Knowledge and Information Systems (2022) 64:3373–3399
https://doi.org/10.1007/s10115-022-01748-8

REGULAR PAPER

Risk-aware temporal cascade reconstruction to detect asymptomatic cases

Hankyu Jang[1] · **Shreyas Pai**[2] · **Bijaya Adhikari**[1] · **Sriram V. Pemmaraju**[1]

Received: 28 January 2022 / Revised: 7 August 2022 / Accepted: 13 August 2022 /
Published online: 15 September 2022

Abstract
This paper studies the problem of detecting *asymptomatic cases* in a temporal contact network in which multiple outbreaks have occurred. We show that the key to detecting asymptomatic cases well is taking into account both individual risk and the likelihood of disease-flow along edges. We consider both aspects by formulating the asymptomatic case detection problem as a *directed prize-collecting Steiner tree* (DIRECTED PCST) problem. We present an approximation-preserving reduction from this problem to the *directed Steiner tree* problem and obtain scalable algorithms for the DIRECTED PCST problem on instances with more than 1.5M edges obtained from both synthetic and fine-grained hospital data. On synthetic data, we demonstrate that our detection methods significantly outperform various baselines (with a gain of 3.6×). We apply our method to the infectious disease prediction task by using an additional feature set that captures exposure to detected asymptomatic cases and show that our method outperforms all baselines. We further use our method to detect infection sources ("patient zero") of outbreaks that outperform baselines. We also demonstrate that the solutions returned by our approach are clinically meaningful by presenting case studies.

Keywords Asymptomatic cases · C. diff infections · Prize-collecting Steiner tree · Temporal contact networks

For the CDC MInD Healthcare Network.

This paper is an extended version of the work published in ICDM 2021.

✉ Sriram V. Pemmaraju
sriram-pemmaraju@uiowa.edu

Hankyu Jang
hankyu-jang@uiowa.edu

Shreyas Pai
shreyas.pai@aalto.fi

Bijaya Adhikari
bijaya-adhikari@uiowa.edu

[1] Department of Computer Science, University of Iowa, Iowa City 52242, IA, USA

[2] Department of Computer Science, Aalto University, Espoo, Finland

1 Introduction

For many infections, e.g., Zika virus disease, malaria, *methicillin-resistant Staphylococcus aureus* (MRSA) infection, and *Clostridioides difficile* (C. diff) *infection* (CDI), *asymptomatic* cases present a major obstacle to precisely understand how the infection is spread, and they make implementing effective interventions that much more challenging [21, 22, 28, 40]. Indeed, asymptomatic cases are widely believed to play a substantial role in the spread of COVID-19 [6], and asymptomatic transmission of SARS-CoV-2 has been called the "Achilles' heel" of control strategies for COVID-19.

Ideally, we would like to detect asymptomatic individuals and apply infection-control policies (e.g., quarantine, isolation) to them as well. However, detecting asymptomatic cases is challenging for several reasons. First, since asymptomatic cases do not show symptoms (by definition), only costly, blanket surveillance strategies can detect these cases. Second, asymptomatic cases may not have the same risk factors as symptomatic cases, and therefore, risk factors discovered for symptomatic cases may not be a valid proxy for asymptomatic cases. Third, from a data mining point of view, it is hard to learn risk factors for asymptomatic cases because "ground-truth" data on asymptomatic cases are essentially non-existent.

The focus of this paper is the detection of asymptomatic cases of *healthcare-associated infections* (HAIs). An HAI is an infection that a patient acquires in a healthcare facility while being treated for another condition. At any given time, one in 25 patients in the USA has an HAI [37]. CDI and MRSA infection are among the most common HAIs [37]. Some of the experimental results we present are for detecting asymptomatic cases of CDI, but our methods are widely applicable. The main novelty and strength of our approach are that it takes into account both individual risk and disease-flow through a contact network. Prior work on detecting "missing infections," e.g., [33, 41, 42], has largely ignored individual risk. The main takeaway from our results is that both aspects of disease spread are critical. When evaluated on large-scale synthetic data and actual hospital data, our approach outperforms methods that ignore either the individual risk or disease flow.

1.1 Informal problem description

Our input consists of a hospital mobility log that tells us time-stamped locations (e.g., hospital rooms) of patients and *healthcare professionals* (HCPs). We represent this mobility log as a temporal network $\mathcal{G} = (G_1, G_2, \ldots, G_T)$, where $G_i = (V_i, E_i, W_i, \mathbf{F}_i)$ is the static graph that captures interactions at time i. At each time i, the edge set E_i represents the interactions between nodes in V_i and W_i is the associated set of edge weights, representing the "strength" of these interactions. $\mathbf{F}_i[v]$ is the attribute vector for node $v \in V_i$ at time i, representing individual risk factors such as demographics, length of stay, and prescriptions. We assume that there is a hidden disease-spread process that starts independently from multiple sources at possibly different times. At each time-stamp i, the set of infected nodes $\mathcal{I}_i \subseteq V_i$ get a single chance to infect their healthy neighbors. A distinguishing feature of our model is that the attribute vector $\mathbf{F}_i[v]$ influences the likelihood of a node becoming infected. Each infected node also has a single chance to recover. Those nodes that are newly infected and those that fail to recover at time i are infected at the beginning of time-stamp $i + 1$. This process continues till time T. Additionally, we are given time-stamped positive test results for an HAI. In other words, for each time i, a subset $S_i \subseteq \mathcal{I}_i$ of the infected nodes are revealed to us and the remaining infected nodes $A_i = \mathcal{I}_i \setminus S_i$ are hidden asymptomatic cases. Our problem can now be stated informally as:

ASYMPTOMATIC CASE DETECTION
Given a temporal network $\mathcal{G} = (G_1, G_2, \ldots, G_T)$ and a sequence $(S_1, S_2, \ldots, S_T)$ of observed cases, find the asymptomatic cases $\mathcal{A} = \cup_{i=1}^{T} A_i$.

