DIMACS Technical Report 2005-26
July 2005

Critical Groups in Dynamic Social Networks

by

Tanya Y. Berger-Wolf[1]
Center for Discrete Mathematics and Theoretical Computer Science (DIMACS)
CoRE Bldg, Rutgers University
96 Frelinghuysen Rd., Piscataway, NJ 08854
and
Department of Computer Science, University of Illinois at Chicago,
851 S. Morgan, Chicago, IL 60607
`tanyabw@cs.uic.edu`

Jared Saia[2]
Department of Computer Science, University of New Mexico,
Albuquerque, NM 87131
`saia@cs.unm.edu`

[1]DIMACS Postdoctoral Fellow, suported by the National Science Foundation postdoctoral fellowship grant EIA 02-05116
[2]Supported by the National Science Foundation grant CCR-0313160 and by the Sandia University Research Program Grant No. 191445

ABSTRACT

We address the issue of fragility of a network of interactions, such as a social network, in a dynamic setting. We propose a new mathematical and computational framework that allows analysis of dynamic social networks addressing the time component explicitly. In this framework, we pose the question of finding a critical set of groups at various times whose lack of existence would leave no persistent social structure in the network. We formulate this question in terms of a graph optimization problem, prove that it is NP-hard, and provide a polynomial time algorithm for one important special case.

1 Introduction

Finding patterns of social interaction within a population has wide-ranging applications including: disease modeling [10, 14, 20], cultural and information transmission [2, 4, 5, 7, 15, 30, 31, 34], intelligence and surveillance [1, 2, 19, 25, 24], business management [3, 6, 28, 29], conservation biology and behavioral ecology [8, 9, 22]. Until recently, questions regarding social structures and social interactions in human or animal populations were answered using statistical tools, which summarized aggregate information over the population. Recently, scientists have started to model social interaction with graphs (networks) [12, 16, 17, 18, 26, 27, 23, 32, 33].

Typically, each individual is represented by a node in the graph, and there is an edge between two nodes if a social interaction occurred at any point in time between the two individuals represented by these nodes. Depending on the source of data, a social interaction could be a verbal or written communication, scientific collaboration, sexual contact, physical proximity, affiliative or antagonistic behavior, or potential disease transmission. Sometimes, the edges in the graph are weighted with the frequency or the probability of interaction.

The technique of modeling social interactions using this type of graph has been very successful. However, a major drawback of this model is that it is essentially static in that all information about the time that social interactions take place is discarded or long time series are averaged to discern the overall or long-term strength of connections. The static nature of the graph gives rise to two major problems. First, it can give us inaccurate or inexact information about patterns in the data. For example, as Figure 1 illustrates, in some cases, very different dynamic data can give rise to the same static graph. The decisions made based solely on the static data may be flawed. For example, in the epidemiological setting Figure 1 represents potential disease transmission edges. Suppose we are interested in vaccinating a limited number of people (1 person for the purpose of this example) so as to minimize the total number of infected individuals at the end, assuming any contact will facilitate the transmission and any individual may be infected initially. In the static graph there is no difference between the three individuals and it seems no matter which one individual is vaccinated there will be two people infected at the end. However, the dynamic situation may be different and other factors may come into play. In the first case, assuming the initial infection starts at step 0 and lasts for no longer than 2 time steps, only b would need to be vaccinated, and no matter whether a or c is infected initially, only one individual will stay infected.

A second, more serious, problem with the static graph representation is that it prevents us from even asking certain fundamental questions about either the causes or consequences of social patterns. How quickly can a disease spread through the population? Which individuals would we inoculate to slow down its spread? Are there specific gatherings (e.g. on a college campus) that are responsible for most of the disease spread? There are many more questions that are impossible to pose in the static social networks model. To be able to answer these questions, we need to have information on when social interactions occurred.

