Constant-Factor Approximation Algorithms for Identifying Dynamic Communities

Chayant Tantipathananandh
Dept. of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
ctanti2 @uic.edu

Tanya Berger-Wolf*
Dept. of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
tanyabw@uic.edu

ABSTRACT

We propose two approximation algorithms for identifying communities in dynamic social networks. Communities are intuitively characterized as "unusually densely knit" subsets of a social network. This notion becomes more problematic if the social interactions change over time. Aggregating social networks over time can radically misrepresent the existing and changing community structure. Recently, we have proposed an optimization-based framework for modeling dynamic community structure. Also, we have proposed an algorithm for finding such structure based on maximum weight bipartite matching. In this paper, we analyze its performance guarantee for a special case where all actors can be observed at all times. In such instances, we show that the algorithm is a small constant factor approximation of the optimum. We use a similar idea to design an approximation algorithm for the general case where some individuals are possibly unobserved at times, and to show that the approximation factor increases twofold but remains a constant regardless of the input size. This is the first algorithm for inferring communities in dynamic networks with a provable approximation guarantee. We demonstrate the general algorithm on real data sets. The results confirm the efficiency and effectiveness of the algorithm in identifying dynamic communities.

Categories and Subject Descriptors: F.2.0 [Analysis of Algorithms and Problem Complexity]: General

General Terms: Algorithms.

Keywords: Approximation Algorithms, Community Identification, Dynamic Social Networks.

1. INTRODUCTION

Social networks are graphs representing interactions among individuals and have been widely used as the abstraction of choice in various social studies of human and animal populations. Edges can represent social interactions, organizational structures, physical proximity, or even more abstract interactions such as hyperlinks or similarity. Social networks have attracted a large amount of attention from epidemiologists [25, 29, 31], sociologists [5, 32, 38], ecologists (animal behavior) [8, 7, 12, 33, 36], the intelligence community (terrorism networks) [3, 27, 28], and more recently also from computer scientists [1, 14, 18, 21, 22, 23, 26], to name some. While social creatures interact in diverse ways, some of the interactions are accidental while others are a consequence of the underlying explicit or implicit social structures. One of the most important questions in sociology is the identification of such structures, or "communities", which are loosely defined as collections of individuals who interact unusually frequently [17, 16, 20, 22, 30, 38]. The identification of communities often reveals interesting properties shared by the members, such as common hobbies, social functions, occupations, etc. In a more general setting, including hyperlinked documents such as the WWW, these properties include related topics or common viewpoints, which has led to a large amount of research on identifying communities in the web graph or similar settings [19, 26].

In analyzing social networks one property, until recently, has largely been ignored: interactions change over time. When faced with dynamic social networks, most studies either analyze a snapshot of a single point in time, or an aggregation of all interactions over a possibly large time window. Both approaches may miss important tendencies of these dynamic networks; indeed, the ongoing change of a network and its possible causes may be among the most interesting properties to observe. The necessity to delve into the dynamic aspects of networking behavior may be clear, yet it would not be feasible without the data to support such explicitly dynamic analysis. Rapidly growing electronic networks, such as emails, the Web, blogs, and friendship sites, as well as mobile sensor networks on cars and animals, provide an abundance of dynamic social network data that for the first time allow the temporal component to be explicitly addressed in network analysis.

Recently, Berger-Wolf and Saia [4] proposed a framework for identifying communities in dynamic social networks, making explicit use of temporal changes. Most communities tend to evolve gradually over time [2], as opposed to assembling or disbanding spontaneously. Thus, whenever temporal information about events in the social network is available, it is desirable to use this information in order to identify not only communities with high intra-community similarity,

*Work supported in part by NSF grants IIS-0705822 and CAREER IIS-0747369

KDD'09, June 28–July 1, 2009, Paris, France.

but also observe their persistence and development in time. In [35] Sun *et al.* propose a clustering approach based on minimizing compressed description of the dynamic network. Such clusters may be considered to be dynamic communities. In [37] we have proposed the first formal framework for identifying communities in dynamic networks. We formalized the problem of dynamic community identification, proved that it is NP-complete and APX-hard, and proposed several practical heuristics.

In this paper, we show that, under the assumption of no missing data, one of the algorithms presented in [37] is a small constant factor approximation. We use a similar idea to design an approximation algorithm for the general setting with possible missing data. We show that the approximation ratio remains constant, albeit increases twofold. This is the first approximation algorithm with provable performance guarantee for identifying dynamic communities. While the theoretic analysis provides an upper bound on the worst case performance of the approximation algorithm, we show that in practice the algorithm performs very well, producing a solution close to the optimum. We improve the solution in practice (though not theoretically) even further by applying the Dynamic Programming approach in the second stage of the algorithm. We evaluate the practical performance of our algorithm on several real world dynamic networks, spanning a wide range of parameters and sizes. The algorithm performs consistently well, demonstrating that it is a viable practical approach to inferring dynamic communities, including in networks of several thousand nodes.

2. PRELIMINARIES

2.1 Notations and Definitions

We model social interactions as graphs over nodes representing individuals. To model dynamic interactions, we use the model of [37] which follows and slightly extends the approach of Berger-Wolf and Saia [4] and that of affiliation networks [38, Section 8].

Let $\mathcal{I} = \{1, 2, \ldots, n\}$ denote the set of individuals and $\mathcal{T} = \{1, 2, \ldots, T\}$ denote the set of discrete time steps. Let $\mathcal{H} = \langle H_1, H_2, \ldots, H_T \rangle$ be an interaction sequence of n individuals over T times where $H_t = \{g_{t,j}\}$ is the collection of groups of the individuals observed at time t. The interpretation is that the individuals in group $g_{t,j}$ are observed interacting among themselves at time t. Such a group may correspond to a physical or virtual gathering of its members, such as a committee meeting or writing a joint paper.

We stress that, in our terminology, groups and communities are not the same: groups capture only a snapshot of interaction at one point in time, while communities are latent concepts which should explain many of the actual observed interactions, though not necessarily all of them.

Lastly, we use colors to identify communities. For each time t, a *community interpretation* of the snapshot H_t is a coloring $\chi_t : H_t \cup \mathcal{I} \to \mathbb{N}$ of the groups and individuals. Groups of the same color represent the same community. The color of an individual indicates with which community it is affiliated at the time. Note that χ_t assigns a color to all individuals, both observed and unobserved at time t. In this way, we can determine whether it is possible for an individual to retain its affiliation during the unobserved times. A community interpretation [37] of the entire interaction sequence $\mathcal{H}$ is a set of colorings $\chi = \{\chi_t\}$ of all H_t.

