Full Diffusion History Reconstruction in Networks

Zhen Chen*, Hanghang Tong† and Lei Ying‡
*‡School of Electrical, Computer and Energy Engineering
†School of Computing, Informatics and Decision Systems Engineering
Arizona State University
Tempe, Arizona, 85281
Email: {*zhen.chen.1, †hanghang.tong, ‡lei.ying.2}@asu.edu

***Abstract*—Diffusion processes in networks can be used to model many real-world processes. Analysis of diffusion traces can help us answer important questions such as the source of diffusion and the role of each node in the diffusion process. However, in large-scale networks, it is very expensive if not impossible to monitor the entire network to collect the complete diffusion trace. This paper considers diffusion history reconstruction from a partial observation and develops a greedy, step-by-step reconstruction algorithm. It is proved that the algorithm always produces a diffusion history that is consistent with the partial observation. Our experimental results based on real networks and real diffusion data show that the algorithm significantly outperforms some existing methods.**

I. Introduction

Diffusion processes in networks can be used to model many real-world phenomena including the spread of an infectious disease, the propagation of a computer virus, the gradual adoption of a new product, etc. Loosely speaking, the research on diffusion processes in networks can be categorized into two groups: prospective analysis which focuses on the structural properties of diffusion processes and networks that lead to epidemic-type outbreaks and algorithms to minimize or maximize network diffusion, and retrospective analysis which focuses on network inference such as identifying the source or underlying network of diffusion.

In this paper, we go beyond identifying the source of diffusion and study the problem of reconstructing the entire history of a diffusion process, named as *diffusion history reconstruction*, which has been studied only very recently [1]. We assume that the diffusion process follows the Susceptible-Infected (SI) model, a variant of the popular SIR model first proposed in [2], and a *single* snapshot of the network is given, which includes the set of "infected" nodes, and the corresponding infection time. The nodes with known "infection" time can be thought as monitor nodes that were placed in the network. Each monitor node can record the time at which the node is "infected" and report the infection time. The main contributions of this paper are summarized below.

- We formulate the diffusion history reconstruction problem as a maximum a posteriori (MAP) estimate problem, and prove that the problem is NP-hard by reducing an arbitrary set cover problem to a diffusion history reconstruction problem.
- We propose a greedy and step-by-step reconstruction algorithm to reconstruct the most likely network state at time slot τ based on the network state at time slot $\tau-1$ while guaranteeing the state is consistent with partial observation, and further develop a greedy algorithm for a single-step construction. The key idea of the single-step construction algorithm is to convert the problem to the weighted set cover problem, for which a well-known greedy algorithm provides a guarantee on the approximation ratio.
- We prove that the diffusion history obtained by the step-by-step reconstruction algorithm is always consistent with the partial observation, and the computational complexity of the algorithm is $O(V_I^3)$, where V_I is the number of infected nodes observed in the snapshot.
- We evaluate the performance of the algorithm on the Western States Power Grid of the United States [3] and Internet autonomous systems (IAS) network [4], with simulated diffusion processes following the SI model. We also test our algorithm on the Weibo dataset (Weibo is a famous Chinese microblogging website). In all scenarios, we observe significant improvements of the proposed algorithm compared with other heuristic and existing algorithms.

A. Related Work

In this section, we review the related work in diffusion process on networks, which can be categorized into two parts: prospective analysis and retrospective analysis.

Prospective Analysis. Many research works in diffusion process [5]–[10] have been devoted to studying the so-called epidemic threshold, that is, to determine the condition under which an epidemic will break out. While earlier works [11] focus on some specific types of graph structure (e.g., random graphs, power-law graphs, etc), Wang et al. [12] and its follow-up paper by Ganesh et al. [13] found that, for the flu-like SIS model, the epidemic threshold for any *arbitrary, real* graph is determined by the leading eigenvalue of the adjacency matrix of the graph. Prakash et. al. [14] further discovered that the leading eigenvalue (and a model-dependent constant) is the only parameter that determines the epidemic threshold for other virus propagation models. On the algorithmic side, Hayashi et al. [15] derived the extinction conditions under random and targeted immunization for the SHIR model (Susceptible, Hidden, Infectious, Recovered). Tong et al. [16] proposed an effective node immunization strategy for the SIS model by approximately minimizing the leading eigenvalue. Briesemeister et al. [17] studied the defending policy in power-law graphs. Prakash et. al. [18], [19] proposed effective algorithms to perform node immunization on time-varying graphs.

978-1-4799-9926-2/15/$31.00 ©2015 IEEE

Retrospective Analysis. Earlier work along this line focuses on identifying the source of diffusion [20]–[23] and inferring the underlying network of diffusion [24]–[26]. An even more challenging problem is to reconstruct the diffusion history, which has been received sparse attention so far. Paper [1] tried to reconstruct the history by using multiple snapshots of the network at different time under the discrete time SEIRS model and it proposed an algorithm based on submodularity with some provable performance guarantee. The information used to reconstruct the diffusion history in [1] is multiple snapshots of the whole network at different time slots, which can be considered as a time domain partial information. In contrast, in this paper, we use a single snapshot with the infection time of partial infected nodes, which is a space domain partial information. A method of finding the most possible diffusion path by using the infection time information of all infected nodes is proposed in [27], in which the authors considered the diffusion path as a tree. In [28], the authors tried to estimate the diffusion path and infection time of some nodes by using the infection time of partial infected nodes. A heuristic algorithm was proposed in [28] based on the integer programming problem formulated by the authors. Besides the high complexity, the heuristic algorithm in [28] involves iterations between finding the infection path and estimating infection time, while the convergence is not guaranteed. In [29], the authors focused on inferring the diffusion path of the diffusion process based on the independent cascade model by using partial observations. A heuristic algorithm derived from minimum Steiner tree was proposed in [29]. Compared with independent cascade model, the Susceptible-Infected model used in this paper goes beyond the assumption that each infected node only has one chance to infect its neighbors.

II. Problem Formulation

In this section, we define the diffusion history reconstruction problem. We assume the diffusion process starting from a single source in the network denoted by $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges. Node u can infect node v if there is an edge $u \to v$. Node u is called an incoming neighbor of node v, and node v is called an outgoing neighbor of node u. We further assume the diffusion process starts from time slot 0, and a snapshot of the network is taken at time slot T. The snapshot includes the state of every node in the network at time T, as well as the infection time of a subset of infected nodes. The goal is to infer the complete diffusion history from the partial observation.

In this paper, we use the capital letter for constants (e.g, T), calligraphic fonts for sets (e.g., $\mathcal{E}$), and bold upper-case for matrices or vectors (e.g., $\mathbf{X}$). We use $\mathbf{X}_t$ to represent the t^{th} column of matrix $\mathbf{X}$. For each set, we use its associated capital letter for the cardinality of the set (e.g., $V = |\mathcal{V}|$). A graph is defined by its node set and edge set, e.g., a graph G with node set $\mathcal{V}$ and edge set $\mathcal{E}$ can be written as $G(\mathcal{V}, \mathcal{E})$. We use tilde to represent the matrix, set or vector to be reconstructed (e.g., $\tilde{\mathbf{X}}$). The key notation used throughout the paper is summarized in Table I.