To address the dynamic aspect of social interactions we use the graph (network) population model to capture the time snapshot information. We then build a time series graph

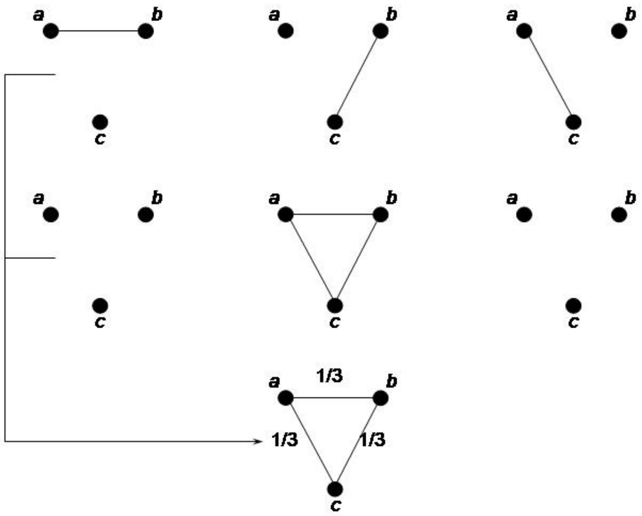

Figure 1. The top two rows represent social interactions over three time steps. The bottom row represents the static graph obtained. The social interactions specified in the first row of graphs are very different from the social interactions specified in the second row of graphs. However both sets of data give rise to the same static graph.

that connects the snapshots. The connections within a time snapshot are the individual associations that exist at that moment. We impose a metric on sets of individuals within different time steps that allows us to track changes in social structures over time. In this model, many questions about the social population dynamics become the classical questions of graph connectivity. Some of these are easily answered, yet many pose challenging algorithmic problems. One specific and important such question is finding critical sets of interactions at critical times whose absence would leave no discernible social structure in the population. For example, on a college campus or in a city there may be a set of critical times and locations where a mass vaccination effort would prevent an epidemic. Preventing critical groups of individuals from passing information between them could prevent a planning effort involving a large network of which they are a part.

The rest of the paper is organized as follows: in Section 2 we discuss the type of input our approach assumes and formally state the dynamic social network analysis framework. In Section 3 we present the formal Critical Groups Set problem, show that it is NP-hard and give a polynomial time algorithm for an important special case. Section 4 lists some open problems with the related population questions and comments on some possible solution approaches.

2 The Formal Dynamic Social Networks Analysis Framework

Here we present our mathematical and computational framework, the context, the assumptions and the necessary definitions. We start by discussing the input data.

2.1 The Input Data

To be able to answer questions about social structure in a dynamic setting, we assume that the input is as follows. A population of individuals is monitored in some way. There are three types of input data:

1. Information about each individual such as id number and GPS or other type of location or action identification. Pairwise connection data has to be inferred (from the geometric proximity data of GPS or from a common part of the action - in surveillance data).

2. Pairwise connection data (from cellphone communication, emails, sexual contact information, co-publication, etc.). Group information has to be inferred.

3. Direct group information, such as photographs, satellite data, class registration roster, etc. These may provide either an instant complete snapshot of the entire population (satellite data) or a time window snapshot of some parts of the population.

For the purpose of this paper, we assume that the input is in the form of the partition of the individuals into groups at every time step. This may be a result of the direct observation data or the output of processing of the pairwise connection data.

2.2 Definitions

We now state the formal input representation. Given a set of all the individuals in the population $X = \{x_1, ..., x_n\}$, we define a *group* to be a subset of individuals $g \subseteq X$. We assume that the input is a set of partitions, $P_1, P_2, ..., P_T$, and that each partition, P_i, is a disjoint set of *groups*:

$$\forall i,\ \ 1 \leq i \leq t\ \ P_i = \{g_{i1}, ..., g_{ii_m}\}$$

$$\cup_{j=1}^{i_m} g_{ij} = X \text{ and } \forall p \neq q,\ \ 1 \leq p, q \leq i_m,\ \ g_{ip} \cap g_{iq} = \emptyset$$

We denote by $P(g)$ the index of the partition to which g belongs. That is, if $g \in P_i$ then $P(g) = i$.

Given two groups, g and h, a set similarity measure $sim(,)$, and a *turnover threshold* β, we consider the two groups *similar* if

$$sim(g, h) \geq \beta.$$

Our definition here is independent of the set similarity measure. There may be many possible such measures, from the standard Jaccard similarity measure [13] (the size of the intersection over the union) to application-specific measures, such as genomic distance, that may be NP-hard to compute. For the purposes of this paper, we also assume that for any two groups we are given their set similarity measure, either by easy (polynomial) computation or from an outside source.

We now define the main concept of our framework - a metagroup.