2.2 Problem Formulation

In general, we formulate the problem of identifying dynamic communities as a class of optimization problems: Given an interaction sequence $\mathcal{H}$, find an optimal community interpretation χ of $\mathcal{H}$ with respect to some measure of goodness. One way to define goodness of a community interpretation is to consider the social costs that individuals incur in the course of interactions. These social costs are, for example, *switching* a community affiliation (and thus, possibly, loosing friends or social status), *visiting* a community with which one is not affiliated (and feeling uncomfortable, out of place, or without supporters), and being *absent* from a group representing one's community (and possibly missing important community information or opportunity). For a society's history of interactions, we aim to identify the most parsimonious underlying communities. The inferred community structure should explain as many of the observed interactions as possible. Stated conversely, we seek the community structure that minimizes those interactions considered to be among individuals in separate communities and, thus, minimizes the overall social costs of switching, visiting, and absence incurred by individuals.

In formulating the problem, in addition to this philosophical view of dynamic communities, we make some technical assumptions (discussed fully in [37]). We assume that the groups at each time are pairwise disjoint. In other words, each snapshot of interactions H_t is a partition of the observed subset of $\mathcal{I}$. We also consider only the community interpretations that regard groups at each time as distinct communities (otherwise, the individuals in different groups would be interacting together as one group). In [37], we formalized this view of finding a good community interpretation as the Minimum Community Interpretation problem. Given the value of the costs of *switching* $\alpha \geq 0$, being *absent* $\beta_1 \geq 0$, and *visiting* $\beta_2 \geq 0$, the objective is to minimize the sum of switching, absence, and visit costs incurred by all individuals, as well as the number of colors that each individual has. We justified and validated this definition on several data sets. We proved that Minimum Community Interpretation problem is NP-complete and APX-hard. In this paper, we generalize the problem by dropping the number of colors from the objective function and leaving just the sum of the three social costs. It is easy to verify that the problem remains NP-complete and APX-hard.

To formally state the problem, we construct a *cost graph* whose vertices are to be colored. It is a graph $G = (V, E)$ with $V = V_1 \cup \cdots \cup V_T$ where V_t is the set of vertices corresponding to the groups and individuals at time t. Specifically, in each time t the set V_t contains vertices $g_{t,j}$ for all groups $g_{t,j} \in H_t$ and vertices i_t for all individuals $i \in \mathcal{I}$.

There are three kinds of edges corresponding to the three social costs: $E = E_\alpha \cup E_{\beta_1} \cup E_{\beta_2}$.

1. E_α contains *switching* edges (i_t, i_{t+1}), corresponding to the events of the individual i switching its community affiliation between times t and $t + 1$.
2. E_{β_1} contains *absence* edges (i_t, g) for all $i \in \mathcal{I}$ and $g \in H_t$ such that $i \notin g$, corresponding to individual i being absent from group g.
3. E_{β_2} contains *visit* edges (i_t, g) for all $i \in \mathcal{I}$ and $g \in H_t$ such that $i \in g$, corresponding to i visiting group g.

The induced subgraph $G[V_t]$ of the cost graph within time step t is essentially a complete bipartite graph where there

are absence and visit edges between the group vertices and the individual vertices. Note there can be at most one visit edge incident on each individual vertex since we assume that groups within each time step are pairwise disjoint. Figure 1 shows an example of an interaction sequence and the corresponding cost graph representation, with a coloring showing the social costs in the setting according to the interpretation.

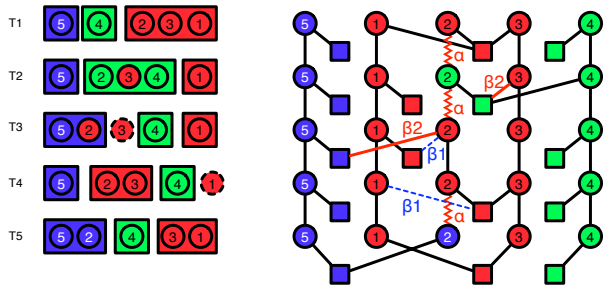


Figure 1: An example of an interaction sequence (left) and the corresponding cost graph (right). In the interaction sequence, each row corresponds to a time step, with time going from top to bottom. Each rectangle represents an observed group and the circles within are the member individuals. The circles outside the rectangles are the unobserved individuals. Similarly, in the cost graph, the squares are group vertices and the circles are individual vertices. Not all edges that incur cost are drawn for visibility. The coloring shows the costs of switching (α), absence (β_1), and visiting (β_2).

For a coloring χ, let χ_{uv} be the indicator variable which equals to 1 if the vertices u and v have the same color and 0 otherwise. Since we assume that groups within a time step are manifestations of distinct communities, we say that a coloring χ is *valid* if it assigns distinct colors to the groups in each time step $t = 1, \ldots, T$. We define the cost $c(\chi)$ of a valid coloring χ as the total cost incurred by all the switches, absences, and visits:

$$c(\chi) = \alpha \sum_{uv \in E_\alpha} (1 - \chi_{uv}) + \beta_1 \sum_{uv \in E_{\beta_1}} \chi_{uv} + \beta_2 \sum_{uv \in E_{\beta_2}} (1 - \chi_{uv}).$$

The Minimum Community Interpretation problem is, given an interaction sequence $\mathcal{H}$ and costs α, β_1, $\beta_2 \geq 0$, to find a minimum cost valid coloring of the corresponding cost graph.

3. APPROXIMATION ALGORITHMS

As we have noted, the Minimum Community Interpretation problem is NP-complete. In this section, we present two approximation algorithms: the first solves a special case of the problem and the other solves the general problem. Both algorithm produce a solution within a constant factor of the optimum, regardless of the input size. Generally, for a minimization problem, a ρ-approximation algorithm is an algorithm which, on all instances of the problem, always produces in polynomial time a solution which is at most ρ times of the optimal solution.

The idea for the algorithm is to deal with one type of cost at a time. Note, that for the optimization it is not the absolute but the relative values of the costs which are important. If the switching cost α is much higher than either β (*e.g,* death for betrayal) then individuals will never switch the color from that of the starting community and would prefer to incur the absence and the visit costs. On the other hand, if the absences and visits are much more expensive than a switch (*e.g.*, a customer membership program) then individuals would rather switch their affiliation when necessary. In this case, we would like to find a community interpretation which minimizes the number of switches. We begin by considering a special case with the assumption that every individual is observed at all times. Under this assumption, we present an algorithm based on maximum weight bipartite matching and show that it is a ρ_1-approximation where $\rho_1 = \alpha / \min\{\alpha, \beta_2/2\} = \max\{1, 2\alpha/\beta_2\}$. In Section 3.3, we consider the general problem where some individuals might be unobserved at some point in time. We present another algorithm based on minimum weight path cover problem and show that it is a ρ_2-approximation where $\rho_2 = 2\alpha / \min\{\alpha, \beta_1, \beta_2/2\} = \max\{2, 2\alpha/\beta_1, 4\alpha/\beta_2\}$.

3.1 Group Graph

To design the algorithm we use another auxiliary graph. The *group graph* of an interaction sequence $\mathcal{H}$ is a directed acyclic graph $D = (V, E)$. The intuition is that the group graph represents how the individuals *flow* from one group to another over time. Given an interaction sequence $\mathcal{H}$, we create the group graph as follows.