TABLE I: Notation table

Symbol	Definition & Description
$G(\mathcal{V}, \mathcal{E})$	a graph G with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$.
T	the time when the snapshot of the network is taken
$\mathbf{I}$	infected nodes vector, which describes the infected nodes at time T
$\mathbf{T}$	infection timing vector, which saves the observed infection time of nodes
$\mathbf{X}$	the diffusion history matrix
$\tilde{\mathbf{X}}$	the reconstructed diffusion history matrix
$\mathbf{X}_t$ $(0 \le t \le T)$	the network state vector at time t
$\tilde{\mathbf{X}}_t$ $(0 \le t \le T)$	the reconstructed network state vector at time t
$X_{v,t}$	a binary variable, which is equal to 1 when node v is in the infected state at time t, otherwise, it is 0
p_{uv}	infection probability of edge (u, v)
$\tilde{\mathcal{S}}_t$ $(0 \le t \le T)$	the set of susceptible nodes on $G(\mathcal{V}, \mathcal{E})$ which has infected incoming neighbors according to $\mathbf{X}_t$
$\tilde{\mathcal{I}}_n$	the set of infected nodes in $\mathbf{X}_n$
$\mathcal{I}$	the set of infected nodes in the snapshot at time T
$\mathcal{N}_v$	the set of incoming neighbors of node v in network $G(\mathcal{V}, \mathcal{E})$

A. Diffusion model

We assume the diffusion process follows the discrete-time Susceptible-Infected (SI) model with a single source. Each node in the network has two states: susceptible (S) and infected (I). In each time slot, every susceptible node, v, can be infected by each of its infected incoming neighbors, u, with probability p_{uv}. Once the susceptible node gets infected, it will stay at the infected state forever. The diffusion starts at time $t = 0$ from the source of the diffusion.

B. Problem statement

Given a network $G(\mathcal{V}, \mathcal{E})$, we assume the observation we have for reconstructing the diffusion history is a pair $(\mathbf{T}, \mathbf{I})$, such that $\mathbf{I}$, named *infected nodes vector*, is a V-dimensional vector such that $I_v = 1$ if node v is infected and $I_v = 0$ otherwise; and $\mathbf{T}$, named *infection timing vector*, is also a V- dimensional vector such that T_v is infection time of node v if node v's infection time is observed and $T_v = -1$ otherwise. Let $\mathcal{I}$ denote the set of infected nodes. Define a $V \times T$ matrix $\mathbf{X}$ to be the diffusion history such that

$$X_{v,t} = \begin{cases} 1 & \text{Node } v \text{ is in the infected state at time } t, \\ 0 & \text{Node } v \text{ is susceptible at time } t. \end{cases} \quad (1)$$

Note that $\mathbf{X}$ defines the entire history of the diffusion process and under the SI model, $X_{v,t} = 1$ if $X_{v,\tau} = 1$ for $\tau < t$.

We can use the column vector $\mathbf{X}_t$ of diffusion history $\mathbf{X}$ to represent the network state at time t. Then the diffusion history matrix $\mathbf{X}$ can be written as $\mathbf{X} = (\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_T)$. The *diffusion history reconstruction* problem can be defined as follows:

Diffusion History Reconstruction Problem

Input: The underlying network for the diffusion, $G(\mathcal{V}, \mathcal{E})$, the time when the snapshot is taken, T, the infected nodes vector $\mathbf{I}$ and the infection timing vector $\mathbf{T}$.

Output: A diffusion history $\tilde{\mathbf{X}}$ such that

$$\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}} \Pr(\mathbf{X}|\mathbf{I}, \mathbf{T}). \quad (2)$$

$\diamond$

In order to find the $\tilde{\mathbf{X}}$, the exclusive search needs to calculate $\Pr(\mathbf{X}|\mathbf{I},\mathbf{T})$ for all possible diffusion history $\mathbf{X}$, which increases exponentially as the number of infected nodes increases. The problem is NP-hard and the proof is presented in the appendix.

Theorem 1: The diffusion history reconstruction problem defined in (2) is NP-hard.

Proof: See the appendix. □

III. A Step-by-Step Reconstruction Algorithm

Since the diffusion history reconstruction problem defined in (2) is difficult to solve, we propose a heuristic algorithm with polynomial complexity in this section. According to Bayes' theorem, we have

$$\begin{aligned}\Pr(\mathbf{X}|\mathbf{I},\mathbf{T}) &= \frac{\Pr(\mathbf{X},\mathbf{I},\mathbf{T})}{\Pr(\mathbf{I},\mathbf{T})}\\ &\propto \Pr(\mathbf{X},\mathbf{I},\mathbf{T})\\ &= \Pr(\mathbf{X})\Pr(\mathbf{I},\mathbf{T}|\mathbf{X})\end{aligned} \tag{3}$$

When the diffusion history $\mathbf{X}$ is known, the infection time of nodes and the set of infected nodes are fixed. Thus, in (3), the value of $\Pr(\mathbf{I},\mathbf{T}|\mathbf{X})$ is either 0 or 1 :

$$\Pr(\mathbf{I},\mathbf{T}|\mathbf{X}) = \begin{cases}1 & \mathbf{X} \text{ is consistent with } (\mathbf{I},\mathbf{T}),\\ 0 & \mathbf{X} \text{ is inconsistent with } (\mathbf{I},\mathbf{T}).\end{cases} \tag{4}$$

where we say that the diffusion history $\mathbf{X}$ is **consistent** with observation $(\mathbf{I},\mathbf{T})$ if the following two conditions hold:

- **H1** $X_{v,T} = 1$ when node v is an infected node according to $\mathbf{I}$ and $X_{v,T} = 0$ otherwise, and
- **H2** $X_{v,t} = 1$ for $t \geq T_v$ if $T_v \neq -1$.

We further define the network state at time τ ($\mathbf{X}_\tau$) to be **consistent** with $(\mathbf{I},\mathbf{T})$ if the following conditions hold:

- **S1** If $I_v = 0$, then $X_{v,\tau} = 0$. In other words, node v should be in the susceptible state if it is in the susceptible state at time T.
- **S2** If $I_v = 1$ and $0 \leq T_v \leq \tau$, then $X_{v,\tau} = 1$. In other words, node v should be in the infected state at time τ if it was infected at or before time slot τ.
- **S3** If $I_v = 1$ and $T_v > \tau$, then $X_{v,\tau} = 0$, and one of the following two conditions must hold
  - **c1** There exists node u with $T_u > \tau$ such that $d(u,v) \leq T_v - T_u$.
  - **c2** There exists node u with $X_{u,\tau} = 1$ and $d(u,v) \leq T_v - \tau$.

  Here $d(u,v)$ is defined to be the length of the shortest *Infection-Time-Free path* (or ITF-path) between node u and node v, where an ITF-path is a path that includes only infected nodes such that $X_{w,\tau} = 0$ except the two end nodes. The condition (c1) means node u, who was infected at time slot T_u, can infect node v via an ITF-path at T_v. The condition (c2) means node u, who has already been infected at time slot τ, can infect node v via an ITF-path at T_v.
- **S4** If $I_v = 1$ and $T_v = -1$, then either $X_{v,\tau} = 1$, or $X_{v,\tau} = 0$ and one of the following two conditions must hold
  - **c1** There exists node u with $T_u > \tau$ such that $d(u,v) \leq T - T_u$.
  - **c2** There exists node u with $X_{u,\tau} = 1$ and $d(u,v) \leq T-\tau$.

  Here the condition (c1) means node u, who was infected at time slot T_u, can infected node v via an ITF-path before or at time T. The condition (c2) means node u, who has already been infected at time slot τ, can infect node v via an ITF-path before or at time T.

According to the discussion above, the problem of

$$\max_{\mathbf{X}} \Pr(\mathbf{X}|\mathbf{I},\mathbf{T})$$

is equivalent to

$$\begin{aligned}&\max_{\mathbf{X}} \Pr(\mathbf{X})\\ &\text{subject to: } \mathbf{X} \text{ is consistent with } (\mathbf{I},\mathbf{T}).\end{aligned} \tag{5}$$

Since $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_T$ form a Markov chain under the SI model, we have

$$\begin{aligned}\Pr(\mathbf{X}) &= \Pr(\mathbf{X}_0,\mathbf{X}_1,\ldots,\mathbf{X}_T)\\ &= \Pr(\mathbf{X}_0)\Pr(\mathbf{X}_1|\mathbf{X}_0)\ldots\Pr(\mathbf{X}_T|\mathbf{X}_{T-1})\\ &= \Pr(\mathbf{X}_0)\prod_{\tau=1}^{T}\Pr(\mathbf{X}_\tau|\mathbf{X}_{\tau-1}).\end{aligned}$$

Now to solve (5), our greedy approach is to recursively solve the following *single-step reconstruction* problem with a given $\mathbf{X}_{\tau-1}$

$$\begin{aligned}&\max_{\mathbf{X}_\tau} \Pr(\mathbf{X}_\tau|\mathbf{X}_{\tau-1})\\ &\text{subject to: } \mathbf{X}_\tau \text{ is consistent with } (\mathbf{I},\mathbf{T}).\end{aligned} \tag{6}$$

The first step of this greedy algorithm needs input $\mathbf{X}_0$, i.e., the identify of the source. Define $\mathbf{1}^{(v)}$ to be a V-dimensional vector such that $\mathbf{1}^{(v)}_v = 1$ and $\mathbf{1}^{(v)}_m = 0$ for $m \neq v$. The algorithm will set $\tilde{\mathbf{X}}^{(v)}_0 = \mathbf{1}^{(v)}$ for each v that is the possible source and then calculates $\tilde{\mathbf{X}}^{(v)}_\tau$ by recursively solving (6) (again with a greedy algorithm which will be presented in the next subsection). Then the diffusion history is set to be the most likely $\tilde{\mathbf{X}}^{(v)}$. The pseudo code is presented in Algorithm 1.