1.2 Solution approach and contributions

Our overall solution approach to the ASYMPTOMATIC CASE DETECTION problem is shown in Fig. 1. We now describe this approach while highlighting our main contributions.

- **Directed prize-collecting Steiner tree formulation:** We model the ASYMPTOMATIC CASE DETECTION problem as the *Directed prize-collecting Steiner tree* (DIRECTED PCST) problem. DIRECTED PCST takes two inputs: (i) a *time-expanded network* that models infection flow and observed infections and (ii) individual patients' risks (probabilities) of being colonized. The output to the DIRECTED PCST problem is a tree that uses a combination of edges likely to permit infection flow and nodes likely to be asymptomatic cases, thus taking into account these dual aspects of disease-spread. We identify nodes in the output tree that are not observed cases as asymptotic cases. Our work seems to be the first to apply the DIRECTED PCST formulation to problems in disease spread.
- **Scalable algorithms for DIRECTED PCST:** The DIRECTED PCST is computationally very challenging [17], even to solve approximately. We present a new approximation-preserving reduction from DIRECTED PCST to the *directed Steiner tree* (DST) problem. We then leverage this reduction to present three alternative algorithms for DIRECTED PCST: (i) an approximation algorithm via the greedy DST approximation algorithm of [7], (ii) a flow-based linear programming (LP) relaxation, and (iii) a simple and fast heuristic based on *minimum cost arborescence* (MCA). Using these algorithms, we are able to evaluate our approach for detecting asymptomatic cases on a time-expanded network containing more than 1.6 million edges.
- **Learning individual risk:** One of the inputs we provide to the DIRECTED PCST problem is individual patients' risks of being colonized. Learning these risks is a challenging problem by itself due to the absence of "ground-truth" data. We present an approach, grounded in CDI risk literature, to using patients' attributes such as demographics, length of stay, and prescriptions for learning patients' risks of being an asymptomatic CDI case. Our approach can be generalized to other HAIs.
- **Extensive large-scale evaluation:** We present extensive experimental evaluation of our approach on synthetically generated HAI data overlaid on temporal contact networks obtained from fine-grained mobility data from the University of Iowa Hospitals and Clinics (UIHC). The UIHC is an 800-bed comprehensive academic medical center and

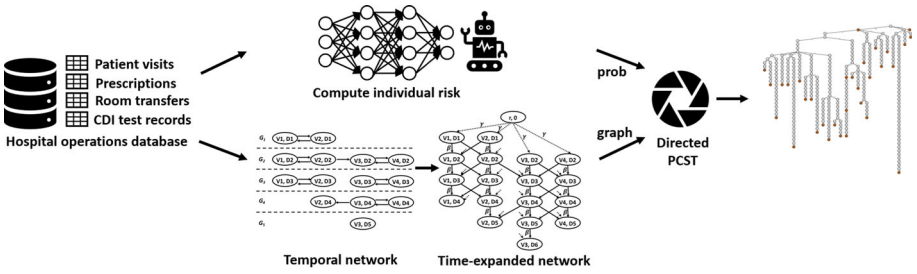


Fig. 1 This schematic shows our overall approach to solving the ASYMPTOMATIC CASE DETECTION problem

a regional referral center. Our approaches significantly outperform all the baselines, including CuLT [33], a Steiner-tree-based approach that ignores individual risk. Our best performing method achieves an F_1-score of 0.281, while our nearest competitor achieves only 0.078.
- **Application to predicting CDI cases:** We present a novel application of our methods to predicting (symptomatic) CDI cases at the UIHC. Using asymptomatic cases identified by our method, we create new features that we call *asymptomatic pressures* that measure exposure to asymptomatic cases. We then compare models for symptomatic HAI prediction that include these asymptomatic pressures against (i) models that do not include these pressures and (ii) models that include these pressures, but computed via other methods (e.g., CuLT). We show that using asymptomatic pressures computed by our method as a feature significantly outperforms all other competitors.
- **Application to source detection:** We also leverage our methods to detect sources of HAI outbreaks. Our approaches reconstruct temporal cascades by extracting temporal forests (a collection of temporal trees) corresponding to the "most likely" routes taken by the underlying infection flow. Note that our cascades span both the observed cases and nodes with high individual risks. We then simply select the roots of the reconstructed cascades as the sources of the outbreak. Finally, we demonstrate that the cascades reconstructed by our approaches lead to more accurate sources than the baseline approaches.
- **Case studies:** We also demonstrate that the inferred asymptomatic cases are clinically meaningful via case studies. Specifically, we investigate the details of (anonymous) patients at the UIHC who have been identified as asymptomatic CDI cases by our methods.

2 Problem formulation

In this section, we formalize the ASYMPTOMATIC CASE DETECTION problem informally stated in the previous section. First, we assume that we have learned a function A from the space of feature vectors $\mathbf{F}_i$ to [0, 1], representing probabilities that nodes (which are not positive HAI cases) are asymptomatically infected. Given that no "ground-truth" data are available on asymptomatic infections, this by itself is a non-trivial problem. We address this in Sect. 5 for CDI, but in principle our methods can be used for any HAI. Second, we transform the temporal network $\mathcal{G} = (G_1, G_2, \ldots, G_T)$ and observed case sequence $(S_1, S_2, \ldots, S_T)$ into a time-expanded network $G_S(V_S, E_S, r, S, W_e, W_v)$ with edge weights W_e, node weights W_v, a set $S \subseteq V_S$ of terminals, and a root $r \in V_S$. We describe this transformation below (Fig. 2).