For a technical reason, we add dummy groups as follows. For each (real) group g observed at time $t \geq 2$, let $g' \subseteq g$ be the individuals in g who are observed for the first time in g If g' is not empty, then we add a dummy group g' at time 1. Similarly, for each (real) group g observed at time $t \leq T$, let $g' \subseteq g$ now be the individuals in g who are observed for the last time in g. If g' is not empty, then we add a dummy group g' at time T. When all individuals are observed at all times, there are no dummy groups.

Next, we create a vertex $g \in V$ for every real or dummy group g. We create edges going out from each vertex $g \in V$ as follows. For each individual $i \in g$, we find the next group h containing i such that i does not appear in any other groups in between. If the edge (g, h) has not been created, we create it and set its label to be $\lambda(g, h) = \{i\}$. If there already is an edge (g, h), we update its label to $\lambda(g, h) \cup \{i\}$. Finally, we set the edge weight to be $w(g, h) = |\lambda(g, h)|$. Intuitively, the set $\lambda(g, h)$ contains the individuals that *flow* along the edge (g, h) and the edge weight $w(g, h)$ signifies the amount of the flow. From now on, we use the words groups and the vertices of the group graph interchangeably.

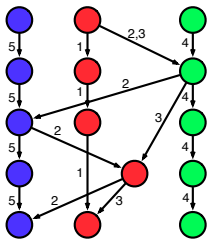


Figure 2: The group graph that corresponds to the interaction sequence and the cost graph in Figure 1. The edges are labeled with the individuals in $\lambda(g, h)$. There are no dummy vertices since every individual is observed at the first and the last times.

Given an interaction sequence $\mathcal{H}$, its corresponding group graph D can be created in polynomial time by the follow-

ing simple algorithm. We note that the algorithm runs in $\Theta(Tk^2)$ time. This cannot be improved since it is possible that $w(g_i, g_j) > 0$ for every $i < j$.

Algorithm 1 CreateGroupGraph

Require: interaction sequence $\mathcal{H}$.
1: $g_1, \ldots, g_k \leftarrow$ the real and dummy groups in $\mathcal{H}$ in increasing order of time, breaking ties arbitrarily.
2: $V \leftarrow \{g_1, \ldots, g_k\}, E \leftarrow \emptyset$
3: **for** $i = 1, \ldots, k-1$ **do**
4: $\quad A \leftarrow g_i$
5: $\quad$ **for** $j = i+1, \ldots, k$ **do**
6: $\quad\quad$ **if** $g_j \cap A \neq \emptyset$ **then**
7: $\quad\quad\quad E \leftarrow E \cup \{(g_i, g_j)\}$
8: $\quad\quad\quad w(g_i, g_j) \leftarrow |A \cap g_j|$
9: $\quad\quad\quad A \leftarrow A \setminus g_j$
10: $\quad\quad$ **end if**
11: $\quad$ **end for**
12: **end for**
13: **return** $D = (V, E)$

We note the similarity between the group graph and the meta-group graph by Berger-Wolf and Saia [4]. In particular, the group graph is a transitive reduction (or Hasse diagram) of the meta-group graph.

3.2 Approximation via Bipartite Matching

In this section, we assume that all individuals are observed at all time steps. The algorithm for identifying communities in dynamic networks was first presented without analysis in [37] and is as follows.

Algorithm 2 MatchingCommunities (MC)

Require: An interaction sequence $\mathcal{H}$.
1: $D = (V, E) \leftarrow$ undirected version of the group graph of $\mathcal{H}$, dropping edge orientations.
2: **for** time $t = 1, \ldots, T-1$ **do**
3: $\quad G_t \leftarrow$ bipartite subgraph of D induced by $V_t \cup V_{t+1}$ where V_t is the set of groups at time t.
4: $\quad M_t^* \leftarrow$ maximum weight matching on G_t.
5: **end for**
6: $D' = (V, E') \leftarrow$ the group graph D with the edge set replaced by the matched edges $E' = \cup_{t=1}^{T-1} M_t^*$.
7: Color (the groups in) each connected component of D' by a distinct color.
8: Color each individual at each time step by the same color as the group in which it was observed so that all groups are monochromatic.

The heart of the above algorithm is finding a maximum weight matching on bipartite graphs, which can be done efficiently, in particular, in polynomial time [24]. Coloring the groups according to the matching can be done in time linear in $|V|$ and $|E|$. Coloring the individuals can be done in time linear in n and T. Thus, the overall time of the Algorithm MC is bounded by the time of finding a maximum weight matching on bipartite graphs.

3.2.1 Performance Analysis

Recall that, to show that an algorithm is a ρ-approximation for a minimization problem we need to give two bounds. For any input, we would like an upper bound U on the cost $c(S)$ of a solution S produced by the algorithm, and a lower bound L on the cost $c(S^*)$ of the optimal solution S^*. If $U/L \leq \rho$, then we have that $\frac{c(S)}{c(S^*)} \leq \frac{U}{L} \leq \rho$, and the algorithm is a ρ-approximation.

We first give an upper bound on the cost of a coloring produced by Algorithm MC. Then, we give a lower bound on the cost of an optimal coloring. For any set of edges M, we write $w(M) = \sum_{e \in M} w(e)$ to denote its total weight.

Proposition 1. *Let $\mathcal{H}$ be an interaction sequence with all individuals present at all times. Let $M_1^*, \ldots, M_{T-1}^*$ be the matchings that produces a coloring χ of $\mathcal{H}$ in Algorithm* MC. *Then, the cost of χ is*

$$c(\chi) = \alpha \sum_{t=1}^{T-1} (n - w(M_t^*)).$$

Proof. We note that Algorithm MC colors the groups and individuals such that all groups are monochromatic. Thus, the resulting coloring χ does not have any absence or visit costs, and has only the switching costs. Consider each time $t \leq T-1$. The individuals who incur switching costs are those whose groups at times t and $t+1$ are not matched by M_t^*. Since there are $n - w(M_t^*)$ such individuals, the proposition holds. □

Now, we give a lower bound on the cost of an optimal coloring by giving a lower bound on the cost of any valid coloring of $\mathcal{H}$.

Lemma 2. *Let $\mathcal{H}$ be an interaction sequence with all individuals present at all times. Let χ be a valid coloring of $\mathcal{H}$ with cost $c(\chi)$. Let $G_1, \ldots, G_{T-1}$ be as in Algorithm* MC. *Let $\mu_1 = \min\left\{\alpha, \frac{\beta_2}{2}\right\}$ for convenience. Then, there exist matchings $M_1, \ldots, M_{T-1}$ on $G_1, \ldots, G_{T-1}$, respectively, such that*

$$c(\chi) \geq \mu_1 \sum_{t=1}^{T-1} (n - w(M_t)). \tag{1}$$

Proof. For any valid coloring χ, let $c_t(\chi)$ denote the cost that χ incurs between times t and $t+1$. Since there are no absence costs, this can include three possible types of costs: one for switching color between t and $t+1$ and two for visiting other communities at time t or $t+1$. We claim that, for a valid coloring χ and time $t \leq T-1$, there exists a matching M_t on G_t such that

$$c_t(\chi) \geq \mu_1 (n - w(M_t)). \tag{2}$$

Let M_t be the matching containing all the edges whose end points (groups) are colored the same in χ. This is well-defined since, for any color a of χ, there can be at most one group from each time colored a, since χ is a valid coloring. Thus, a vertex in V_t is matched to at most one vertex in V_{t+1} and vice versa.