Algorithm 1: The Step-by-Step Reconstruction Algorithm (SSR)

```
for each possible source, i.e., v ∈ V_s do
    Set X̃_0^(v) = 1^(v);
    for 1 ≤ τ ≤ T do
        Set X̃_τ^(v) to be solution of problem (6) with
        X_{τ-1} = X̃_{τ-1}^(v).
    end
    Set γ_v = ∏_{1≤τ≤T} Pr( X̃_τ^(v) | X̃_{τ-1}^(v) ).
end
Set v* ∈ arg max_{v∈V_s} γ_v.
return X̃^(v*)
```

Note that the set of possible sources, $\mathcal{V}_s$, can be determined by the observation $(\mathbf{I},\mathbf{T})$. For $v \in \mathcal{V}$ with $T_v \neq -1$ and

$I_v = 1$, we know that the distance from the source to v must be $\leq T_v$; for $v \in \mathcal{V}$ with $T_v = -1$ and $I_v = 1$, the distance from the source to v must be $\leq T$. Therefore, the set of infected nodes satisfied previous conditions can form the set $\mathcal{V}_s$.

A. Single-Step Reconstruction

We now focus on solving (6) when $\tilde{\mathbf{X}}_{\tau-1}$ is given. In the following discussion, we assume $\tilde{\mathbf{X}}_{\tau-1}$ is given. Note that node v can become an infected node at time τ if $I_v = 1$, $\tilde{X}_{v,\tau-1} = 0$ and it has at least an infected incoming neighbor at time $\tau-1$. Denote by $\tilde{\mathcal{S}}_{\tau-1}$ the set of susceptible nodes with infected incoming neighbors according to $\tilde{\mathbf{X}}_{\tau-1}$, $\mathcal{N}_v$ the set of incoming neighbors of node v, and $\tilde{\mathcal{I}}_{\tau-1}$ the set of infected nodes at time $\tau - 1$. For each node $v \in \tilde{\mathcal{S}}_{\tau-1}$, the probability that v is not infected by its incoming infected neighbors at time τ is

$$\prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv}).$$

Furthermore, the objective in (6) can be written as

$$\begin{aligned} \Pr(\mathbf{X}_\tau | \tilde{\mathbf{X}}_{\tau-1}) = \prod_{v \in \tilde{\mathcal{S}}_{\tau-1}} & \underbrace{\Big(1 - \prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})\Big)^{X_{v,\tau}}}_{(a)} \\ & \underbrace{\Big(\prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})\Big)^{1 - X_{v,\tau}}}_{(b)}, \end{aligned} \tag{7}$$

where term (a) is the probability that v gets infected at time τ while term (b) represents the probability that v stays susceptible at time τ. The log-likelihood can be written as

$$\begin{aligned} & \log \Pr(\mathbf{X}_\tau | \tilde{\mathbf{X}}_{\tau-1}) \\ = & \sum_{v \in \tilde{\mathcal{S}}_{\tau-1}} \Big(X_{v,\tau} \log \big(1 - \prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})\big) \\ & \quad + (1 - X_{v,\tau}) \log \prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv}) \Big) \\ = & \sum_{v \in \tilde{\mathcal{S}}_{\tau-1}} X_{v,\tau} \log \frac{1 - \prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})}{\prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})} \\ & \quad + \log \prod_{u \in \mathcal{N}_v \cap \tilde{I}_{\tau-1}} (1 - p_{uv}) \\ = & \sum_{v \in \tilde{\mathcal{S}}_{\tau-1}} \alpha_v^\tau X_{v,\tau} + \beta_v^\tau, \end{aligned} \tag{8}$$

where

$$\alpha_v^\tau = \log \frac{1 - \prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})}{\prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})} \tag{9}$$

and

$$\beta_v^\tau = \log \prod_{u \in \mathcal{N}_v \cap \tilde{\mathcal{I}}_{\tau-1}} (1 - p_{uv})$$

are two constants whose values depend only on $\tilde{\mathbf{X}}_{\tau-1}$. Therefore, optimization problem described in (6) can be written as

$$\begin{aligned} \max_{\mathbf{X}_\tau} \quad & \sum_{v \in \tilde{\mathcal{S}}_{\tau-1}} \alpha_v^\tau X_{v,\tau} + \beta_v^\tau \\ \text{subject to} \quad & \mathbf{X}_\tau \text{ is consistent with } \mathbf{I} \text{ and } \mathbf{T}. \end{aligned} \tag{10}$$

Note that only nodes in $\tilde{\mathcal{S}}_{\tau-1} \cap \mathcal{I}$ can change from susceptible to infected at time slot τ. The problem is combinatoric in nature since $X_{v,\tau} \in \{0, 1\}$. Next we reduce the problem above to a weighted set cover problem. Note that the consistency conditions are to guarantee all infected nodes can be infected at the observed infection time or by the time at which the snapshot was taken. It is not difficult to see that we only need to check the consistency of nodes in $\mathcal{I} \setminus \tilde{\mathcal{I}}_{\tau-1}$ for $\mathbf{X}_\tau$ because nodes in $\tilde{\mathcal{I}}_{\tau-1}$ have been successfully infected by time $\tau - 1$ and nodes in $\mathcal{V} \setminus \mathcal{I}$ should always stay as susceptible. Problem (10) can be converted to a weighted set cover problem with the following steps:

(1) Define the *universe* to be

$$\mathcal{U} = \mathcal{I} \setminus \tilde{\mathcal{I}}_{\tau-1},$$

i.e., the set of nodes whose consistency needs to be verified in $\mathbf{X}_\tau$.

(2) Set $\mathcal{S} = \tilde{\mathcal{S}}_{\tau-1} \cap \mathcal{I}$. For each node in $\mathcal{S}$, say node u, initiate a set $\mathcal{S}_u = \emptyset$.

(3) For each node in the universe (say node v), construct a modified-breadth-first-search (MBFS) tree as follows: Reverse all edges of the infected subgraph. On the reversed graph, starting from the node, we conduct the breadth-first search (BFS). When BFS hits an infected node in $\tilde{\mathcal{I}}_{\tau-1}$ or with observed infection time, BFS at this node stops, i.e., the node becomes a leaf-node of the MBFS tree. Note that a path from the root to any leaf-node of the MBFS-tree is an ITF-path.

- If one of leaf-nodes on the MBFS-tree, which is not in $\mathcal{S}$, satisfies the consistency condition **S3** or **S4** for root node v, then node v is removed from the universe, i.e.,

$$\mathcal{U} = \mathcal{U} \setminus \{v\}$$

because node v is consistent in $\mathbf{X}_\tau$ regardless of the states of nodes in $\mathcal{S}$.