Definition 1. *Given an input in the form of partitions of a set of individuals X, $P_1, ..., P_T$, a set similarity measure $sim(,)$, a turnover threshold β and a function $\alpha(T)$, a* metagroup *MG is a sequence of groups $MG = \langle g_1, ..., g_l \rangle$, $\alpha(T) \leq l \leq T$ such that*

1. *no two groups in MG are in the same partition and the groups are ordered by the partition time steps:*
$$\forall i, j,\ \ 1 \leq i < j \leq l,\ \ P(g_i) < P(g_j),$$
2. *the consecutive groups in MG are similar:*
$$\forall i,\ \ 1 \leq i < l,\ \ sim(g_i, g_{i+1}) \geq \beta.$$

We call the parameter α the persistence *of a metagroup.*

We note that the intersection between g_1 and g_l may be null by this definition; our only constraint is that the groups change gradually. A metagroup is a timeline evolution history of a group of individuals. Just like any real group, it may change its members over time (unless β is too high to allow it), yet locally remains essentially the same group. Our definition of a metagroup is flexible enough to accommodate the range of social structures, from the "never changing, always together" to the "meeting every now and then of more or less the same individuals".

The precise values of α (persistence) and β (turnover), the meaning of a "group" and the similarity measure are application specific. Our goal is to create a framework that is independent of these definitions and is capable of providing meaningful answers for a wide range of applications.

We use a weighted multipartite directed graph for the conceptual representation: $G = (V_1, ..., V_T, E)$ where V_i is the set of groups in partition P_i and $(h, g) \in E$ if $P(h) < P(g)$ and h and g are similar. Note that this is a directed acyclic graph (DAG) since all the edges are directed from an earlier time step to a later one. The weight $w(h, g) = sim(h, g)$. A metagroup in this graph is a path of length at least $\alpha(T)$. We shall call this graph a *metagroup β-graph.*

Note that the number of metagroups may easily be exponential in the number of time steps T even if the total number of groups is small and $\alpha(T) = T$. Figure 2 shows an example of such a graph. Thus, it may be impractical to explicitly list all existing metagroups. However, it is also unnecessary to do so to answer many important and interesting questions, such as the most persistent or the largest metagroup, the total number of metagroups and the connectivity of a given subset of groups. We leave the long list of the algorithms to answer specific queries about the metagroups for another paper. We now focus on the question of finding critical groups.

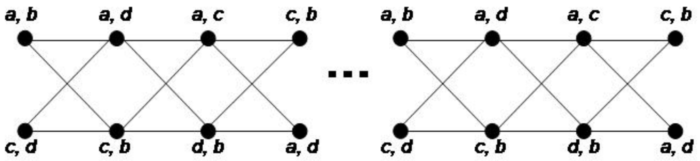

Figure 2. Here the turnover threshold β is such that groups sharing half of their members are considered similar. Each path from a group in P_1 to a group in P_T is a metagroup. Any path has a choice of two vertices in any time step, hence there are 2^T possible such paths while there are only $2T$ groups.

From now on we assume the definition of a metagroup that satisfies a priori given thresholds of α and β. We assume the input is in the form of a metagroup β-graph described above with the present edges of weight no less than β.

3 Critical Group Set

We now approach the question of fragility of a social network which can be formulated in many ways. One way to ask it is to consider the smallest set of groups whose absence would leave no recognizable overall social structures. Formally, CRITICAL GROUPS SET: Find the smallest set of groups whose removal leaves no metagroups (with respect to given β and α). This question is particularly important in an epidemiological context where it is safe to have prolonged interactions for periods no longer than the non-contagious incubation period of a disease, yet it is important to quarantine people otherwise. In this context some groups may be considered more important (or easier) to dissipate than others and this would lead to a vertex-weighted version of the problem. More generally, MIN K-PATH VERTEX SHATTERING SET: for an arbitrary graph $G = (V, E)$ find the smallest (weight) subset of vertices $U \subseteq V$ such that the subgraph induced by $V - U$ has no paths longer than $k - 1$. We show that the complexity of this problem very much depends on the specific value of k.

First, we show that the general problem is NP-complete by demonstrating that the unweighted MIN 2-PATH VERTEX SHATTERING SET is equivalent to MAX INDEPENDENT SET, which is a well studied NP-complete problem.