Now, we consider each edge (g, h) of G_t unmatched by M_t. Since groups g and h have different colors in χ, each individual $i \in g \cap h$ must incur at least μ_1. This is so since if i switches colors, then i incurs α. Otherwise, i must visit g or h (or both) and, thus, incurs β_2 which is at least $\frac{\beta_2}{2}$. The reason for the half factor is that, for each visiting cost at time t, we might count it twice: once in c_{t-1} and once in c_t. Thus i incurs at least μ_1. Since there are $n - w(M_t)$ individuals whose groups are unmatched, inequality (2) holds as claimed. Since we count every cost no more than once, $c(\chi) \geq \sum_{i=1}^{T-1} c_t(\chi)$. Now, the lemma follows. □

Now we show approximation factor of the Algorithm MC.

THEOREM 3. *For convenience, let*

$$\mu_1 = \min\left\{\alpha, \frac{\beta_2}{2}\right\}, \qquad \rho_1 = \frac{\alpha}{\mu_1} = \max\left\{1, \frac{2\alpha}{\beta_2}\right\}.$$

Given an interaction sequence $\mathcal{H}$ with all individuals present at all times, Algorithm MC *produces, in polynomial time, a coloring with cost at most ρ_1 times of the optimal.*

PROOF. Let χ^* be an optimal coloring of $\mathcal{H}$. Let $M = \{M_t\}$ be the set of matchings as in Lemma 2. For convenience, let $\bar{w}(M) = \sum_{t=1}^{T-1}(n - w(M_t))$ (and thus, by Lemma 2, $\mu_1\bar{w}(M) \leq c(\chi^*)$). Let χ be the coloring produced by Algorithm MC. Let $M^\chi = \{M_t^\chi\}$ be the set of matchings used in producing χ and $\bar{w}(M^\chi) = \sum_{t=1}^{T-1}(n - w(M_t^\chi))$. We observe that,

$$\mu_1 \cdot \bar{w}(M^\chi) \leq \mu_1 \cdot \bar{w}(M) \leq c(\chi^*) \leq c(\chi) = \alpha \cdot \bar{w}(M).$$

The first inequality holds since M_t^χ are of maximum weight, the second follows from Lemma 2, and the third follows from the optimality of χ^*. The last equality holds by Proposition 1. Since, $\frac{c(\chi^*)}{c(\chi)} \leq \frac{\alpha}{\mu_1} = \rho_1$, the theorem follows. □

If $\alpha \geq \frac{\beta_2}{2}$, then $\rho_1 = 1$ and the algorithm always produces an optimal coloring. Note, that it is straightforward to convert Algorithm MC into a streaming algorithm, since we only need to store the groups at the latest time and their color while using space that is constant in n and T.

3.3 Approximation via Path Cover

In the previous section we made an assumption that all individuals are observed at all time steps. This assumption does not always hold for real data. In this section, we consider the general case of the problem in which some individuals might be unobserved at times. We present a ρ_2-approximation algorithm for the general case and analyze its performance guarantee. Before describing the algorithm, we recall the definition of a path cover of a graph.

3.3.1 Path Cover Problem

In a directed graph $D = (V, E)$, a directed path is a sequence of distinct vertices $P = v_1, \ldots, v_k$ such that (v_i, v_{i+1}) is an edge of D for every $i = 1, 2, \ldots, k-1$. Two directed paths P_1 and P_2 are vertex-disjoint if they share no vertices. A path cover $\mathcal{P}$ on D is a set of pairwise vertex-disjoint paths [10] in which every vertex lies on (covered by) *exactly* one path in $\mathcal{P}$. The MINIMUM PATH COVER problem is to find a path cover with the minimum number of paths. The decision version of the problem on general graphs is NP-complete [15]. However, on directed acyclic graphs (DAGs), the problem can be solved in polynomial time via a reduction to the matching problem in bipartite graphs [6].

Equivalently, the problem is to find the maximum the number of edges covered by $\mathcal{P}$. For a weighted directed graph $D = (V, E)$ with edge weights $w(e)$, the objective of the path cover is to maximize the total weight of covered edges, $w(\mathcal{P}) = \sum_{P\in\mathcal{P}}\sum_{e\in P} w(e)$. It is straightforward to extend the solution approach to the weighted case.

It is more convenient to use the minimization version of the weighted path cover problem, since we define our problem of identifying communities as a minimization problem. We describe the minimum weight path cover problem as follows. Let $W = \sum_{e\in E} w(e)$ and $\bar{w}(\mathcal{P}) = W - w(\mathcal{P})$. In other words, $\bar{w}(\mathcal{P})$ is the total weight of the uncovered edges. By the definition of $\bar{w}$, maximizing $w(\mathcal{P})$ is equivalent to minimizing $\bar{w}(\mathcal{P})$. From now one, let us consider only the uncovered weight $\bar{w}(\mathcal{P})$ of a path cover $\mathcal{P}$.

3.3.2 Algorithm Description

Now we describe the approximation algorithm. We reserve one color ϵ not to be assigned to any group. Intuitively, individuals with color ϵ are considered to be in the "missing" community at the time. We refer to ϵ as the color of absence.

Algorithm 3 PATHCOVERCOMMUNITIES (PCC)

Require: An interaction sequence $\mathcal{H}$.
1: $D = (V, E) \leftarrow$CREATEGROUPGRAPH($\mathcal{H}$)
2: $\mathcal{P}^* \leftarrow$ minimum weight path cover on D.
3: Color (real groups in) each path $P \in \mathcal{P}^*$ by a distinct color.
4: **for all** edges $(g, h) \in E$ **do**
5: Color the individuals in $\lambda(g, h)$ from the time they were in g until they were in h by the same color as g.
6: **end for**
7: Color the remaining vertices by ϵ.

First, we observe that the algorithm runs in polynomial time. Furthermore, we observe that a coloring χ by Algorithm PCC is valid. If χ is not valid, then there exist two groups g, h at some time step t to which χ assign the same color. By construction, g and h must lie on the same path P for some $P \in \mathcal{P}$. Since every edge in D is oriented from some time t to another time $t' > t$, any path in D lists its group vertices in strictly increasing order of time. Thus, g and h are at different time steps, which is a contradiction.

3.3.3 Performance Analysis

As before, we first provide an upper bound on the cost of the solution, then a lower bound on the optimum. Finally we combine the two bounds to give an approximation factor.