- **Nodes:** Consider V_i, the node set for the time-i contact network G_i. For each node $v \in V_i$, we add two nodes (v, i) and $(v, i+1)$ to V_S. (Note that if $v \in V_i$ and $v \in V_{i+1}$, then $(v, i+1)$ is added only once to V_S.) We use the term *layer* i to denote the subset of all nodes in V_S whose time-stamp label is i.
- **Edges:** For each edge (u, v) in G_i, we create a "cross" edge $((u, i), (v, i+1))$. Additionally, for every $v \in V_i$, we create a "straight" edge $((v, i), (v, i+1))$.
- **Edge weights:** The "cross" edge $((u, i), (v, i+1))$ in E_S inherit its weight from the edge (u, v), i.e., it is assigned weight $W_i(u, v)$. For some parameter, $\beta > 0$, all "straight" edges of the form $((v, i), (v, i+1))$ are assigned weight β. This assignment of edge weights in G_S is denoted by W_e.
- **Node weights:** Each node $(v, i+1)$ in G_s is assigned the probability $A(\mathbf{F}_i[v])$.

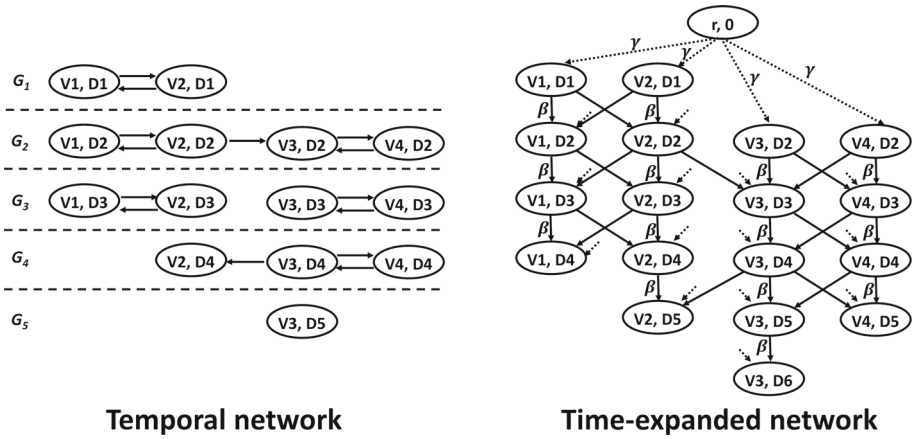


Fig. 2 The temporal graph $\mathcal{G}$ on the left is transformed into the time-expanded network $G_S(V_S, E_S, r, S, W_e, W_v)$ on the right. Even though, in order to avoid clutter, the figure only shows four edges leaving node r, there is an edge from r to every node in the graph with weight γ

- **Terminals:** The set of observed cases $\mathcal{S}$ is designated the set of terminals of the graph G_S.
- **Root:** We add a "dummy" root node r to V_S and connect it to every other node in V_S. For some parameter $\gamma > 0$, we make γ the weight of every edge leaving r. The γ parameter controls the number of connected components in our solution upon removal of r. These connected components are trees and can be interpreted as distinct outbreaks. Larger values of γ will favor few outbreaks in an optimal solution.

An important (and easily verified) observation about G_S is that it is a directed acyclic graph (DAG). This property of G_S will play a crucial role in the efficiency of the algorithms we consider in Sect. 3.

We now formulate a precise version of the ASYMPTOMATIC CASE DETECTION problem as a *directed prize-collecting Steiner tree* problem (DIRECTED PCST).

DIRECTED PRIZE- COLLECTING STEINER TREE (DIRECTED PCST)
Given $G_S(V, E, r, S, W_e, W_v)$ and a parameter $\alpha > 0$, find a tree $T^*(V^*, E^*)$ rooted at r and spanning terminal set S, such that

$$T^* = \arg\min_{T} \sum_{(a,b)\in E(T)} W_e(a,b) + \alpha \cdot \sum_{a\in V\setminus V(T)} W_v(a) \tag{1}$$

The objective function of the DIRECTED PCST problem aims to balance two weights: one due to *edges included* in the tree and other due to *nodes excluded* from the tree. As a result, an optimal DIRECTED PCST solution T^* uses a combination of low-weight edges and high-weight nodes. The connection between DIRECTED PCST and the ASYMPTOMATIC CASE DETECTION problem is now natural. Given a tree T^* that is a solution for DIRECTED PCST, we interpret the non-terminal nodes in T^* as likely asymptomatic infections.

The parameter α provides a way of controlling the relative importance of included edge weights versus excluded node weights. A large value of α places more importance on node weights. Setting $\alpha = 0$ yields the DST [7, 44] problem as a special case. While the DIRECTED PCST problem is parameterized by α, the time-expanded network G_S that is input to the

problem is parameterized by quantities β and γ. In our experiments, we explore the space of these three parameters.

3 Scalable algorithms for DIRECTED PCST

The DIRECTED PCST problem is computationally very challenging. In fact, its special case, the DST problem is also very challenging. It has been shown by [17] that there is no quasi-polynomial-time algorithm for DST that achieves an approximation ratio of $O(\log^{2-\epsilon} k)$, for any constant $\epsilon > 0$, unless all problems in NP can be solved in time $O(n^{\text{polylog } n})$ by zero-error probabilistic algorithms. While there are constant-factor approximation algorithms for the undirected version of PCST [3], except for the message-passing heuristic [36], which provides no approximation guarantee, nothing seems to be known for DIRECTED PCST. In fact, this is the situation not just for arbitrary directed graphs, but also for DAGs [44]; this can also be seen in Theorem 1 in [32]).

We present the following three approaches to solving the DIRECTED PCST. All three approaches depend on an approximation-preserving reduction from DIRECTED PCST to DST, which we provide in Sect. 3.1. (i) We use the greedy DST approximation algorithm of [7] to approximately solve DIRECTED PCST (Sect. 3.2). (ii) We solve a flow-based LP relaxation of DST [32] (Sect. 3.3). Even though solution returned by the LP is fractional, as we show below, it can still be meaningfully interpreted in the context of the ASYMPTOMATIC CASE DETECTION problem. (iii) We solve the MCA problem on the metric graph induced by the terminal set S and the root r (Sect. 3.4). Even though this approach does not come with a provable approximation guarantee, our experimental results indicate that this is a fast algorithm that outputs near-optimal solution.