- Otherwise, for each node (say node u) in $\mathcal{S}$ with $T_u > \tau$, remove u from $\mathcal{S}$, i.e.,

$$\mathcal{S} = \mathcal{S} \setminus \{u\}.$$

For each node u in $\mathcal{S}$:

- If $T_v \neq -1$, check whether the depth of node u on the MBFS-tree is $\leq T_v - \tau$. If it is the case, node v is added to $\mathcal{S}_u$, i.e.,

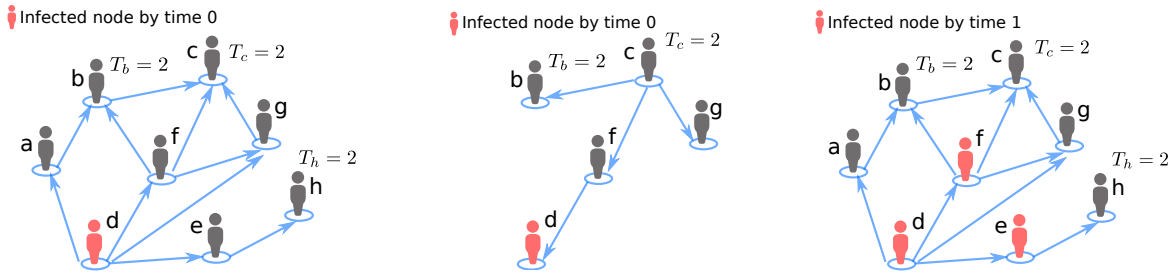
$$\mathcal{S}_u = \mathcal{S}_u \cup \{v\}.$$

- If $T_v = -1$, check whether the depth of node u on the MBFS-tree is $\leq T - \tau$. If it is the case, node v is added to $\mathcal{S}_u$.

(4) According to (9), we calculate the weight of set $\mathcal{S}_u$: $w_u = -\alpha_u^\tau$ for $u \in \mathcal{S}$. For each $u \in \mathcal{S}$ with $w_u < 0$, set $X_{u,\tau} = 1$, change the universe $\mathcal{U}$ to $\mathcal{U} \setminus \mathcal{S}_u$, and remove u from $\mathcal{S}$.

Note that $v \in \mathcal{S}_u$ implies that node v's consistency is guaranteed if node u becomes infected at time slot τ. The problem (10), therefore, is equivalent to identifying a set of $\mathcal{S}_u$ ($u \in \mathcal{S}$) with the smallest summation of weights to cover the universe $\mathcal{U}$.

Consider a simple network in Figure 1a. Assume the infection probability of any edge is 0.3, the snapshot is taken



(a) The network when the time slot is 1.

(b) The MBFS-tree when node c is the root.

(c) The reconstructed network state for time 1.

Fig. 1: The example of diffusion network state single-step reconstruction.

at $T = 4$, and all nodes are in the infected state at time 4. Furthermore, assume the infection time of nodes b, c and h is known ($T_b = 2$, $T_c = 2$ and $T_h = 2$). We next outline the key steps to solve (10) by starting from node d, i.e., assuming $\tilde{\mathbf{X}}_0 = [0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^{tr}$ where tr means the transpose. So

$$\tilde{\mathcal{I}}_0 = \{d\}$$

and

$$\mathcal{U} = \mathcal{I} \setminus \tilde{\mathcal{I}}_0 = \{a, b, c, e, f, g, h\}.$$

Furthermore,

$$\tilde{\mathcal{S}}_0 = \{a, f, g, e\}.$$

Now consider the MBFS rooted at node c. The algorithm first explores the outgoing neighbors of c, which are nodes b, f and g. The edges (c, b), (c, f) and (c, g) are added to the MBFS tree. Since b's infection time is known, the outgoing neighbors of b should not be explored in the next step. In the next step, the MBFS checks the outgoing neighbors of f and g. Since node d is an outgoing neighbor of node f, edge (f, d) is added to the MBFS-tree. And the outgoing neighbors of node g, which are f and d, are already explored by the MBFS, the MBFS at g stops. Since node d does not have any outgoing neighbors, the MBFS stops and the MBFS-tree at root c is in Figure 1b.

On the MBFS-tree in Figure 1b, nodes b and d do not satisfy **S3.c1** or **S3.c2** for node c, so node c cannot be removed from the universe. We then check the ITF-paths to nodes f and g on the MBFS-tree, and both ITF-paths have a length $\leq T_c - 1$. So node c is added to $\mathcal{S}_f$ and $\mathcal{S}_g$. After doing a similar procedure for other MBFS-trees, we have $\mathcal{U} = \{b, c, f, g, h\}$, $\mathcal{S}_a = \{b\}$, $\mathcal{S}_e = \{e, h\}$, $\mathcal{S}_f = \{c, b\}$, and $\mathcal{S}_g = \{c\}$. According to (9), the weights are $w_a = -\alpha_a^1 = 0.847$, $w_f = -\alpha_f^1 = 0.847$, $w_g = -\alpha_g^1 = 0.847$, and $w_e = -\alpha_e^1 = 0.847$. For this example, the solution to the weighted set cover problem is easy to find, which are $\mathcal{S}_f$ and $\mathcal{S}_e$, i.e., nodes f and e should become infected at time slot 1. The reconstructed network state at time slot 1 is shown in Figure 1c.

In general, the weighted set cover problem is NP-hard. However, we can use the greedy set cover algorithm in [30] to find a feasible solution with performance guarantee.

In the discussion above, we need the MBFS-tree of every remaining-infected-node in $\tilde{\mathbf{X}}_{\tau-1}$, while the MBFS-tree depends on the state of $\tilde{\mathbf{X}}_{\tau-1}$. Running the MBFS at every single step is very time-consuming. Instead, we run the MBFS starting from every infected-node at the beginning of the algorithm and save the MBFS-trees. Then in each iteration, when a remaining-infected-node is selected to be an already-infected-node (say node u is selected), we prune the subtree starting from node u from all MBFS-trees but keep node u. The reason the subtree starting from node u can be pruned is because for any node on the subtree, say node y, we have $d(v, y) > d(v, u)$, where v is the root of the MBFS-tree, so node v can be infected by node y via an ITF-path after τ only if it can be infected by node u via an ITF-path.

An single-step reconstruction algorithm based on weighted set cover is stated in Algorithm 2, which is formed by four steps:

1) Prune each MBFS-tree rooted at a susceptible node v in previous network state with $I_v = 1$;
2) Convert the problem to a weighted set cover problem by following the procedure in Page 4;
3) Solve the weighted set cover problem by using greedy set cover algorithm;
4) Obtain the network state according to the result of greedy set cover algorithm and calculate the objective in (10).

As described in Algorithm 1, by using Algorithm 2 recursively, we can get a diffusion history, which is consistent with the observation.

Define $G_I(\mathcal{V}_I, \mathcal{E}_I)$ to be the infected subgraph of $G(\mathcal{V}, \mathcal{E})$, which is formed by the infected nodes observed at T. Then we have $V_I = |\mathcal{V}_I|$. The next theorem summarizes the theoretical guarantees of SSR. The proof can be found in the appendix. Notice that the time complexity only depends on the number of infected nodes V_I, which is often much smaller than the size of the underlying network V.

Theorem 2: Algorithm 1 produces a diffusion history consistent with the observation with worst-case computational complexity of $O\left(V_I^3\right)$.

Proof: See the appendix. □

IV. Performance Evaluation

In this section, we compare the performance of SSR with other heuristics using following three performance measures.

Algorithm 2: Single-Step Reconstruction

Input : Network $G(\mathcal{V}, \mathcal{E})$, the previous reconstructed network state $\tilde{\mathbf{X}}_{\tau-1}$, the observed information $(\mathbf{I}, \mathbf{T})$, current time τ and the MBFS-tree $\mathbb{T}_v$ $(v \in \mathcal{I} \setminus \tilde{\mathcal{I}}_{\tau-1})$ rooted at node v from previous step.

Output: The network state $\tilde{\mathbf{X}}_\tau$, the value of objective in (10), o, and the MBFS-trees after pruning.