PROPOSITION 4. *Let $\mathcal{P}^*$ be a minimum weight path cover on D with weight $\bar{w}(\mathcal{P}^*)$. Let χ be the coloring produced from $\mathcal{P}^*$ by Algorithm* PCC *with cost $c(\chi)$. Then,*

$$c(\chi) \leq 2\alpha \cdot \bar{w}(\mathcal{P}^*).$$

PROOF. By construction, χ incurs only α costs. We consider each edge (g, h) of D uncovered by $\mathcal{P}^*$ and each individual $i \in \lambda(g, h)$:

- If g, h are consecutive in time, then i incurs a cost α.
- If g, h are not consecutive in time, then i incurs 2α, for switching to the color of absence ϵ and switching back.

Thus, each uncovered edge (g, h) incurs a cost of at most $2\alpha \cdot \bar{w}(g, h)$, resulting in the total of at most $2\alpha \cdot \bar{w}(\mathcal{P}^*)$. □

To present the lower bound, we use the following notation. Let χ be a fixed coloring. For each edge (g, h) of D, we associate with it the sum of the following costs of χ:

- α for every individual $i \in \lambda(g, h)$ that switches color between the times i was in g and h.
- β_1 for every individual $i \in \lambda(g, h)$ absent from group h' after the time i was in g and before i was in h.

- $\frac{\beta_2}{2}$ for every individual $i \in \lambda(g,h)$ that visits group g.
- $\frac{\beta_2}{2}$ for every individual $i \in \lambda(g,h)$ that visits group h.

Note that we omit the cost for individual $i \in \lambda(g,h)$ being absent from some group h' at the same time as from g or h. Although we could add $\frac{\beta_1}{2}$, this suffices for our purposes.

For any subgraph $D' \subseteq D$, we define $c(\chi|D')$ to be the sum of the costs of χ associated with the edges of D' as described above. As noted above, since some costs are omitted,

$$c(\chi) \geq c(\chi|D). \tag{3}$$

This notation is useful when we decompose group graph D into pairwise edge-disjoint subgraphs $D_1, \ldots, D_k$ such that $D = \cup_j D_j$. It is easy to see that,

$$c(\chi|D) \geq \sum_j c(\chi|D_j). \tag{4}$$

When it is clear from the context, we write $c(\chi|E)$ to denote $c(\chi|D)$ where E is the edge set of D.

We also use a similar notation for path cover. For any path cover $\mathcal{P}$ on group graph D, we write $\bar{w}(\mathcal{P}|D)$ to emphasize that it is the total weight of the edges of D uncovered by $\mathcal{P}$. Let the group graph D be decomposed into vertex-disjoint subgraphs $D_1, \ldots, D_k$, where $D = \cup_j D_j$. Let $C = \{e \in E : \forall j\, e \notin D_j)\}$ be the set of edges that are not in any of the parts $D_1, \ldots, D_k$. Let $w(C) = \sum_{e \in C} w(e)$ be the total weight of C. Let $\mathcal{P}_j$ be any path cover on each D_j with weight $\bar{w}(\mathcal{P}_j|D_j)$. Then, $\mathcal{P} = \cup_j \mathcal{P}_j$ is a path cover on D with weight,

$$\bar{w}(\mathcal{P}|D) = \sum_j \bar{w}(\mathcal{P}_j|D_j) + w(C). \tag{5}$$

Having defined the necessary notation, we give a lower bound on the cost of any valid coloring, including the optimum.

LEMMA 5. *Let $D = (V, E)$ be the group graph of an interaction sequence $\mathcal{H}$. Let χ be a valid coloring of $\mathcal{H}$. Then, there exists a path cover $\mathcal{P}$ on D such that*

$$c(\chi|D) \geq \mu_2 \cdot \bar{w}(\mathcal{P}|D),$$

where $\mu_2 = min\left\{\alpha, \beta_1, \frac{\beta_2}{2}\right\}$.

PROOF. We show this by induction on the number of edges $|E|$. We consider the following three cases.

Case 1: D is not (weakly) connected. Then, D can be decomposed into connected components $D_1, \ldots, D_k$ for some $k \geq 2$. Since $|E(D_j)| < |E|$ for all j, there exists a path cover $\mathcal{P}_j$ on each D_j such that $c(\chi|D_j) \geq \mu_2 \cdot \bar{w}(\mathcal{P}_j|D_j)$ holds, by induction. By inequality (4) and induction,

$$c(\chi|D) \geq \sum_j c(\chi|D_j) \geq \mu_2 \sum_j \bar{w}(P_j|D_j).$$

Let $\mathcal{P} = \cup_j \mathcal{P}_j$ be the path cover of D. Since there are no edges going between D_j's, equation (5) amounts to $\bar{w}(\mathcal{P}|D) = \sum_j \bar{w}(\mathcal{P}_j|D_j)$ and the desired result follows,

$$c(\chi|D) \geq \mu_2 \sum_j \bar{w}(P_j|D_j) \geq \mu_2 \cdot \bar{w}(\mathcal{P}|D).$$

Case 2: D is (weakly) connected and not monochromatic. Consider the partition of V induced by the coloring χ. That is, each part in the partition is the set of vertices of each color in χ. If D is not monochromatic, then χ partitions V into parts $V_1, \ldots, V_\ell$, for some $\ell \geq 2$, where V_j is the set of groups colored j. Let $D_j = D[V_j]$ be the induced subgraph of D corresponding to the monochromatic part j.

Let C be the set of edges that are not in any of the subgraphs D_j. We observe that, for each such edge $(g,h) \in C$, the groups g and h have different color in χ since they belong to different parts. By a similar argument as in the proof of Lemma 2, each individual $i \in \lambda(g,h)$ must incur the cost of at least $\mu_1 \geq \mu_2$. Thus, $c(\chi|C) \geq \mu_2 \cdot w(C)$.

Since $|E(D_j)| < |D|$ for all j, there exists a path cover $\mathcal{P}_j$ on each D_j such that $c(\chi|D_j) \geq \mu_2 \cdot \bar{w}(\mathcal{P}_j|D_j)$ holds, by induction. Let $\mathcal{P} = \cup_j \mathcal{P}_j$ be the path cover on D. By inequality (4), induction, and the above observation about $c(\chi|C)$, we obtain,

$$\begin{aligned} c(\chi|D) &\geq \sum_j c(\chi|D_j) + c(\chi|C) \\ &\geq \mu_2 \sum_j \bar{w}(P_j|D_j) + \mu_2 \cdot w(C). \end{aligned}$$

Now the desired result follows from equation (5),

$$c(\chi|D) \geq \mu_2 \sum_j \bar{w}(P_j|D_j) + \mu_2 \cdot w(C) = \mu_2 \cdot \bar{w}(P).$$

Case 3: D is (weakly) connected and monochromatic. Suppose for the moment that D has some edges. Let (g,h) be one of the shortest edges in D in the sense that g and h were observed closest in time. Let $h_1, \ldots, h_d$ be the out-neighbors of g in increasing order of time, and $g_1, \ldots, g_e$ be the in-neighbors of h in decreasing order of time. Note that $g = g_1$ and $h = h_1$.