3.1 Reducing DIRECTED PCST to directed steiner tree

We reduce DIRECTED PCST to DST as follows. Let $E_S \subseteq E$ denote the edge set $\{(a, b) \in E \mid b \in S\}$. Let $\Psi := \sum_{a \in V} W_v(a)$. From G_S, we create a new graph $G'(V, E, r, S, W'_e)$ with *only* edge weights, given by the function $W'_e : E \to \mathbb{R}$, such that for all $(a, b) \in E$

$$W'_e(a, b) = \begin{cases} W_e(a, b) - \alpha \cdot W_v(b), & \text{for } (a, b) \in E \setminus E_S \\ W_e(a, b) + \alpha \cdot \frac{\Psi}{|S|}, & \text{for } (a, b) \in E_S. \end{cases}$$

The reduction is illustrated in Fig. 3. Note that the new edge weights $W'_e(a, b)$ can be negative, especially for large α. The reduction is quite efficient, taking $O(n + m)$ time when the graph G_S has n nodes and m edges. For any directed tree T in G that is rooted at r and spans S, let $W_{PCST}(T, G)$ denote the objective function value (i.e., the expression in (1)) of tree T for the DIRECTED PCST problem on graph G_S. For any directed tree T in G' that is rooted at r and spans S, let $W_{DST}(T, G')$ denote the objective function value of tree T for the DST problem on graph G'. We prove the following lemma.

Lemma 1 *For any directed tree $T(V_T, E_T)$, $V_T \subseteq V$, $E_T \subseteq E$, rooted at r and spanning S, $W_{DST}(T, G') = W_{PCST}(T, G)$. Furthermore, if T is an optimal directed Steiner tree for G', then T is also an optimal prize-collecting Steiner tree for G.*

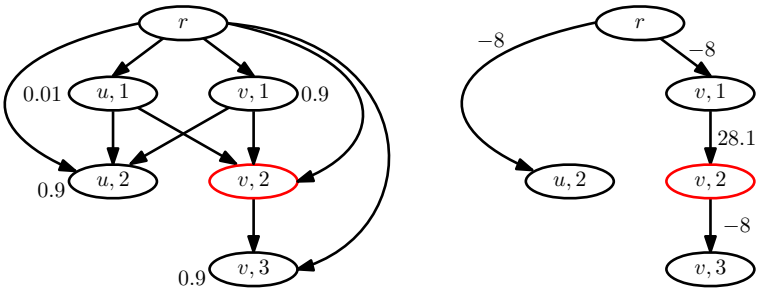


Fig. 3 **a** A time-expanded network G_S with terminal set $S = \{(v, 2)\}$ (shown in red) and node weights is shown. To obtain edge weights assume that $\beta = \gamma = 1$ and $W_e((u, 1), (v, 2)) = W_e((v, 1), (u, 2)) = 1$. Suppose we want to solve DIRECTED PCST with $\alpha = 10$. **b** After reducing DIRECTED PCST to DST, we get a graph G' with modified edge weights and no node weights. For example, $W'_e(r, (v, 1)) = 1 - 10 \times 0.9 = -8$ and $W'_e((v, 1), (v, 2)) = 1 + 10 \times (2.71/1) = 28.1$. Note here that $\Psi = 2.71$. An optimal directed Steiner tree T on graph G' is shown on the right; $W_{DST}(T, G') = 3 \times (-8) + 28.1 = 4.1$. Also note that $W_{PCST}(T, G_S) = 4 \times 1 + 10 \times 0.01 = 4.1$, showing that $W_{DST}(T, G') = W_{PCST}(T, G_S)$. Note that since edge weights can be negative, leaves of an optimal directed Steiner tree need not be terminals

Proof Let $T(V_T, E_T)$, $V_T \subseteq V$, $E_T \subseteq E$, be an arbitrary directed tree rooted at r and spanning S. Then,

$$W_{DST}(T, G') = \sum_{(a,b)\in E(T)} W'_e(a, b).$$

The right-hand side of this equality can be simplified as

$$\begin{aligned}
&= \sum_{(a,b)\in E(T)\setminus E_S} \Big(W_e(a, b) - \alpha \cdot W_v(b)\Big) + \sum_{(a,b)\in T\cap E_S} \Big(W_e(a, b) + \alpha \cdot \frac{\Psi}{|S|}\Big) \\
&= \sum_{(a,b)\in E(T)} W_e(a, b) + \alpha \cdot \Psi - \alpha \cdot \sum_{a\in V(T)} W_v(a) \\
&= \sum_{(a,b)\in E(T)} W_e(a, b) + \alpha \cdot \sum_{a\in V\setminus V(T)} W_v(a) \\
&= W_{PCST}(T, G).
\end{aligned}$$

Now suppose that T is an optimal directed Steiner tree for G'. To obtain a contradiction, let us assume T is not the optimal prize-collecting Steiner tree for G. It implies there is another tree directed tree T', rooted at r and spanning S such that $W_{PCST}(T', G) < W_{PCST}(T, G)$. Since $W_{DST}(T', G') = W_{PCST}(T', G)$, this means that $W_{DST}(T', G') < W_{DST}(T, G')$, contradicting the fact that T is an optimal directed Steiner tree for G'. □

In fact, we prove a stronger, approximation-preserving relation between the two problems as shown by the following lemma.