Theorem 1. MIN 2-PATH VERTEX SHATTERING SET *is polynomially equivalent to* MAX

Independent Set.

Proof. The Min 2-Path Vertex Shattering Set problem is to remove the smallest subset of vertices so that the remaining induced subgraph has no paths, i.e. the remaining subgraph is an independent set. Since the set of removed vertices and the independent set do not intersect, finding the largest independent set is equivalent to finding the smallest complimentary set. ∎

Corollary 1. Min k-Path Vertex Shattering Set *on an arbitrary graph G is NP-complete.*

The special structure of a metagroup β-graph does not improve the complexity of the Min 2-path Vertex Shattering Set problem.

Theorem 2. Min 2-path Vertex Shattering Set *on a metagroup β-graph is NP-complete.*

Proof. Let $G = (V, E)$ be an arbitrary instance of Max Independent Set. Starting with any vertex v, we use the orientation induced by the breadth first search (BFS) from v to obtain a directed acyclic graph on G. To ensure that this graph is a metagroups β-graph we need to assign individuals to every vertex consistent with the edges. We do this in the following way: starting with the minimal (no incoming edges) vertex v, we assign it a set x and proceed in the BFS order. For every vertex u with the currently representative individual x_p, we assign to its q children the individuals $x_{p1}, x_{p2}, ..., x_{pq}$, respectively, and replace x_p by the set $\{x_{p1}, x_{p2}, ..., x_{pq}\}$ at every vertex it appears, all the way to the root. If one of the children had another individual y_i already assigned to it, we add the additional assignment induced by u. Figure 3 shows an example of such an assignment. For each vertex u we calculate the longest path distance from v in the induced DAG, $d(u)$. A vertex u now corresponds to a group in a time step $d(u)$. It is not difficult to see that each individual appears at most once in any vertex in a given time step. It is now left to ensure that each individual appears exactly once in every time step. To do so, for every edge (u_1, u_2) such that $d(u_2) - d(u_1) > 1$ there exists an alternate path from u_1 to u_2 with every edge shorter than $d(u_2) - d(u_1)$. We add the individuals in u_2 to every vertex on that path. For example, in Figure 3 the individual x_1 will be added to the individuals x_{21}, x_{22} in time 1. Again, it is not hard to see that after this procedure for every edge longer than 1 every individual appears exactly once in every time step. Therefore, the set of vertices at every time step represents a partition of the individuals in the population. We use the similarity measure of the size of the intersection of the two sets, $sim(h, g) = |h \cap g|$ and set $\beta = .5$. With these parameters, the resulting graph is a metagroup β-graph. If we can solve the Min 2-path Vertex Shattering Set on a metagroup β-graph, we can use it to find a Max Independent Set in an arbitrary graph. Therefore, Min 2-path Vertex Shattering Set in a metagroup β-graph is NP-hard. Given a solution to the Min 2-path Vertex Shattering Set it is easy to verify its correctness, therefore the problem is NP-complete. ∎

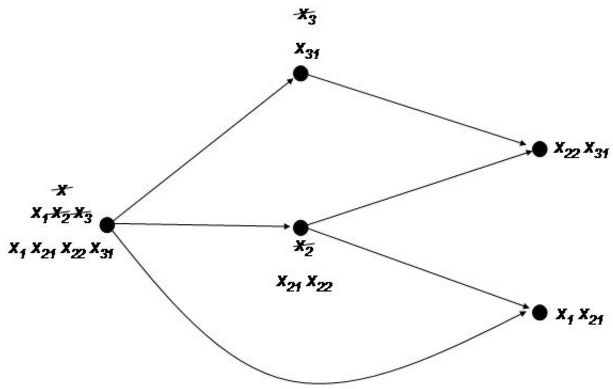

Figure 3. An example of an assignment of individuals to vertices in an arbitrary DAG.

The general problem of finding the minimum number of vertices to be deleted to obtain a subgraph with property P is well studied [21]. For any non-trivial hereditary property P the problem is APX-hard [21]. A graph property is hereditary if it is closed under taking induced subgraphs. In our case, the property $P =$ THE GRAPH HAS NO PATHS OF LENGTH k OR GREATER is certainly hereditary. However, it does not naturally fall into any of the known constant factor approximation requirements.