$o \leftarrow 0$;
$\mathcal{U} \leftarrow \mathcal{I} \setminus \tilde{\mathcal{I}}_{\tau-1}$;
$\tilde{\mathbf{X}}_\tau \leftarrow \tilde{\mathbf{X}}_{\tau-1}$;
$\mathcal{S} \leftarrow \tilde{\mathcal{S}}_{\tau-1} \cap \mathcal{I}$;
let $\mathcal{S}_v \leftarrow \emptyset$ for each $v \in \mathcal{S}$;
$\mathcal{M} \leftarrow \{v | v \in \mathcal{I}, T_v \neq -1\}$;
for $v \in \mathcal{I} \setminus \tilde{\mathcal{I}}_{\tau-1}$ **do**
  let $t_v \leftarrow T_v$ if $v \in \mathcal{M}$. Otherwise, $t_v \leftarrow T$;
  for $u \in \{$*the nodes on* $\mathbb{T}_v\}$ **do**
    if $u \in \tilde{\mathcal{I}}_{\tau-1}$ **then** remove the subtree rooted at u on $\mathbb{T}_v$ except u;
  end
  if *there exists a node* $u \in \{$*the nodes on* $\mathbb{T}_v\} \cap (\mathcal{M} \setminus \tilde{\mathcal{I}}_{\tau-1})$ *such that the depth of* u *on* $\mathbb{T}_v$ *is* $\leq t_v - T_u$ *and* $T_u > \tau$ **then** remove v from $\mathcal{U}$;
  else if *there exists a node* $u \in \{$*the nodes on* $\mathbb{T}_v\} \cap \tilde{\mathcal{I}}_{\tau-1}$ *such that the depth of* u *on* $\mathbb{T}_v$ *is* $\leq t_v - \tau$ **then** remove v from $\mathcal{U}$;
  else
    for $u \in \mathcal{S}$ **do**
      if $u \in \mathcal{M}$ *and* $T_u > \tau$ **then continue**;
      else if $u \in \{$*the nodes on* $\mathbb{T}_v\}$ *and the depth of* u *on tree* $\mathbb{T}_v$ *is* $\leq t_v - \tau$ **then**
        $\mathcal{S}_u \leftarrow \mathcal{S}_u \cup \{v\}$;
      end
    end
  end
end
remove any node $v \in \mathcal{S}$ with $T_v > \tau$;
if *the union of* $\mathcal{S}_v$ *for any* $v \in \mathcal{S}$ *is not equal to* $\mathcal{U}$ **then** $o \leftarrow -\infty$;
else
  $w_v \leftarrow -\alpha_v^\tau$ for any $v \in \mathcal{S}$;
  for $v \in \mathcal{S}$ **do**
    if $w_v < 0$ *or* $T_v = \tau$ **then**
      $X_{v,\tau} \leftarrow 1$;
      $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{S}_v$;
      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{v\}$;
    end
  end
  let $\mathcal{R}$ to be the result of the greedy set cover algorithm [30] on the subset $\mathcal{S}_v$ for any $v \in \mathcal{S}$ and the universe $\mathcal{U}$;
end
if $o \neq -\infty$ **then**
  set $X_{v,\tau} \leftarrow 1$ for any $v \in \mathcal{S}$ with $\mathcal{S}_v \in \mathcal{R}$;
  calculate the objective value in (10), o;
end
return $\tilde{\mathbf{X}}_\tau$, o and $\mathbb{T}_v$ for $v \in \mathcal{V}$;

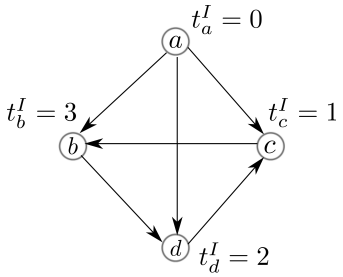


Fig. 2: An example of diffusion. For $v \in \{a, b, c, d\}$, t_v^I is the infection time of node v. In this example, a is the source.

- **Kendall's τ_b coefficient:** Since the diffusion history includes the infection time of each infected node in the network, we compare the infection order of the obtained diffusion history with the true infection order. Since there could be ties in the infection order because more than one nodes may be infected in the same time slot, we use Kendall's τ_b statistic [31], which takes ties into consideration. The value of τ_b varies from -1 to 1, where $\tau_b = 1$ means the two orders are in perfect agreement and $\tau_b = -1$ means the two orders are perfect inversion to each other.
- **Edge precision:** From a diffusion history, we can further infer the set of edges involved in the diffusion process. We call edge $u \to v$ a diffusion edge if node v was infected by node u in information diffusion. Under the SI model, given a diffusion history, each edge $u \to v$ satisfying $t_u^I < t_v^I$ is a possible diffusion edge. Define $\mathcal{E}_d$ to be the set of possible diffusion edges based on the true diffusion history $\mathbf{X}$ and $\tilde{\mathcal{E}}_d$ to be the set of possible diffusion edges based on the reconstructed diffusion $\tilde{\mathbf{X}}$. We consider the following performance metric:

$$P = \frac{|\mathcal{E}_d \cap \tilde{\mathcal{E}}_d|}{|\mathcal{E}_d|}.$$

There is few work on using partial infection time information to reconstruct the diffusion history. The only one in the literature which can be used in our setting is A_ILP developed in [28]. Therefore, we compare our algorithm with A_ILP, and two other heuristics: the breadth-first-search (BFS) heuristic and the infection-simulation (IS) heuristic.

- BFS: On the infected subgraph, we construct the breadth-first search (BFS) tree from each possible source and set the infection time of a node to its distance to the root. Then we consider the set of infected nodes with observed infection time, and compare its infection order on the breadth-first search tree with the actual infection order using Kendall's τ_b coefficient. The BFS tree with the largest τ_b is chosen to be the diffusion history.
- IS: For each possible source, we generate an infection sequence using the SI model on the original network. The diffusion stops when the diffusion process "infects" all observed infected nodes. We again extract the infection order of the nodes with observed infection time and compare the order of it with the true infection order using Kendall's τ_b coefficient. The infection sequence with the largest τ_b is chosen as the diffusion history.

In [28], the source is assumed to be known. However, in our setting, the source of the diffusion process is unknown. Therefore, when we implement the A_ILP algorithm, we try to reconstruct the diffusion path for each possible source and then choose the reconstructed diffusion path with the largest value of optimization objective derived in [28] as the result of A_ILP.

We tested our algorithm on both synthetic diffusion data and real data. The networks used in generating the synthetic diffusion data include

- The power network: This network is used to represent the topology of the Western States Power Grid of the United States, which contains 4941 nodes and 6594 edges [3].
- The BA network: This is a network generated by using the Barabási-Albert model [32] with 300 nodes. Each new node is connected to 3 existing nodes.
- The IAS network: This is the Internet Autonomous Systems network [4], which contains 10670 nodes and 22002 edges. This is a small-world network.

In our experiment, we first generated a diffusion sequence by using the discrete time SI model with an equal infection probability p for each edge and a randomly chosen source. At time T, we took a snapshot of the network. Define s_{rate} to be the fraction of infected nodes with infection time observed. For example, if $s_{rate} = 20\%$, we randomly choose 20% of infected nodes and reveal their infection time.

We further evaluated the performance of our algorithm on the Weibo dataset provided by the WISE 2012 challenge[1], which contains the data of Sina Weibo[2], a famous microblogging website in China. The dataset consists of two parts: the friendship graph and a set of tweets.

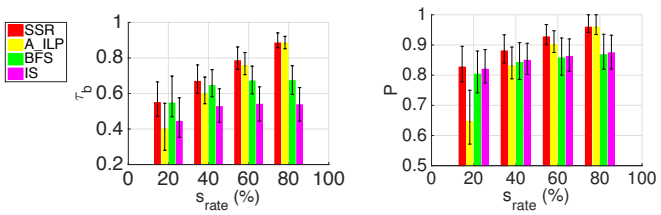
Since each tweet in the dataset contains the post time, user id, retweet path and message id, we extracted the tweets for a specific message and considered the post time of each tweet as the infection time of that user.