Lemma 2 *For any $\rho \geq 1$, if a tree T is a ρ-approximate directed Steiner tree for G', then T is a ρ-approximate directed prize-collecting Steiner tree for G.*

Proof Since T is a ρ-approximate directed Steiner tree for G',

$$W_{DST}(T, G') \leq \rho \cdot W_{DST}(T^*, G'),$$

where T^* is an optimal directed Steiner tree for G'. By Lemma 1, $W_{PCST}(T, G) = W_{DST}(T, G')$ and $W_{PCST}(T^*, G) = W_{DST}(T^*, G')$. Therefore,

$$W_{PCST}(T, G) \leq \rho \cdot W_{PCST}(T^*, G). \tag{2}$$

By Lemma 1, T^* is also an optimal prize-collecting Steiner tree for G, and therefore, using (2), we see that T is a ρ-approximate directed prize-collecting Steiner tree for G. □

3.2 Greedily solving DST approximately

Having reduced DIRECTED PCST to DST, we use the clever, greedy algorithm of [7] to obtain an approximation algorithm for DIRECTED PCST. The Charikar et al. algorithm achieves an $O(i^2k^{1/i})$-approximation ratio in time $O(n^ik^{2i})$ for any fixed integer $i \geq 1$, where k is the number of terminals. Setting $i = 1$ gives an $O(k)$ approximation algorithm in $O(nk^2)$ time and setting $i = 2$ gives an $O(\sqrt{k})$ approximation algorithm in $O(n^2k^4)$ time. We use GREEDY$_i$ to denote the Charikar et al. algorithm with parameter i.

3.3 Using an LP relaxation of DST

Given a directed graph $G'(V, E, r, S, W'_e)$, with a root node $r \in V$, terminal set $S \subseteq V$, and edge-weight function $W'_e : E \to \mathbb{R}$, the DST problem can be modeled by the following flow-based integer linear program (ILP) [32]. The variables $f_{s,e} \in \{0, 1\}$ for each $s \in S$ and $e \in E$ represent the presence of 1 unit of flow from the root r to terminal s via edge e. The variable $y_e \in \{0, 1\}$ indicates the use of edge e for *some* flow. For any node $v \in V$, $\delta^+(v)$ (respectively, $\delta^-(v)$) is the set of edges leaving (respectively, entering) v. The first set of constraints ensures that there is 1 unit of flow leaving the root, for each terminal s. The second set of constraints ensures that there is 1 unit of flow intended for s entering each terminal s. The third set of constraints ensures flow conservation, of flows intended for all terminals, at all nodes. The fourth set of constraints forces the variables y_e, for each $e \in E$, to be 1 only when edge e is used for some flow. The fifth constraint set ($\sum_{e\in\delta^-(v)} y_e \leq 1$) ensures that every vertex v has at most one unit of incoming flow and this in turn ensures that the paths induced by the flows form a tree.

$$\begin{aligned}
\min \sum_{e\in E} W'_e(e) \cdot y_e & \\
\text{s.t.} \quad \sum_{e\in\delta^+(r)} f_{s,e} - \sum_{e\in\delta^-(r)} f_{s,e} &= 1 && \forall s \in S \\
\sum_{e\in\delta^+(s)} f_{s,e} - \sum_{e\in\delta^-(s)} f_{s,e} &= -1 && \forall s \in S \\
\sum_{e\in\delta^+(v)} f_{s,e} - \sum_{e\in\delta^-(v)} f_{s,e} &= 0 && \forall v \in V \setminus \{r\}, s \in S \setminus \{v\} \\
f_{s,e} &\leq y_e && \forall s \in S, e \in E \\
\sum_{e\in\delta^-(v)} y_e &\leq 1 && \forall v \in V \\
f_{s,e} &\in \{0, 1\} && \forall s \in S, e \in E \\
y_e &\in \{0, 1\} && \forall e \in E
\end{aligned}$$

It is easy to verify that this ILP models DST. An LP relaxation of this ILP is obtained by replacing the two sets of integrality constraints at the end of the program by $0 \leq f_{s,e} \leq 1$ and $0 \leq y_e \leq 1$ for all $s \in S, e \in E$. While solving the ILP optimally is not computationally feasible, solving this LP relaxation is. The following theorem formalizes the connection between DIRECTED PCST and this LP relaxation and indicates how we use the LP relaxation.

Theorem 1 *Let T^* be an optimal directed prize-collecting Steiner tree for input $G_S(V, E, r, S, W_e, W_v)$ and parameter $\alpha > 0$. Let the graph $G'(V, E, r, S, W'_e)$ be obtained from G_S via the reduction in Sect. 3.1. Let W^* be the cost of the solution returned by the above LP relaxation on G'. Then, $W^* \leq W_{PCST}(T^*, G)$.*

Proof Let G' be the graph obtained from G and parameter α via the reduction in Sect. 3.1. Suppose T^* is an optimal directed Steiner tree for G'. Then, by Lemma 1, T^* is an optimal directed prize-collecting Steiner tree for G and $W_{DST}(T^*, G') = W_{PCST}(T^*, G)$.

Consider the DST LP relaxation for graph G'. A feasible solution to this LP can be obtained from T^* as follows. There is a unique path P_s in T^* from r to each terminal s. For each edge e in P_s, set $f_{s,e} = 1$; set $f_{s,e} = 0$ for all other e. Set $y_e = 1$ for all e in T^*. The following facts are easy to verify: (i) the above-described setting of variables $f_{s,e}$, y_e is feasible for the LP and (ii) the objective function value with this setting of variables is $W_{DST}(T^*, G')$. Together (i) and (ii) imply that $W^* \leq W_{DST}(T^*, G')$. Therefore, $W^* \leq W_{PCST}(T^*, G)$. □

At first glance, it may be unclear whether a fractional solution to the LP relaxation has a useful interpretation in the context of identifying asymptomatic cases. We propose the following interpretation. For an integral solution to the LP, for each non-terminal node v, either $\sum_{e \in \delta^-(v)} y_e = 1$ or $\sum_{e \in \delta^-(v)} y_e = 0$. If the former is true, then v is in the tree and consider an asymptomatic case. For a fractional solution to the LP, for each non-terminal node v, $0 \leq \sum_{e \in \delta^-(v)} y_e \leq 1$, and we interpret $\sum_{e \in \delta^-(v)} y_e$ as the probability that v is an asymptomatic case. This idea is inspired by the technique of randomized rounding [29] for obtaining good integral solutions from optimal fractional solutions. Note that with this approach, we are not finding a Steiner tree; we are simply identifying a set of non-terminal nodes, i.e., asymptomatic cases. In summary, our algorithm for identifying asymptomatic cases, based on the DST LP relaxation, is described in the following.