If g has more than one out-neighbor, we consider each individual $i \in \cup_{j \geq 2} \lambda(g, h_j)$. If i switches color at any point between the time that group g was observed and the time that group h_j was observed, then i incurs α, which we map to (g, h_j). Otherwise, either i visits both groups g and h_j, or i is absent from group h (since g, h, h_j have the same color). In the former case, i incurs $2\beta_2$, half of which we map to (g, h_j). In the latter case, i incurs β_1, which we map to (g, h_j). In all cases, each $i \in \lambda(g, h_j)$ maps the cost of at least $\min\{\alpha, \beta_1, \beta_2\} \geq \mu_2$ to (g, h_j). If h has more than one in-neighbor, a similar argument also works for individuals $i \in \cup_{j \geq 2} \lambda(g_j, h)$.

Let $C = \{(g, h_j)\} \cup \{(g_{j'}, h)\}$ be the set of edges going out from g and coming into h. The above argument gives a lower bound $c(\chi|C) \geq \mu_2 \cdot w(C \setminus \{(g,h)\}) \geq \mu_2 \cdot w(C)$. We remove the edges in C from D. Let $D' = (V, E \setminus C)$ be the resulting graph. Since $(g,h) \in C$, $|E(D')| < |E|$ and exists a path cover $\mathcal{P}'$ on D' such that $c(\chi|D') \geq \mu_2 \cdot \bar{w}(\mathcal{P}'|D')$, by induction. We join the path in $\mathcal{P}'$ ending at g with the path in $\mathcal{P}'$ starting at h. The result is a path cover $\mathcal{P}$ on D. Using inequality (4), induction, and the above lower bound on $c(\chi|C)$, we obtain,

$$c(\chi|D) \geq c(\chi|D') + c(\chi|C) \geq \mu_2 \cdot \bar{w}(P'|D') + \mu_2 \cdot w(C).$$

Now the desired result follows from equation (5),

$$c(\chi|D) \geq \mu_2 \cdot \bar{w}(P'|D') + \mu_2 \cdot w(C) = \mu_2 \cdot \bar{w}(P).$$

It remains to show Case 3 for D without any edges. Since D is connected, D is a vertex. Let the path cover $\mathcal{P}$ be the entire graph D. Thus, $c(\chi|D) \geq \bar{w}(\mathcal{P}|D) = 0$ trivially holds.

Therefore, the lemma follows. □

Now we give the performance guarantee of Algorithm PCC.

THEOREM 6. *Algorithm* PCC *is a* ρ_2*-approximation where*

$$\rho_2 = \frac{2\alpha}{\mu_2} = \max\left\{2, \frac{2\alpha}{\beta_1}, \frac{4\alpha}{\beta_2}\right\}, \quad \mu_2 = \min\left\{\alpha, \beta_1, \frac{\beta_2}{2}\right\}.$$

PROOF. The theorem follows from the lower bound from Lemma 5, application of inequality (3), and the upper bound from Proposition 4. The argument is similar to the proof of Theorem 3. □

4. PRACTICAL ISSUES

In this section, we discuss the applications and scalability of Algorithm PCC.

4.1 Dynamic Programming

We observe that the only input to Algorithm PCC is an interaction sequence, it produces the same coloring regardless of the values of α, β_1, β_2. In particular, if we fix β_1, β_2 and increase α, there is a point at which no individual switches color in the optimal coloring. When α is higher than this point, increasing α does not change the cost of the optimal coloring. However, the cost by Algorithm PCC increases linearly in α. Nevertheless, there is a simple way to improve the algorithm. In particular, we use Algorithm PCC to color the groups, then use the Dynamic Programming to color the individuals [37]. In fact, since we discard the color cost, the time complexity the Dynamic Programming algorithm is improved from fixed-parameter tractable $\Theta(nTc^2 2^c)$ to polynomial $\Theta(nTc^2)$, where c the number of colors.

4.2 Iterative Path Cover Heuristic

One problem arises when we apply the Dynamic Programming. In particular, Algorithm PCC may produce a coloring which uses a large number of colors. This increases the complexity of the Dynamic Programming above. We use a simple heuristic to recolor two paths that are not overlapping in time with the same color. Due to space limitations, we very briefly describe Algorithm IterativePathCoverCommunities (IPCC) in the following.

We start each iteration with a path cover $\mathcal{P}$ which is initially set to be the trivial path cover in which each path contains a single vertex. We create another group graph D' whose vertices correspond to the paths in $\mathcal{P}$. Then, we create an edge in a similar manner as in Algorithm 1. We set the edge weight $w(P, Q)$ to be the number of individuals that the last group (vertex) of P and the first group (vertex) of Q have in common. Then, we find a minimum weight path cover P' on D'. Then, we set $P = P'$ and iterate. We stop when the constructed group graph D' contains no edges. Finally, we produce a coloring in the same way as before. We note that the first iteration is the same as PCC. Thus, the coloring produced at the end of the first iteration has cost at most ρ_2 times of the optimum. From the second iteration onward, we combine two communities that do not overlap in time, thus, incurring no additional costs. Thus, we intuitively see that the coloring produced at the end is also at most ρ_2 times of the optimum. We omit the formal proof of this claim for brevity.

5. EXPERIMENTAL RESULTS

We showed theoretically that Algorithms MC and IPCC are efficient and guarantee a good approximation. In this section, we evaluate numerically the performance of Algorithm IPCC. To find a path cover, we use a commercial package solver (CPLEX) to solve an integer program. After IPCC produces a coloring of the groups, we run the Dynamic Programming to color the individuals. We ran this on several data sets, namely, Grevy's zebra, Haggle, Onagers, Plain zebra, Reality Mining, and the Southern Women. Table 1 shows size statistics of the data sets.

Data Set	individuals	times	groups
Grevy's zebra	27	44	75
Haggle-264	264	425	1411
Haggle-41	41	418	2131
Onagers	29	82	308
Plain zebra	2510	1268	7907
Reality Mining	96	1577	3958
Southern Women	18	14	14

Table 1: Statistics of the data sets

On each data set we find the optimal solution either by exhaustive search or by estimating the lower bound on the optimum using a branch-and-bound algorithm (we omit the details of the algorithm due to space limitations). We compare the solution obtained by the IPCC algorithm to this benchmark both numerically and, where possible, structurally. For each cost setting, we compare the theoretical approximation ratio of ρ_2 to the actual ratio of the cost of the algorithm to the optimum (shown in parentheses in all the results tables). Note, that the coloring cost of the algorithm does not change when β_2 changes since the algorithm incurs only α costs. Table 2 shows the performance results of PCC and IPCC algorithms on all data sets.

5.1 Social Network Data Sets

5.1.1 Southern Women

Southern Women [9] is a data set collected in 1933 in Natchez, TN, by a group of anthropologists conducting interviews and observations over a period of 9 months. It tracks 18 women and their participation in 14 informal social events such as garden parties and card games. The data set has been extensively studied, and used as a benchmark for community identification methods [16].