We pre-processed the dataset as follows:

1) We added links used in the retweet path into the friendship graph to form the network of the diffusion.
2) We removed the nodes whose infection time is not consistent with the network.
3) We selected the weakly connected component formed by all the nodes with infection time, and the first infected node on this component is viewed as the source.
4) We calculated the average infection time according to
$$\bar{t} = \frac{\sum_{(u,v)\in\mathcal{E}, t_u^I < t_v^I} (t_v^I - t_u^I)}{m},$$
where $\mathcal{E}$ is defined to be the set of edges on the weakly connected component and m is the number of directed edge (u, v) such that $t_u^I < t_v^I$. Here t_v^I is defined to be the infection time of node v.
5) We discretized the infection time according to
$$t'_u = \lceil \frac{t_u^I - t_s^I}{\bar{t}} \rceil,$$
where s is the first infected node of the component.
6) We deleted the node whose adjusted infection time is not consistent with the network structure.

After these steps, we obtained 357 diffusion traces with average size of 81.82 nodes/trace.

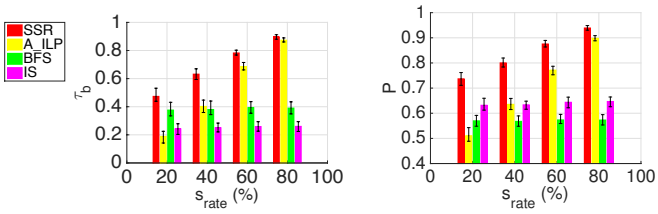
[1]http://www.wise2012.cs.ucy.ac.cy/challenge.html
[2]http://www.weibo.com



(a) Average τ_b for different s_{rate} with 25-75 percentile

(b) Average P for different s_{rate} with 25-75 percentile.

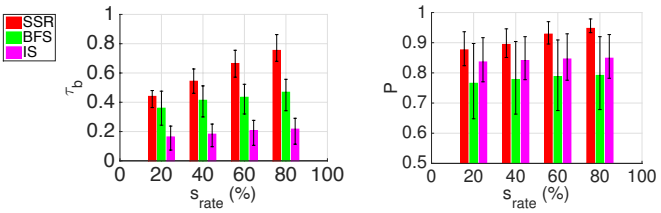
Fig. 3: Power Network with $p = 0.3$ and $T = 10$ (best viewed in color).



(a) Average τ_b for different s_{rate} with 25-75 percentile.

(b) Average P for different s_{rate} with 25-75 percentile.

Fig. 4: BA Network with $p = 0.3$ and $T = 10$ (best viewed in color).



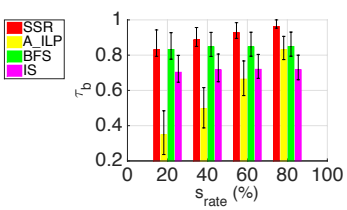
(a) Average τ_b for different s_{rate} with 25-75 percentile.

(b) Average P for different s_{rate} with 25-75 percentile.

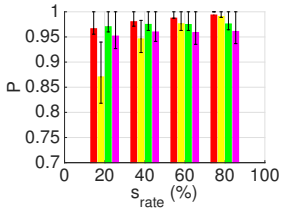
Fig. 5: IAS Network with $p = 0.04$ and $T = 4$ (best viewed in color).

A. *Performance Evaluation with Synthetic Diffusion Traces*

Figure 3, Figure 4 and Figure 5 show the performance of SSR and other algorithms based on the synthetic diffusion traces on different real-world networks. A_ILP has to solve a linear integer programming multiple times, and becomes very time-consuming on large-size networks such as the IAS network. So the performance of A_ILP is not included in Figure 5. In Power Network, BA Network and IAS Network,

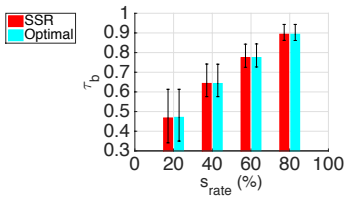


(a) Average τ_b for different s_{rate} with 25-75 percentile.

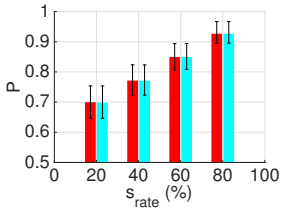


(b) Average P_1 for different s_{rate} with 25-75 percentile.

Fig. 6: Weibo dataset (best viewed in color).



(a) Average τ_b for different s_{rate} with 25-75 percentile.



(b) Average P_1 for different s_{rate}.

Fig. 7: Comparison with optimal on Zachery's Karate Club Network with $p = 0.3$ and $T = 10$ (best viewed in color).

our algorithm has the best performance under most of the sample rates in terms of all the metrics, which proves that our algorithm is prominent under tree-like network and small-world network.

B. Performance Evaluation with the Weibo Dataset

In the experiment based on the Weibo dataset, since we do not have the infection probability, we set the infection probability to be 0.8 for each edge. The performance of our algorithm as well as other algorithms is shown in Figure 6. From Figure 6, we can see that our algorithm has the best performance under most sample rates in terms of τ_b and P.

C. Optimality of the Single-Step Reconstruction

In the single-step reconstruction, we converted the problem to a weighted set cover problem, which is well-known to be NP-hard. Thus, we adopted the greedy set cover algorithm in [30], which provides a worst-case approximation ratio guarantee. In this set of simulations, we compared SSR using the greedy set cover solution for single-step reconstruction with SSR using the optimal solution for single-step reconstruction. Here, we used a small-size network, Zachary's Karate Club Network [33], which has 34 nodes and 78 edges. Figure 7 shows the results of the comparison between the two algorithm. We can see that the results are almost identical, which shows that the greedy solution performs reasonably well, at least for small size networks.

D. Efficiency Results

Figure 8 shows the wall-clock time of our algorithm versus the network size. In order to obtain Figure 8, at first, many

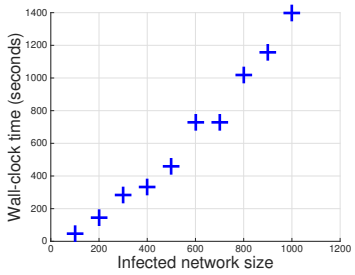


Fig. 8: Wall-clock time vs infected network size.

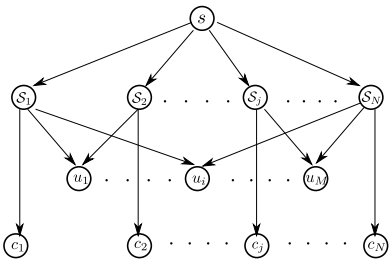


Fig. 9: The graph built in the proof of Theorem 1.

diffusion traces were generated based on the power network [3] with infection time T varying from 5 to 90. Then we used the sample rate 40%. We tested the wall-clock time of our algorithm on these diffusion traces and classified the wall-clock time by the number infected nodes. For example, a diffusion trace with 150 infected nodes was included into the calculation of size 200. Finally, Figure 8 was obtained by calculating the average wall-clock time of each cluster and plotting the figure of wall-clock time versus the cluster size. From Figure 8, we can see that the running time increases in a near-linear trend with the size of infected subnetwork. Note that the x-axis is the size of the infected subnetwork, which is much smaller than the size of the power network [3].

V. Conclusion

In this paper, we studied the problem of diffusion history reconstruction. We formulated the problem as an optimization problem and developed a step-by-step reconstruction algorithm, in which the single-step reconstruction can be converted to a weight set cover problem. Our simulation results show the superior performance over heuristic and existing algorithms. In this paper, we only talk about the single source diffusion process. For diffusion process involved multiple sources, our algorithm is still applicable in principle. However, the number of initial network states is $n-$combinations of $\mathcal{V}_I$, which will increase the complexity of the algorithm. Future work includes developing efficient algorithms to reconstruct diffusion history in such multiple sources scenarios.