While the MIN k-PATH VERTEX SHATTERING SET problem is NP-hard for an arbitrary k, we present a polynomial time algorithms for $k = T$ in a DAG, where T is the length of the longest path in G. Let $P(v, l)$ be the number of paths of length exactly l from a minimal vertex to $v \in V$. The algorithm uses the following function K-GRAPH:

Alg K-GRAPH(k)
In the order of a BFS from the minimal vertices:

$$\forall v, l,\ \ 0 \leq l \leq T\ \ P(v,l) = \begin{cases} 1 & \text{if } v \text{ is minimal and } l = 0 \\ 0 & \text{if } v \text{ is minimal and } l > 0 \\ \displaystyle\sum_{(u,v)\in E} P(u, l-1) & \text{otherwise} \end{cases}$$

FORāll maximal v and $l \geq k$ s.t. $P(v,l) > 0$ **DO**
Trace back the paths of length l that terminate at v
RETURN the resulting traced graph G_k

Lemma 1. *Algorithm k-GRAPH(k) returns a subgraph G_k of G that contains every edge in G which is in some path of length at least k from a minimal to a maximal vertex in G.*

That is, G_k is the subgraph induced by the set of edges that are in some path of length at least k from a minimal to a maximal vertex in G.

Proof. Suppose, to the contrary, there exists an edge in G which is in some path of length at least k (denote k-path), $v_{\min} = v_0, ..., v_l = v_{\max}$, yet it does not appear in G_k. Let (v_{i-1}, v_i) be an edge with the maximum value of i among all such edges. If v_i is a maximal vertex then (v_{i-1}, v_i) is the last edge on a k-path and v_i would be selected for a traceback by the algorithm. The algorithm selects all the vertices x such that $(x, v_i) \in E$ and $P(x, l) > 0$ for $l \geq k - 1$. Certainly v_{i-1} is such a vertex and would be selected. Suppose then v_i is not a maximal vertex. Since (v_{i-1}, v_i) is on a k-path then the edge (v_i, v_{i+1}) is also on a k-path. Since (v_{i-1}, v_i) was chosen to have the maximum i among all unselected edges then the edge (v_i, v_{i+1}) must be in G_k. By the same argument, every edge (v_j, v_{j+1}) for $j \geq i$ is in G_k. However, every edge in G_k is the result of a traceback from a maximal vertex, including $v_{\max}$. $P(v_{\max}, l) > 0$ and $l \geq k$, therefore the algorithm *will* trace back to v_i with the value $P(v_i, l - i)$. Since $P(v_i, l - i) > 0$, the trace will continue to all $(x, v_i) \in E$ such that $P(x, l - i - 1) > 0$. But v_{i-1} is one of such vertices x and the edge (v_{i-1}, v_i) will be selected by the traceback. This is a contradiction, therefore every edge on a k-path in G will be present in G_k. ☐

Corollary 2. *G_k contains all the paths of length at least k from a minimal to a maximal vertex in G.*

The running time of the k-Graph algorithm is $O(|E||V|)$ since each edge is traced exactly once and $T \leq |V|$. Using the function k-Graph we can state the Min T-Path Vertex Shattering Set solution.

Alg Min T-Path Vertex Shattering Set G_T = k-Graph(T)
Add a vertex s connected to all the minimal vertices and a vertex t connected from all the maximal vertices in G_T
U = minimal vertex cut in the resulting graph $G_T + \{s, t\}$
RETURN U.

Theorem 3. *Algorithm* Min T-Path Vertex Shattering Set *is a polynomial time algorithm that returns the smallest subset of vertices U whose removal leaves a subgraph with no path of length T (where T is the length of the longest path in the original graph G).*

Proof. We prove the theorem using the following two lemmas.

Lemma 2. *G_T contains all and only paths of length T from a minimal to a maximal vertex in G.*

Proof. By Lemma 1, every edge in G_T is in some path of length T in G. Moreover, since T is the length of a longest path in G, every path from a minimal to a maximal vertex in G_T is of length T. Suppose, to the contrary, there is a path $v_{\min} = v_0, ..., v_l = v_{\max}$ which is shorter than T. For every edge (v_{i-1}, v_i) at least one of the following must hold:

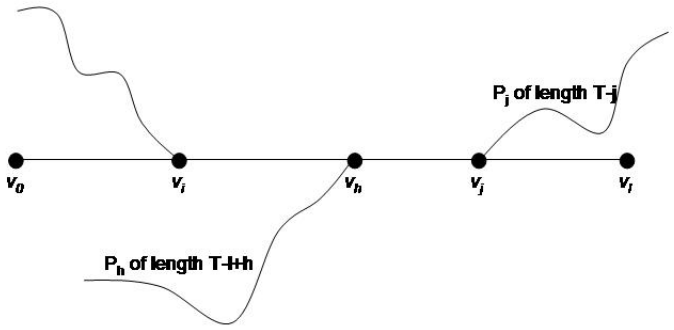

Figure 4. Illustration of the proof of Lemma 2.