5.1.2 Grevy's Zebra

The Grevy's zebra (*Equus grevyi*) data set [36] was obtained by observing spatial proximity of members of a zebra population over three months in 2002 in Kenya. Predetermined census loops were driven approximately twice per week. Individuals were identified by unique stripe patterns, and their locations taken by GPS. In the resulting data set, individuals are in the same group if their GPS locations are very close. The data set contains 28 individuals interacting over a period of 44 time steps. Many of the individuals are missing in many time steps.

5.1.3 Plains Zebra

Similar to the Grevy's zebra data set, the Plains zebra (*Equus burchelli*) data set [13] was collected in Kenya by visual scans of the populations, typically once a day, over a period of several months.

5.1.4 Onagers

Another data set obtained by observing spatial proximity of onagers (*Equus hemionus khur*) in the Little Rann of

Kutch desert in Gujarat, India [36]. The population of 29 onagers was observed from January to May in 2003.

5.2 Bluetooth Devices

5.2.1 Reality Mining

The Reality Mining experiment is one of the largest mobile phone projects attempted in academia. These are the data collected by MIT Media Lab at MIT [11]. They have captured communication, proximity, location, and activity information from 100 subjects at MIT over the course of the 2004-2005 academic year.

5.2.2 Haggle Project - IEEE Infocom Conference

The Haggle Infocomm dataset consists of social interactions among attendees at an IEEE Infocomm conference in the Grand Hyatt Miami [34]. There were two sets of participants. One consisted of 41 participants and the duration of the conference was 4 days. The other experiment had 264 participants. The time quantization period was 10 minutes.

	α, β_1, β_2	Optimum	ρ_2	PCC	IPCC+DP
Grevy's	1,1,3	$\geq$ 56	4	106 (1.89)	76 (1.39)
	1,1,2	$\geq$ 56	4	106 (1.89)	76 (1.36)
	1,1,1	$\geq$ 51	4	106 (2.08)	69 (1.35)
	2,1,1	$\geq$ 59	8	212 (3.59)	98 (1.66)
	3,1,1	$\geq$ 59	12	318 (5.39)	109 (1.85)
Hg-264	1,1,3	$\geq$ 2030	4	3347 (1.65)	2442 (1.20)
	1,1,2	$\geq$ 2030	4	3347 (1.65)	2442 (1.20)
	1,1,1	$\geq$ 1015	4	3347 (3.30)	2194 (2.16)
	2,1,1	$\geq$ 1015	8	6694 (6.60)	3111 (3.06)
	3,1,1	$\geq$ 1015	12	10041 (9.89)	3700 (3.65)
Hg-41	1,1,3	$\geq$ 1013.0	4	1547 (1.53)	1218 (1.20)
	1,1,2	$\geq$ 1013.0	4	1547 (1.53)	1218 (1.20)
	1,1,1	$\geq$ 506.5	4	1547 (3.05)	1158 (2.29)
	2,1,1	$\geq$ 506.5	8	3094 (6.11)	1700 (3.36)
	3,1,1	$\geq$ 506.5	12	4641 (9.16)	2101 (4.15)
Onagers	1,1,3	$\geq$ 125	4	282 (2.26)	192 (1.54)
	1,1,2	$\geq$ 125	4	282 (2.26)	192 (1.54)
	1,1,1	$\geq$ 125	4	282 (2.26)	175 (1.43)
	2,1,1	$\geq$ 121	8	564 (4.66)	255 (2.11)
	3,1,1	$\geq$ 121	12	846 (6.99)	298 (2.46)
Plains	1,1,3	$\geq$ 44925.0	4	85630 (1.91)	45785 (1.02)
	1,1,2	$\geq$ 44925.0	4	85630 (1.91)	45785 (1.02)
	1,1,1	$\geq$ 22462.5	4	85630 (3.81)	45785 (2.04)
	2,1,1	$\geq$ 22462.5	8	171260 (7.62)	55578 (2.47)
	3,1,1	$\geq$ 22462.5	12	256890 (11.44)	58928 (2.62)
Reality	1,1,3	$\geq$ 12498.0	4	21374 (1.71)	17066 (1.36)
	1,1,2	$\geq$ 12489.0	4	21374 (1.71)	17066 (1.36)
	1,1,1	$\geq$ 6244.5	4	21374 (3.42)	14943 (2.39)
	2,1,1	$\geq$ 6244.5	8	42748 (6.85)	19792 (3.17)
	3,1,1	$\geq$ 6244.5	12	64149 (10.27)	22147 (3.55)
SW	1,1,3	48	4	78 (1.62)	50 (1.04)
	1,1,2	48	4	78 (1.62)	50 (1.04)
	1,1,1	36	4	78 (2.17)	43 (1.19)
	2,1,1	41	8	156 (3.80)	50 (1.22)
	3,1,1	42	12	234 (5.57)	53 (1.26)

Table 2: Performance of PCC and IPCC with the Dynamic Programming.

5.3 Coloring Results

In this section, we show visualizations of the actual structure of the colorings of the Southern Women data set, produced by our algorithms under the cost setting $(\alpha, \beta_1, \beta_2) = (1, 1, 1)$. Figure 3 shows the colorings produced by three algorithms. Note, that the overall structure of dynamic communities remains very similar despite the difference in the costs of the colorings. Thus, qualitatively the algorithm infers communities that are close to the optimal.

6. CONCLUSIONS

We continue our line of research on identifying communities in dynamic networks started in [37]. We generalize the formulation of the Minimum Community Interpretation problem and present two approximation algorithms: the first approximates the special case with no missing data and the other approximates the general case. We analyze the performance guarantee of the algorithms. The proposed algorithms are the first rigorous computational solutions to the Minimum Community Interpretation problem in dynamic networks with provable performance guarantees. While the theoretical analysis guarantees a constant factor approximation, in practice the best implementation of our algorithm finds solutions very close to the optimum numerically, and even closer structurally. Moreover, both algorithms completed in less than 2 minutes on networks of several thousand individuals. Thus, our algorithms can be used in practice to infer communities in dynamic social networks.

7. REFERENCES

[1] J. Aizen, D. Huttenlocher, J. Kleinberg, and A. Novak. Traffic-based feedback on the web. *Proc. National Academy of Sciences*, 101(Suppl.1):5254–5260, 2004.

[2] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *Proc. 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2006.

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

[4] T. Y. Berger-Wolf and J. Saia. A framework for analysis of dynamic social networks. In *Proc. 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 523–528, New York, NY, USA, 2006. ACM Press.

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

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, 2001.

[7] D. P. Croft, R. James, P. Thomas, C. Hathaway, D. Mawdsley, K. Laland, and J. Krause. Social structure and co-operative interactions in a wild population of guppies (*Poecilia reticulata*). *Behavioural Ecology and Sociobiology*, In Press.

[8] P. C. Cross, J. O. Lloyd-Smith, and W. M. Getz. Disentangling association patterns in fission-fusion societies using African buffalo as an example. *Animal Behaviour*, 69:499–506, 2005.