Algorithm: LP-based Asymptomatic Detection

1. Reduce the input $G_s(V, E, r, W_e, W_v)$ and $\alpha > 0$ of DIRECTED PCST to the input $G'(V, E, r, S, W'_e)$ of DST (as described in Section 3.1).
2. Set up LP relaxation of DST (as described above) and solve to obtain $f_{s,e}$ and y_e values for all terminals $s \in S$ and edges $e \in E$.
3. Let $p(v) = \sum_{e \in \delta^-(v)} y_e$ for each non-terminal node $v \in V \setminus S$.
4. Independently select each non-terminal node $v \in V \setminus S$ to be an asymptomatic case with probability $p(v)$.

The size of an LP is defined by the number of variables and the number of constraints it uses. The number of variables in the LP relaxation of DST is $m(k + 1)$, and the number of constraints is $O(mk)$. While LPs can be solved in polynomial time, in the number of variables and constraints, a specific running time is difficult to mention. This is because state-of-the-art LP solvers use a combination of algorithms (including the simplex method that runs in worst-case exponential time) and a wide variety of heuristics.

3.4 Minimum cost arborescence heuristic

Given an edge-weighted directed graph $G(V, E, W_e)$ and a vertex $r \in V$, an *arborescence* (rooted at r) is a tree T such that (1) T is a spanning tree of G if we ignore the direction of edges and (2) there is a unique directed path in T from r to each other node $v \in V$. An MCA is an arborescence of smallest total weight.

For general directed graphs with n nodes and m edges, we can compute an MCA in $O(m + n \log n)$ time [16]. This improves the naive implementation that runs in $O(nm)$ time. For DAGs, this algorithm can be simplified to run in $O(m+n)$ time. The algorithm is simply this: For each node $v \neq r$, add to the solution the edge incoming into v with minimum edge weight (breaking ties arbitrarily). Since the input is a DAG, there is no danger that this algorithm will create a cycle.

We use an MCA algorithm to produce a directed prize-collecting Steiner tree on $G_S(V, E, r, S, W_e, W_v)$ as follows. We first transform G_S into $G'(V, E, r, S, W'_e)$ as per the reduction from DIRECTED PCST to DST from Sect. 3.1. From G', we construct a new directed graph H whose vertex set is $S \cup \{r\}$. We add an edge (u, v) to H iff there is a directed path in G' from u to v. The weight assigned to (u, v) in H is the shortest path distance from u to v in G'. It is easy to verify that since G' is a DAG, H is also a DAG. To obtain a directed Steiner tree on G', we compute an MCA on H, replace each edge (u, v) in the MCA by a shortest path in G' from u to v, and finally return tree obtained by taking the union of these shortest paths. This algorithm is summarized in the following.

Algorithm: MCA-based Directed PCST

1. Reduce the input $G_s(V, E, r, W_e, W_v)$ and $\alpha > 0$ of DIRECTED PCST to the input $G'(V, E, r, S, W'_e)$ of DST (as described in Section 3.1).
2. From G' construct an edge-weighted directed graph $H = (S \cup \{r\}, E_H, W_H)$, where E_H consists of all edges (u, v) such that there is directed path from u to v in G' and W_H is an edge weight function that assigns to each edge (u, v) in H the length of a shortest path from u to v in G'.
3. Solve MCA on H to construct an arborescence T_H rooted at r.
4. Construct a Steiner tree T of G' by processing each edge (u, v) in T_H and adding to T a shortest path from u to v in G'.

Suppose that the input to the algorithm, G_S, has n nodes, m edges, and k terminals. As mentioned earlier, the reduction in Step 1 takes $O(m + n)$ time. Step 2, which involves solving $k + 1$ single source shortest path (SSSP) problems, takes $O(k(n + m))$ time. Note that even though some of the edges in G' have negative weights, solving SSSP takes only $O(n + m)$ because G' is a DAG. Step 3 could take $\Theta(k^2)$ time in the worst case because H could have $\Theta(k^2)$ edges. However, since typically $k \ll n$, Step 3 will be cheap relative to Step 2. Step 4 can be completed in $O(n)$ time since shortest paths have already been computed in Step 2. Thus, the overall running time, which is dominated by Step 2, is $O(k(n + m))$.

4 Processing and generating data

Our experimental results use an extensive, fine-grained hospital operations dataset collected from the *University of Iowa Hospitals and Clinics* (UIHC). The subset of these data used in this paper consists of architecture data (complete set of CAD files for a 3.2M square feet

facility), admission-discharge-transfer data (273K inpatient hospitalizations between 2003 and 2013), prescription data (7.8M prescriptions), and surveillance data (2K positive CDI lab tests between 2005 and 2011). Using these data and given a size-T time window, we construct a temporal network $\mathcal{G} = (G_1, G_2, \ldots, G_T)$ and a sequence of observed cases $(S_1, S_2, \ldots, S_T)$, as described next.

Note: All individuals present in our data (patients and HCPs) are completely anonymous. For this reason, this project was human subjects research exempt.