1. there is a path of length $T - i$ from v_i to a maximal vertex, or

2. there is a path of length $T - l + i - 1$ from a minimal vertex to v_{i-1}.

Let v_i be the vertex with the smallest i such that its incoming degree in G_T is greater than 1. (There must be such a vertex, otherwise every vertex v_i, including v_l, has only one incoming path, namely the path from $v_{\min}$, and therefore every edge (v_{i-1}, v_i) is only on that unique path from $v_{\min}$ to $v_{\max}$ which is of length $l < T$.) Since there is no path of length $T-l+i-1$ from a minimal vertex to the vertex v_{i-1}, there must be a path of length $T - i$ from v_i to a maximal vertex. It cannot be a path $v_i, v_{i+1}, ..., v_l$ since that one is of length $l - i < T - i$. Thus, there must be a vertex v_j, $j \geq i$ that has outgoing degree greater than 1 (otherwise there is only one path from v_i and it is the path to $v_{\max}$) and a path from v_j to a maximal vertex of length $T-j$. Call that path P_j. Let v_j be such a vertex with the largest j. Consider the edge (v_j, v_{j+1}). Since there is no path from v_{j+1} to $v_{\max}$ of length $T - j - 1$, there must be a path of length $T - l + j$ from a minimal vertex to v_j. It cannot be the path $v_0, v_1, ...v_j$ since it is shorter than $T - l + j$. Thus, there must be a vertex v_h, $h \leq j$ with an incoming path of length $T - l + h$. Call that path P_h. Figure 4 illustrates the situation. But consider the following path: P_h, followed by the path $v_h, v_{h+1}, ..., v_j$, followed by P_j. The total length of this path is $(T - l + h) + (j - h) + (T - j) = 2T - l$. Since $l < T$, then $2T - l > T$, which is a contradiction to the fact that T is the length of the longest path in G. Thus, every path from a minimal to a maximal vertex in G_T is of length T. Together with the fact that every edge from a path of length T in G is contained in G_T we conclude that G_T contains all and only paths of length T from a minimal to a maximal vertex in G. $\square$

Lemma 3. *Let $G_T + \{s, t\}$ be the graph G_T with the addition of a vertex s connected to every minimal vertex in G_T and a vertex t connected from every maximal vertex in G_T. Then $U \subset V$ is a vertex cut in $G_T + \{s, t\}$ if and only if $G - U$, the subgraph of G induced by $V - U$, has no paths of length T.*

Proof. Suppose, to the contrary, there is a path of length T in $G - U$. Since this is also a path of length T in the original graph G, by Lemma 2 it is also a path of length T from a minimal to a maximal vertex in G_T. Since it is a path in $G - U$, it does not pass through any vertex of U. Since s is connected to every minimal vertex and t is connected from every maximal vertex, it is also a path from s to t in $G_T + \{s, t\} - U$. But then by definition U is not a vertex cut in $G_T + \{s, t\}$. This is a contradiction, therefore there are no paths of length T in $G - U$.

For the other direction, suppose, to the contrary, that U is not a vertex cut in $G_T + \{s, t\}$. Then there is a path from s to t that does not pass through the vertices in U. That path is also a path from a minimal to a maximal vertex in G_T, which by lemma 2 is a path of length T in G. Since it does not pass through the vertices of U, it is also a path of length T in $G - U$, which is also a contradiction. □

We now return to the proof of Theorem 3. From Lemma 3, U is a vertex cut in $G_T + \{s, t\}$ if and only if $G - U$ has no paths of length T. Therefore the algorithm MIN T-PATH VERTEX SHATTERING SET returns the smallest subset of vertices whose removal from G leaves no paths of length T. Since a minimum vertex cut in a DAG can be computed in polynomial time [11], it is a polynomial time algorithm. □

So far, we have focused on the goal of breaking *all* the metagroups in the population by removing the minimal number of groups. However, when resources are constrained, we may be able to remove only a limited number of groups, up to k. The goal is then to find k groups to remove so that the remaining number of metagroups is minimized. We shall call this problem CRITICAL k-GROUP SET. Of course, CRITICAL k-GROUP SET and CRITICAL GROUPS SET are related.