Appendix A
Proof of Theorem 1

By reduction from the set cover problem. In the set cover, we are given a set of M elements $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and a set $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}$ of N sets whose union equals to set $\mathcal{U}$. The set cover problem is to identify the smallest subset

of $\mathcal{S}$ whose union equals to the universe. We consider each element in set $\mathcal{U}$ or set $\mathcal{S}$ as a vertex on a graph and think there exists an directed edge from S_j $(0 \leq j \leq N)$ to u_i $(0 \leq j \leq M$) if $u_i \in \mathcal{S}_j$. Then for each $\mathcal{S}_j \in \mathcal{S}$, we add a node c_j to the graph and create a directed edge from $\mathcal{S}_j$ to c_j. After that, we add a node s to the graph and create a directed edge from s to $\mathcal{S}_j$ for each $\mathcal{S}_j \in \mathcal{S}$. An example of the graph we built is provided in Figure 9. We assume there is a diffusion process based on the graph in Figure 9 and SI model. Assume the infection probability on each edge $(s, \mathcal{S}_j)$ $(0 \leq j \leq N)$ is p_1, the infection probability on each edge between $\mathcal{S}_j$ $(0 \leq j \leq N)$ and u_i $(0 \leq i \leq M)$ is 1, and the infection probability on each edge $(\mathcal{S}_j, c_j)$ $(0 \leq j \leq N)$ is p_2 $(p_2 > p_1)$. The snapshot was taken at time $T = 3$ and all nodes are in the infected state at time 3. We observed that the infection time of node u_i $(0 \leq i \leq M)$ is 2, the infection time of node c_j $(0 \leq j \leq N)$ is 3, and the infection time of node s is 0. Thus, we have known the observation $(\mathbf{I}, \mathbf{T})$. To determine the diffusion history, we need to decide when node $\mathcal{S}_j$ $(0 \leq j \leq N)$ is infected. We try to find a diffusion history by solving (2). Since c_j $(0 \leq j \leq N)$ is infected at time 3, node $\mathcal{S}_j$ can only be infected at time 1 or 2. If $\mathcal{S}_j$ is infected at time 1, the probability brought by path $s \to \mathcal{S}_j \to c_j$ is $p_1(1-p_2)p_2$. Otherwise, the the probability brought by path $s \to \mathcal{S}_j \to c_j$ is $p_1(1-p_1)p_2$. Since $p_2 > p_1$, we have $p_1(1-p_2)p_2 < p_1(1-p_1)p_2$, which means node $\mathcal{S}_j$ prefers to be infected at time 2. However, since u_i $(0 \leq i \leq M)$ is infected at time 2, for each u_i, there must exist a node $\mathcal{S}_k$ infected at time 1 such that $u_i \in \mathcal{S}_k$, which means u_i can be infected at time 2. Therefore, the problem described in (2) can be converted to a set cover problem of finding the smallest subset $\mathcal{S}'$ of $\mathcal{S}$ to cover $\mathcal{U}$. Then we may think the nodes in $\mathcal{S}'$ are infected at time 1, while others in $\mathcal{S}$ are infected at time 2. Since

$$\Pr(\mathbf{X}|\mathbf{I}, \mathbf{T}) \propto \Pr(\mathbf{X}) \Pr(\mathbf{I}, \mathbf{T}|\mathbf{X}) \tag{11}$$

and the value of $\Pr(\mathbf{I}, \mathbf{T}|\mathbf{X})$ can only be 1 or 0, this reconstructed diffusion history $\tilde{\mathbf{X}}$ has the maximum probability $\Pr(\tilde{\mathbf{X}})$ while $\tilde{\mathbf{X}}$ is consistent with $(\mathbf{I}, \mathbf{T})$, $\Pr(\mathbf{I}, \mathbf{T}|\mathbf{X}) = \mathbf{1}$. Thus, the set cover problem can be reduced to a special case of our diffusion history reconstruction problem, which means the diffusion history reconstruction problem is NP-hard.

Appendix B
Proof of Theorem 2

At first, we need to prove that if a feasible solution, $\tilde{\mathbf{X}}$ exists, each infected node is infected at a time that is consistent with its observation in $\tilde{X}$. For any $v \in \mathcal{V}$ with $I_v = 1$, define $t_v = T$ if $T_v \neq -1$, and $t_v = T_v$ if $T_v \neq -1$.

Assume in the feasible solution $\tilde{\boldsymbol{X}}$, v is not in the infected states at time t_v, which implies $X_{v,\tau} = 0$, for any $\tau \leq t_v$. Since the feasible solution exists, when we try to solve (10) at time $\tau = t_v$, after line 21 in Algorithm 2, we have $\mathcal{S}_v = \{v\}$, which means node v has to be infected at this time slot to guarantee the consistency of itself. Thus, according to Algorithm 2, we have $X_{v,t_v} = 1$, which contradicts the assumption. For any node v with infection time observed, according to Line 21 in Algorithm 2, by removing v from $\mathcal{S}$, we can guarantee that v cannot be infected before T_v. Thus, every infected node can be infected at a time consistent with its observation if a feasible solution $\tilde{\mathbf{X}}$ exists.

Next, we need to show that the feasible solution $\tilde{\boldsymbol{X}}$ exists. The set of possible sources contains the true source, $s^\star$. Since we want to prove at least one feasible solution can be found by our algorithm, we only discuss the diffusion history reconstruction starting from $s^\star$ and show that the diffusion history reconstructed is feasible. Assume $\tilde{\mathbf{X}}_0$ is the state, in which only $s^\star$ is infected.

When we reconstruct $\tilde{\mathbf{X}}_\tau$ by Algorithm 2 with $\tilde{\mathbf{X}}_{\tau-1}$ is known, for each node $v \in \mathcal{V}$ with $I_v = 1$ and $X_{v,\tau-1} = 0$, a feasible $\tilde{\mathbf{X}}_\tau$ exists if at least one of the following conditions holds:

- **C.1** There exists $u \in \mathcal{M}$ with $t_u \geq \tau$ and $v \neq n$ such that $d(u, v) \leq t_v - t_n$;
- **C.2** There exists $u \in \mathcal{V}$ with $X_{u,\tau-1} = 1$ such that $d(u, v) \leq t_v - \tau$;
- **C.3** There exists $u \in \tilde{\mathcal{S}}_{\tau-1} \cap \mathcal{I}$ with $T_v \leq \tau$ such that $d(u, v) \leq t_v - \tau$.

Here $d(u, v)$ represents the length of the shortest ITF-path between u and v.

For any $2 \leq t \leq T$, assume $\tilde{\mathbf{X}}_{t-1}$ exists and is known. Assume **C.1** holds when $\tau = t - 1$, which means for node v, there exists some $u \in \mathcal{M}$ with $X_{u,t-2} = 0$, such that $d(u, v) \leq t_v - t_u$. If $X_{u,t-1} = 0$, we have $u \in \mathcal{M}$, $t_u \geq t$ and $d(u, v) \leq t_v - t_u$, which means **C.1** is satisfied at $\tau = t$. Otherwise, if $X_{u,t-1} = 1$, which means $t_u = t - 1$, we have

$$\begin{aligned} d(u, v) &\leq t_v - t_u \\ &= t_v - t + 1. \end{aligned} \tag{12}$$

Then there exists either a susceptible node, w, which is an outgoing neighbor of u or an infected node w, on the shortest ITF-path between (u, v) such that $d(w, v) \leq t_v - t$. Thus, either **C.2** or **C.3** is satisfied at $\tau = t$.

Assume **C.2** holds when $\tau = t-1$, which means for node v, there exists u with $X_{u,t-2} = 1$ such that $d(u, v) \leq t_v - t + 1$. Then there exists either a susceptible node, w, which is an outgoing neighbor of u or an infected node w, on the shortest ITF-path between (u, v) such that $d(w, v) \leq t_v - t$. Thus, either **C.2** or **C.3** is satisfied at $\tau = t$.