4.1 Constructing temporal graph from raw data

The subset of data relevant to our experiments consists of the following elements:

1. A collection X of patient visits. Each visit $x \in X$ spans a sequence of consecutive days denoted by the range $[s(x), e(x)]$, and for each day $d \in [s(x), e(x)]$ of a visit x, there is an associated location (patient room) denoted $\ell(x, d)$.
2. The set of locations is denoted by L, and there is a distance metric $D : L \times L \to \mathbb{R}^+$ defined on this set. In our past works [10, 11], we have discretized CAD drawings of the facility to obtain a "walking distance" metric between all pairs of rooms in the hospital. This is represented by D.
3. A partition of $X = C \cup N$, into CDI visits and non-CDI visits. For each CDI visit $x \in C$, there is a day $d \in [s(x), e(x)]$ that corresponds to a positive CDI test; we denote this day of positive test by $d^+(x)$.
4. For each visit $x \in X$, we have associated demographic features and whether there was a previous visit to the hospital within 60 days. In addition, we have features that change over time. Specifically, for each day $i \in [s(x), e(x)]$, we have the length of stay (from admission time to day i) and a list of high-risk antibiotics and gastric acid suppressors prescribed to the patient for day i of the visit. Finally, we also have "exposure" features, i.e., counts of the number of other CDI patients in the same room or unit.

We now fix a time window size T (in days) and without loss of generality assume that the days in this time window are labeled $1, 2, \ldots, T$. For each $i = 1, 2, \ldots, T$, we construct the directed network $G_i = (V_i, E_i, W_i, \mathbf{F}_i)$ and observed case sequence $(S_1, S_2, \ldots, S_T)$ as follows:

- **Node set** V_i: If $i \in [s(x), e(x)]$ for a patient visit $x \in X$, we add x to V_i. In other words, V_i is the set of all patient visits that are taking place on day i.
- **Edge set** E_i: For every $x, y \in V_i$, if locations $\ell(x, i)$ and $\ell(y, i)$ *belong to the same hospital unit*, then we add two directed edges (x, y) and (y, x) to E_i. In other words, all ordered pairs of nodes in V_i that are located (on day i) in the same unit are connected by edges. If locations $\ell(x, i)$ and $\ell(y, i)$ do not belong to the same unit, then for small probability $p \in [0, 1]$ (e.g., $p = 0.01$), we randomly (and independently) add edge (x, y) to E_i. These "long-distance" edges model "weak ties" induced by HCPs (especially physicians) who travel between units. Note that the preponderance of HCP mobility is within units.
- **Weights** W_i: For every edge $(x, y) \in E_i$, we set $W_e(x, y) = D(\ell(x, i), \ell(y, i))$. In other words, edge weights simply represent physical distance between pairs of hospital rooms.
- **Feature vector** F_i: For every node $x \in V_i$, we set $F_i[x]$ using the features described earlier in item (4).
- **Observed cases** S_i: For any node $x \in V_i$, if $x \in C$, i.e., x is a CDI visit, and $d^+(x) = i$, then x is added to S_i.

From this temporal network $\mathcal{G}$, we obtain a time-expanded network $G_S(V_S, E_S, r, S, W_e, W_v)$ as described in Sect. 2. Note that learning the node weights W_v, which represent the probability of a patient being an asymptomatic, is described in Sect. 5.

4.2 Generating synthetic "ground-truth" data

Since we do not have "ground-truth" data on asymptomatic infections, we also use the data described in the previous section to generate synthetic data via a partially hidden, "biased" *susceptible–infectious–susceptible* (SIS) process. The SIS model is designed for infections with no long-lasting immunity; any susceptible agent can get infected with a probability upon contact with an infectious agent, and an infected agent returns to a susceptible state with some probability. Here, we implement a *biased* version of the SIS process. Every node v has an assigned probability (i.e., a bias) that represents the individual node v's risk of being an asymptomatic case; node v participates in the process with this bias. We now describe this process in more detail, carefully differentiating between aspects that are hidden and aspects that are revealed.

Our implementation of the biased SIS process has three main steps.

1. **Generating biases and susceptible nodes.** Recall that in G_S, each node v is assigned a probability $W_v(v)$ that represents the individual node v's risk of being an asymptomatic case. From the distribution of the W_v values, we first learn a probability density function using *kernel density estimation* (KDE). Next, for every visit $x \in X$, we sample a bias $W_x \in [0, 1]$ from the estimated probability density function and then set a bit s_x to 1, independently, with probability proportional to W_x. We then project these quantities, associated with visits, onto individual nodes in G_s. Specifically, every node (x, i) in G_S that corresponds to visit x is assigned the probability W_x; i.e., $W_v(x, i) = W_x$, and we set the state of (x, i) to "susceptible" if $s_x = 1$. For every node (x, i) in G_S, the probability $W_v(x, i)$ is revealed to us, but the state of the node remains hidden.
2. **Running the biased SIS process.** Now, for some positive integer parameter k, we pick k infection sources at random from the set of susceptible nodes. Then, we run an SIS process starting at each of these k sources. Two aspects of the SIS process are worth noting: (i) only the susceptible nodes participate in the SIS process and (ii) the probability of infection flowing along an edge $((x, t), (y, t+1))$ in G_S is inversely proportional to the edge weight $W_e((x, t), (y, t + 1))$. Note that the k sources and the SIS process is entirely hidden from us.
3. **Revealing observed infections.** After running the SIS process, we visit every infected node (x, i) in each of the k infection trees and with a fixed probability q, we independently *reveal* (x, i) to be infected. These revealed infected nodes form the observed set of infections, and we use these as the set S of terminals. We then prune each infection tree so that all leaves are revealed infected nodes. The nodes in each pruned tree that are not revealed to be infected are considered asymptomatically infected.

In summary, the biases and observed infections are revealed to our algorithms, but everything else is hidden. However, we are able to evaluate the performance of our algorithms because the above-described process also provides us with "ground-truth" asymptomatic cases. We emphasize that a critical aspect of our setup is the fact that the SIS process is influenced by the revealed biases. We also note that our experimental setup is quite flexible. For example, the simple functions that govern the relationship between node biases and node susceptibility or edge weights and the likelihood of infection flow along edges can be easily replaced by other, more complicated functions.

For the time-expanded network G_S obtained from 1 month of data (20.9K nodes and 0.5M edges), for values of $\beta = 1, 2, 4$, we generate 18, 20, and 17 terminal nodes, respectively, and 40, 49, and 47 asymptomatic cases, respectively.