Theorem 4. CRITICAL k-GROUP SET *is no easier than* CRITICAL GROUPS SET.

Proof. Given an algorithm that solves the CRITICAL k-GROUP SET in time f (for a given α and β) there is an algorithm that solves the CRITICAL GROUPS SET problem in time $f \log m$ (where m is the total number of groups) for the same α and β. Clearly, we can use a binary search on the limit of groups to be removed as a parameter to the CRITICAL k-GROUP SET. The smallest number k such that the removal of k-group set leaves no metagroups is the answer to the CRITICAL GROUPS SET. □

4 Extensions and Open Problems

We have presented a general computational framework for the analysis of dynamic interaction networks. Our framework is sufficiently general to model problems from a wide-ranging set of applications, yet it is sufficiently complex to answer many queries about such a network. We have explored one of the important questions, namely finding critical groups whose absence destroys the existing social structure of the network, the metagroups. We have shown that in general this problem is NP-complete and have presented a polynomial time algorithm for an important special case.

In our solution we have not distinguished between different groups, just aimed to minimize the number of groups being removed while maximizing the number of destroyed metagroups. A more realistic objective is a vertex-weighted version of the problem, where the weight of a vertex is either the size of the corresponding group or its importance.

A generalization of our framework includes the concept of *membership* in a group, where an individual is a member of a metagroup if it appears in that metagroup at least γ times. Here γ may be as simple as a constant or as complex as a function of T, the size of the metagroup, average or maximum number of times any individual appears in any metagroup, etc. With this definition of membership the problems become more complicated. For example, in the case of Critical Groups Set, removing the groups that have the minimum total membership is not complement to maximizing the total membership of the remaining groups.

The issue of fragility of a dynamic social network may also be asked in terms of removing individuals, rather than groups. That is, Critical Individuals Set problem is to find the smallest number of individuals whose absence from the population leaves no metagroups. The complexity and even the feasibility of the solution of this problem depends on the the parameters α, β, γ. However, it is possible to make general observations based on our framework that address this problem. For example, there is a straight forward pseudo-polynomial algorithm to solve this problem for any α, β, γ.

There are many other questions one may ask about a dynamic society that can be easily modeled computationally in our framework. We gave but a few examples here. We believe this is a very useful tool in computational modeling of various dynamic interactions and there are many deep and important open questions that require novel algorithm design techniques.

Acknowledgments

We are grateful to S. (Muthu) Muthukrishnan, Dan Rubenstein, Siva Sundaresan, Ilya Fischoff, David Kempe, Simon Levin and many others for their help in designing this dynamic social analysis framework.

This work is supported by the National Science Foundation postdoctoral fellowship grant EIA 02-05116 (Tanya Berger-Wolf), by the National Science Foundation grant CCR-0313160 (Jared Saia), and by the Sandia University Research Program Grant No. 191445 (Jared Saia).

References

[1] E. Airoldi and B. Malin. Data mining challenges for electronic safety: the case of fraudulent intent detection in e-mails. In *Proceedings of the Privacy and Security Aspects of Data Mining Workshop, in conjunction with the 4th IEEE International Conference on Data Mining*, Brighton, England, 2004.

[2] J. Baumes, M. Goldberg, M. Magdon-Ismail, and W. Wallace. Discovering hidden groups in communication networks. In *Proceedings of the 2nd NSF/NIJ Symposium on Intelligence and Security Informatics*, 2004.

[3] S. Bernstein, A. Clearwater, S. Hill, C. Perlich, , and F. Provost. Discovering knowledge from relational data extracted from business news. In *Proceedings of the Workshop on Multi-Relational Data Mining, in conjunction with the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[4] A. Broido and K. Claffy. Internet topology: connectivity of ip graphs. In *Proceedings of SPIE ITCom*, 2001.