[9] A. Davis, B. B. Gardner, and M. R. Gardner. *Deep South*. The University of Chicago Press, Chicago, IL, 1941.

[10] R. Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, August 2005.

[11] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *J. Personal and Ubiquitous Computing*, 2006.

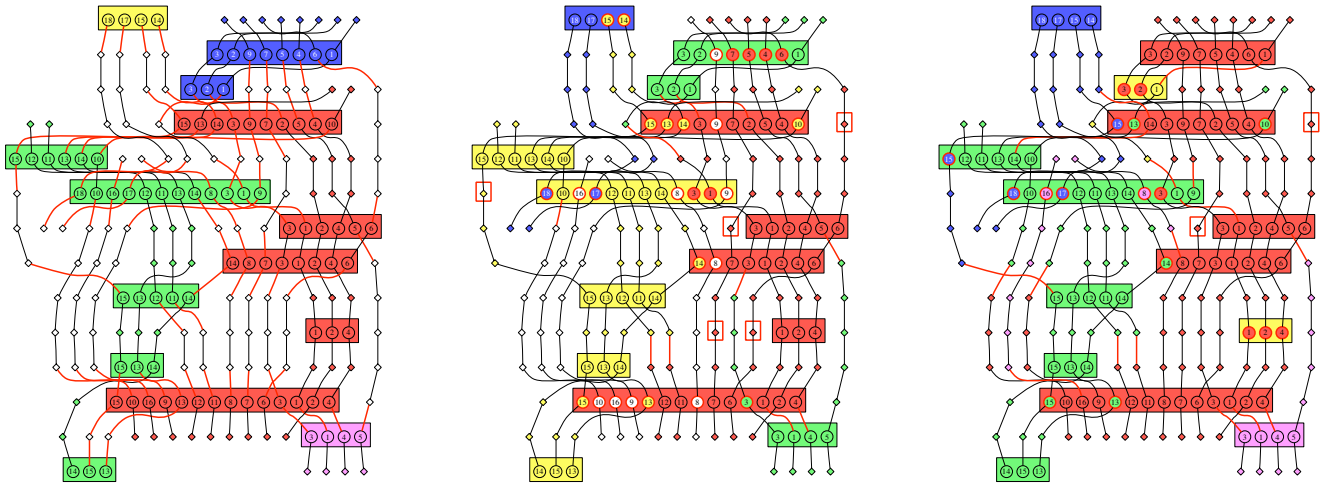
[12] I. R. Fischhoff, S. R. Sundaresan, J. Cordingley, H. M. Larkin, M.-J. Sellier, and D. I. Rubenstein. Social relationships and reproductive state influence leadership roles in movements of plains zebra (*Equus burchellii*). *Animal Behaviour*, 73: 825–831, 2007.

[13] I. R. Fischhoff, S. R. Sundaresan, J. Cordingley, and D. I. Rubenstein. Habitat use and movements of plains zebra (*Equus burchelli*) in response to predation danger from lions. *Behavioral Ecology*, 18(4): 725–729, 2007.

[14] G. Flake, S. Lawrence, C. L. Giles, and F. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3), 2002.

[15] D. Franzblau and A. Raychaudhuri. Optimal Hamiltonian completions and path covers for trees, and a reduction to maximum flow. *Anziam journal*, 44(2):193–204, 2002.

[16] L. Freeman. Finding social groups: A meta-analysis of the southern women data. In R. Breiger, K. Carley, and P. Pattison, editors, *Dynamic Social Network Modeling and Analysis*. The National Academies Press, Washington, D.C., 2003.



(a) the coloring of total cost 74 produced by Algorithm PCC alone

(b) the coloring of total cost 43 produced by IPCC with the Dynamic Programming

(c) the optimal coloring of total cost 36 produced by exhaustive search

Figure 3: Three colorings of the Southern Women data set with cost setting $\alpha = \beta_1 = \beta_2 = 1$. In each figure, each row represents a time step, time flows from top to bottom. The rectangles represent groups in the interaction sequence. Contained in each rectangle are circles representing the individual members of the group. If the circle is different color from the group then the individual is paying a visiting cost. Diamonds represent individuals who are unobserved at the time. Red empty rectangles surrounding a diamond indicate an absent cost being paid. Lines connect an individual to itself. Red lines correspond to a switching cost being paid.

[17] L. C. Freeman. On the sociological concept of "group": a empirical test of two models. *American Journal of Sociology*, 98:152–166, 1993.

[18] L. Getoor and C. Diehl. Link mining: A survey. *SIGKDD Explorations Special Issue on Link Mining*, 7(2), December 2005.

[19] D. Gibson, J. M. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *UK Conf. on Hypertext*, pages 225–234, 1998.

[20] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99:8271–8276, 2002.

[21] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proc. 13th Intl. Conf. on World Wide Web*, pages 491–501, New York, NY, USA, 2004. ACM Press.

[22] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 541–546, 2003.

[23] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2003.

[24] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

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

[26] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proc. 8th Intl. World Wide Web Conf.*, May 1999.

[27] M. Magdon-Ismail, M. Goldberg, W. Wallace, and D. Siebecker. Locating hidden groups in communication networks using hidden markov models. In *Proc. Intl. Conf. on Intelligence and Security Informatics (ISI 2003)*, Tucson, AZ, 2003.

[28] B. Malin. Data and collocation surveillance through location access patterns. In *Proc. North American Association for Computational Social and Organizational Science Conf.*, Pittsburgh, PA, June 2004.

[29] L. A. Meyers, M. Newman, and B. Pourbohloul. Predicting epidemics on directed contact networks. *Journal of Theoretical Biology*, 240:400–418, 2006.

[30] M. Newman, A.-L. Barabási, and D. J. Watts, editors. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[31] J. M. Read and M. J. Keeling. Disease evolution on networks: the role of contact structure. *Proc. R. Soc. Lond. B*, 270:699–708, 2003.

[32] E. M. Rogers. *Diffusion of Innovations*. Simon & Shuster, Inc., 5th edition, 2003.

[33] D. I. Rubenstein, S. Sundaresan, I. Fischhoff, and D. Saltz. Social networks in wild asses: Comparing patterns and processes among populations. In A. Stubbe, P. Kaczensky, R. Samjaa, K. Wesche, and M. Stubbe, editors, *Exploration into the Biological Resources of Mongolia*, volume 10. Martin-Luther-University Halle-Wittenberg, 2007. In press.

[34] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/ haggle/ imote/ infocom (v. 2006-01-31). Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom.

[35] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proc. 13th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 687–696, New York, NY, USA, 2007. ACM.

[36] S. R. Sundaresan, I. R. Fischhoff, J. Dushoff, and D. I. Rubenstein. Network metrics reveal differences in social organization between two fission-fusion species, Grevy's zebra and onager. *Oecologia*, 2006.

[37] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *ope: parameter-free mining of large time-evolving graphs.* In *Proc. 13th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 717–726, New York, NY, USA, 2007. ACM.

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