Assume **C.3** holds when $\tau = t - 1$, which means for node v, there exists a non-empty set $\mathcal{D}$, for any $u \in \mathcal{D}$, we have $u \in \tilde{\mathcal{S}}_{t-2} \cap \mathcal{I}$ with $T_u \leq t - 1$ and $d(u, v) \leq t_v - t + 1$. If **C.1** or **C.2** is also satisfied at $\tau = t-1$, we know that Algorithm 2 can find a feasible $\tilde{\mathbf{X}}_t$. We assume only **C.3** holds at $\tau = t-1$. If for any $u \in \mathcal{D}$ we have $d(u, v) = t_v - t + 1$, at least one node $x \in \mathcal{D}$ needs to be infected at $t - 1$. Then there exists a susceptible node, w, which is an outgoing neighbor of x on the shortest ITF-path between (x, v) such that $d(w, v) \leq t_v - t$. Thus, **C.3** is satisfied at $\tau = t$. If there exists some $u \in \mathcal{D}$ with $d(u, v) < t_v - t + 1$, we have $d(u, v) \leq t_v - t$ and u can be in susceptible state or infected state at time t, which means **C.2** or **C.3** holds.

Therefore, if a feasible $\tilde{\mathbf{X}}_{t-1}$ exists, then we can always find a feasible $\tilde{\mathbf{X}}_t$. It is easy to check when $\tau = 1$, for each node $v \in \mathcal{V}$ with $I_v = 1$ and $X_{v,0} = 0$, at least one of the conditions **C.1**, **C.2** or **C.3**, is satisfied when the initial state is $\tilde{\mathbf{X}}_0$. Thus, we can find a feasible solution $\tilde{\mathbf{X}}$.

In summary, we can find a feasible solution $\tilde{\mathbf{X}}$ such that $P(Q, I|\tilde{\mathbf{X}}) = 1$.

Then we analyze the complexity of our algorithm. Define $G_I(\mathcal{V}_I, \mathcal{E}_I)$ to be the infected subgraph of $G(\mathcal{V}, \mathcal{E})$. In our algorithm, at first we try find the set of possible sources, whose complexity is $O(V_I E_I)$. Then the complexity for MBFS is also $O(V_I E_I)$. In each single step reconstruction, we need to prune the tree first. For each MBFS-tree, after pruning, the nodes removed from the MBFS-tree at the current step won't appear in the MBFS-tree for the future single step reconstructions. Thus, for a specific MBFS-tree, the worse case complexity for pruning in the diffusion history reconstruction is $O(V_I)$. Since there are at most V_I MBFS-trees, the worst case complexity brought by pruning the MBFS-trees for a specific initial state is $O(V_I^2)$. In a similar way, the worst case complexity brought by the greedy set cover is also $O(V_I^2)$ for a specific initial state. The number of initial states depends on the number of possible sources, which is V_I in the worst case. Therefore, the worst case complexity is $O(2V_I E_I + 2V_I^3)$. Since $E_I \leq V_I^2$, the complexity is $O(V_I^3)$. For each possible source, the procedure of the diffusion history reconstruction is the same. Thus, our algorithm can be done in a parallel manner, which means we use each computer to build a diffusion history for a specific source. Then, the worst case complexity for each machine is $O(V_I^2)$.

ACKNOWLEDGMENT

This work was supported in part by the U.S. Army Research Laboratory's Army Research Office (ARO Grant No. W911NF1310279).

REFERENCES

[1] E. Sefer and C. Kingsford, "Diffusion archaeology for diffusion progression history reconstruction," in *IEEE 14th Int. Conf. on Data Mining (ICDM)*, Shenzhen, China, Dec. 2014, pp. 530–539.

[2] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," in *Proc. of the Royal Soc. of London A: Math., Physical and Eng. Sci.*, vol. 115, no. 772, 1927, pp. 700–721.

[3] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[4] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proc. of the 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, Chicago, USA, Aug. 2005, pp. 177–187.

[5] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change in informational cascades," *J. of Political Economy*, vol. 100, no. 5, pp. 992–1026, Oct. 1992.

[6] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Lett.*, vol. 12, no. 3, pp. 211–223, 2001.

[7] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Washington DC, USA, Aug. 2003, pp. 137–146.

[8] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web (TWEB)*, vol. 1, no. 1, p. 5, 2007.

[9] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proc. of the 13th Int. Conf. on World Wide Web*, New York, USA, May 2004, pp. 491–501.

[10] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Edmonton, Canada, Jul. 2002, pp. 61–70.

[11] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review*, vol. 42, no. 4, pp. 599–653, 2000.

[12] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *IEEE Proc. 22nd Int. Symposium on Reliable Distributed Syst.*, Florence, Italy, 2003, pp. 25–34.

[13] A. Ganesh, E. Massouli, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2005, pp. 1455–1466.

[14] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos, "Threshold conditions for arbitrary cascade models on arbitrary networks," in *IEEE 11th Int. Conf. on Data Mining (ICDM)*, Vancouver, Canada, Dec. 2011, pp. 537–546.

[15] Y. Hayashi, M. Minoura, and J. Matsukubo, "Recoverable prevalence in growing scale-free networks and the effective immunization," *arXiv:cond-mat/0305549 v2*, Aug. 2003.

[16] H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau, "On the vulnerability of large graphs," in *IEEE 10th Int. Conf. on Data Mining (ICDM)*, Sydney, Australia, Dec. 2010, pp. 1091–1096.

[17] L. Briesemeister, P. Lincoln, and P. Porras, "Epidemic profiles and defense of scale-free networks," in *Proc. of the 2003 ACM Workshop on Rapid Malcode*, Washington DC, USA, Oct. 2003, pp. 67–75.

[18] B. A. Prakash, H. Tong, N. Valler, M. Faloutsos, and C. Faloutsos, "Virus propagation on time-varying networks: Theory and immunization algorithms," in *Mach. Learning and Knowledge Discovery in Databases*, 2010, pp. 99–114.

[19] N. Valler, B. A. Prakash, H. Tong, M. Faloutsos, and C. Faloutsos, "Epidemic spread in mobile ad hoc networks: Determining the tipping point," in *Networking 2011*, 2011, pp. 266–280.

[20] K. Zhu and L. Ying, "Information source detection in the SIR model: A sample path based approach," *IEEE/ACM Trans. Netw.*, 2015.

[21] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Trans. Inform. Theory*, vol. 57, no. 8, pp. 5163–5181, Aug. 2011.

[22] ——, "Rumor centrality: a universal source detector," in *SIGMETRICS Perform. Eval. Rev.*, New York, NY, USA, 2012, pp. 199–210.

[23] K. Zhu, Z. Chen, and L. Ying, "Locating the contagion source in networks with partial timestamps," *Data Mining and Knowledge Discovery*, pp. 1–32, 2015.

[24] M. Gomez Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *Proc. of the 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Washington DC, USA, Jul. 2010, pp. 1019–1028.

[25] S. Myers and J. Leskovec, "On the convexity of latent social network inference," in *Advances in Neural Inform. Process. Syst.*, 2010, pp. 1741–1749.

[26] B. Abrahao, F. Chierichetti, R. Kleinberg, and A. Panconesi, "Trace complexity of network inference," in *Proc. of the 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Chicago, USA, Aug. 2013, pp. 491–499.

[27] L. M. Gardner, D. Fajardo, and S. Travis Waller, "Inferring contagion patterns in social contact networks using a maximum likelihood approach," *Natural Hazards Review*, vol. 15, no. 3, 2014.

[28] D. Fajardo and L. M. Gardner, "Inferring contagion patterns in social contact networks with limited infection data," *Networks and Spatial Econ.*, vol. 13, no. 4, pp. 399–426, 2013.

[29] B. Zong, Y. Wu, A. K. Singh, and X. Yan, "Inferring the underlying structure of information cascades," in *IEEE 12th Int. Conf. on Data Mining (ICDM)*, Brussels, Belgium, Dec. 2012, pp. 1218–1223.

[30] N. E. Young, "Greedy set-cover algorithms (1974-1979, chvátal, johnson, lovász, stein)," *Encyclopedia of Algorithms*, pp. 379–381, 2008.

[31] A. Agresti, *Analysis of ordinal categorical data*. John Wiley & Sons, 2010, vol. 656.

[32] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Sci.*, vol. 286, no. 5439, pp. 509–512, 1999.

[33] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. of Anthropological Research*, pp. 452–473, 1977.