[5] K. Carley. Communicating new ideas: The potential impact of information and telecommunication technology. *Technology in Society*, 18(2):219–230, 1996.

[6] K. Carley and M. Prietula, editors. *Computational Organization Theory.* Lawrence Erlbaum associates, Hillsdale, NJ, 2001.

[7] L. Chen and K. Carley. The impact of social networks in the propagation of computer viruses and countermeasures. *IEEE Trasactions on Systems, Man and Cybernetics*, forthcoming.

[8] D.P Croft, J. Krause, and R. James. Social networks in the guppy (poecilia reticulata). *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 271:516–519, 2004.

[9] Paul C. Cross, James O. Lloyd-Smith, and Wayne M. Getz. Disentangling association patterns in fissionfusion societies using african buffalo as an example. *Animal Behaviour*, 69:499–506, 2005.

[10] S. Eubank, H. Guclu, V.S. Kumar, M.V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:429:180–184., Nov 2004. Supplement material.

[11] S. Even and R.E. Tarjan. Network flow and testing graph connectivity. *SIAM Journal on Computing*, 4:507–518, 1975.

[12] N. Hummon and K. Carley. Social networks: As normal science. *Social Networks*, 15:71–106, 1993.

[13] P. Jaccard. The distribution of flora in the alpine zone. *The New Phytologist*, 11(2):37–50, 1912.

[14] M. Keeling. The effects of local spatial structure on epidemiological invasions. *Proc. R. Soc. Lond. B*, 266:859–867, 1999.

[15] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[16] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the Thirty-second ACM Symposium on Theory of Computing (STOC)*, 2000.

[17] J. Kleinberg. Small-world phenomena and the dynamics of information. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence,*, Morgan Kaufman, 2001.

[18] J. Kleinberg. The small-world phenomenon and decentralized search. *SIAM News*, 37, 2004.

[19] Gina Kolata. Ideas and trends; enron offers an unlikely boost to e-mail surveillance. *New York Times*, May 22 2005.

[20] M. Kretzschmar and M. Morris. Measures of concurrency in networks and the spread of infectious disease. *Math. Biosci.*, 133:165–195, 1996.

[21] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *Proc. 20th Int. Colloquium on Automata*, volume 700 of *Lecture Notes in Comput. Sci.*, pages 40–51. Springer-Verlag, 1993.

[22] D. Lusseau and M. E. J. Newman. Identifying the role that individual animals play in their social network. *Proc. R. Soc. London B (Suppl.)*, 271:S477–S481, 2004.

[23] Newman M. and Girvan M. Finding and evaluating community structure in networks. *Phys. Rev.*, 69, 2004.

[24] M. Magdon-Ismail, M. Goldberg, W. Wallace, and D. Siebecker. Locating hidden groups in communication networks using hidden markov models. In *Proceedings of the International Conference on Intelligence and Security Informatics (ISI 2003)*, Tucson, AZ., 2003.

[25] B. Malin. Data and collocation surveillance through location access patterns. In *Proceedings of the North American Association for Computational Social and Organizational Science (NAACSOS) Conference*, Pittsburgh, PA, June 2004.

[26] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[27] M. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev.*, 69, 2004.

[28] C. Papadimitriou. Computational aspects of organization theory. *Lecture Notes in Computer Science*, 1997.

[29] C. Papadimitriou and E. Servan-Schreiber. he origins of the deadline: Optimizing communication in organizations. *Complexity in Economics.*, 1999.

[30] M. Tsvetovat, K. Sycara, Y. Chen, and J. Ying. Customer coalitions in electronic marketplaces. In Ulises Cortes Frank Dignum, editor, *Agent-Mediated Electronic Commerce III, Lecture Notes on Artificial Intelligence.* Springer-Verlag, 2003.

[31] J.R. Tyler, D.M. Wilkinson, and B.A. Huberman. Email as spectroscopy: Automated discovery of community structure within organizations. In *Proceedings of the First International Conference on Communities and Technologies*, 2003.

[32] S. Wasserman and Faust K. *Social Network Analysis.* Cambridge University Press, Cambridge, MA, 1994.

[33] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[34] B. Wellman. An electronic group is virtually a social network. In Sara Kiesler, editor, *Culture of the Internet*, pages 179–205. Lawrence Erlbaum, Mahwah, NJ, 1997.