lethality of the network. We tested the results on who to remove by asking counter-terrorism experts to evaluate the results associated with Lashkar-e-Taiba and Al-Qaeda and they uniformly gave STONE a high score.

Acknowledgement. Parts of this work may have been funded by US Army Research Office grant W911NF0910206.

References

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[2] D. Callahan, P. Shakarian, J. Nielsen, and A. N. Johnson. Shaping operations to attack robust terror networks. *arXiv preprint arXiv:1211.0709*, 2012.

[3] K. M. Carley, J.-S. Lee, and D. Krackhardt. Destabilizing networks. *Connections*, 24(3):31–34, 2001.

[4] V. E. Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43–52, 2002.

[5] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[6] V. Latora and M. Marchiori. How the science of complex networks can help developing strategies against terrorism. *Chaos, solitons & fractals*, 20(1):69–75, 2004.

[7] R. Lindelauf, P. Borm, and H. Hamers. The influence of secrecy on the communication structure of covert networks. *Social Networks*, 31(2):126–137, 2009.

[8] B. R. Memon. Identifying important nodes in weighted covert networks using generalized centrality measures. In *EISIC*, pages 131–140, 2012.

[9] N. Memon and H. L. Larsen. Practical algorithms for destabilizing terrorist networks. In *ISI*, pages 389–400, 2006.

[10] K. G. Murty. An Algorithm for Ranking all the Assignments in Order of Increasing Cost. *Operations Research*, 16(3):682–687, 1968.

[11] M. Ovelgönne, C. Kang, A. Sawant, and V. S. Subrahmanian. Covertness centrality in networks. In *ASONAM*, pages 863–870, 2012.

[12] F. Ozgul, J. Bondy, and H. Aksoy. Mining for offender group detection and story of a police operation. In *Proceedings of the sixth Australasian conference on Data mining and analytics - Volume 70*, AusDM '07, pages 189–193, 2007.

[13] R. R. Petersen, C. J. Rhodes, and U. K. Wiil. Node removal in criminal networks. In *Intelligence and Security Informatics Conference (EISIC), 2011 European*, pages 360–365. IEEE, 2011.

[14] M. Sageman. *Understanding Terror Networks*. University of Pennsylvania Press, 2004.

[15] V. Subrahmanian, A. Mannes, A. Sliva, J. Shakarian, and J. Dickerson. *Computational Analysis of Terrorist Groups: Lashkar-e-Taiba*. SpringerLink : Bücher. Springer London, Limited, 2012.

[16] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 4 1998.