5 Learning individual risks

In this section, we describe the training of a model that takes as input the feature vector $\mathbf{F}_i[x]$ of each node $x \notin S_i$ in graph G_i (i.e., nodes not observed to be infected) and estimates the likelihood of x being an asymptomatic CDI case. As mentioned earlier, the fundamental obstacle to training this model is the fact that our data lack "ground-truth" labels. So, we use two simple and well-motivated observations about how asymptomatic CDI cases may relate to observed CDI cases in order to train our model.

Asymptomatic colonization has risk factors such as having CDI previously, antibiotic exposure, and hospital stay [27]; these are also well known to be risk factors associated with symptomatic CDI [12–14]. This leads to the observation that asymptomatic CDI cases and observed CDI cases may have similar risk profiles, which implies that we can train our model using observed CDI cases as instance labels. Then, patients who are assigned a high probability by a model trained in this manner, but are not CDI cases, are inferred to be asymptomatic CDI cases. Variants of this model can be obtained by using different subsets of features. More specifically, we partition the set of features into three groups: (i) *baseline* feature set B, consisting of length of stay, age, gender, prior hospital visit within 60 days, and the use of gastric acid suppressors, (ii) *colonization pressure* feature set CP, consisting of different measures of exposure to other observed CDI cases, and (iii) *antibiotics* (*ABXs*) feature set ABX, consisting of the use of high-risk antibiotics. For each of the four subsets $S \subseteq \{CP, ABX\}$, we create a feature set $\{B\} \cup S$, and train a model on this feature set.

Mechanistic models for CDI (e.g., [43]) often attribute the transition from asymptomatic CDI to symptomatic CDI to the use of additional high-risk antibiotics. This leads to the observation that the mechanism for acquiring (symptomatic) CDI consists of the patient first being an asymptomatic CDI case and then being prescribed high-risk antibiotics. This observation has the following useful implication. Suppose A is the subset of patients who were prescribed high-risk antibiotics during their visit. Then, the subset $A_{CDI} \subseteq A$, consisting of patients who tested positive for CDI is exactly identical to the subset of A of patients who were asymptomatic CDI cases (prior to receiving antibiotics) and $A \setminus A_{CDI}$ is exactly the subset of A of patients who are not asymptomatic CDI cases. Thus for the patients who were prescribed high-risk antibiotics during their visit, the "observed case" label corresponds exactly to the "asymptomatic case" label and we can train our models on this subset of the data. Again, we train four models by considering different subsets of features in addition to the baseline set of features.

The two simple observations are quite powerful in that they allow us to train different asymptomatic CDI case prediction models that we can then evaluate. Using the trained models, we can obtain, for each non-terminal node (x, i) in the time-expanded network G_S, a probability $W_v(x, i)$ of node (x, i) being an asymptomatic CDI case. These probabilities serve as node weights of the time-expanded network G_S provided as input to DIRECTED PCST.

6 Experiments

We now present an extensive evaluation of the accuracy and efficiency of our proposed methods on a large-scale synthetic data. We also leverage our approach for an important application, (symptomatic) CDI case prediction, on real hospital operations data described in Sect. 4. Our code and synthetic data are available for academic purposes.[1] All of our experiments were conducted on an Intel(R) Xeon(R) machine with 528GB memory.

Baselines: Since this is the first work on detecting asymptomatic cases in a temporal network that takes individual risks into account, there are no directly comparable methods. However, we compare the performance of our approach against the following natural baselines and state-of-the-art approach for a closely related task.

- **Frontier:** Nodes neighboring the known symptomatic cases could potentially be carriers. This method selects the neighbors of the terminal nodes as asymptomatic cases. Precisely, we mark a node as asymptomatic in the time-expanded network G_S if it has a directed edge to a terminal node.
- **Contact top** k: People with frequent contacts with others are likely to be exposed to infectious pathogens. This method selects top $k\%$, for $k \in \{5, 10, 15\}$, high-contact nodes based on the out-degree in G_S. We explore k around 10, based on studies that claim that around 10% of admitted patients were asymptomatic C. diff carriers [21, 22].
- **LOS top** k: As length of stay (LOS) of patients in the hospital increases, there is a higher chance for the patient to contract infectious agents. For example, LOS is known to be a risk factor for HAIs [13, 14]. Here, we select top $k\%$, for $k \in \{5, 10, 15\}$, nodes based on the LOS.
- **CuLT:** This is the state-of-the-art Steiner-tree-based missing infection detection approach [33]. Note that algorithms that *CuLT* uses are just a special case of our *Greedy* approaches, where there are no node weights.

6.1 Performance on the synthetic data

We perform a series of experiments on the synthetic data described in Sect. 4.2 to evaluate the performance of our algorithms. We start with a subset of the hospital data restricted to 1 month (Jan 2010) and first derive a temporal network (as described in Sect. 4.1) from this data and then a time-expanded network from this temporal network (as described in Sect. 2). This yields a time-expanded network with more than 20.9K nodes and 0.5M edges. We then run the biased SIS simulation described in Sect. 4.2 on this time-expanded network, starting from multiple sources, to obtain a set of observed symptomatic cases $\mathcal{S}_{GT}$ and a set of asymptomatic cases $\mathcal{A}_{GT}$ as described in Sect. 4.2. Note that the point of using synthetic data is that it provides us with "ground truth" on asymptomatic cases.

6.1.1 Comparison with baselines

The first experiment we conduct is designed to measure effectiveness of our approaches as compared to the baselines. We measure success for method m based on the overlap of the asymptomatic cases $\mathcal{A}_m$ it infers and the ground truth $\mathcal{A}_{GT}$. We use *Matthews correlation coefficient* (MCC) [26] and F_1-score as metrics to quantify success of the methods we evaluate.

[1] https://github.com/HankyuJang/directed-PCST-asymptomatic-detection.

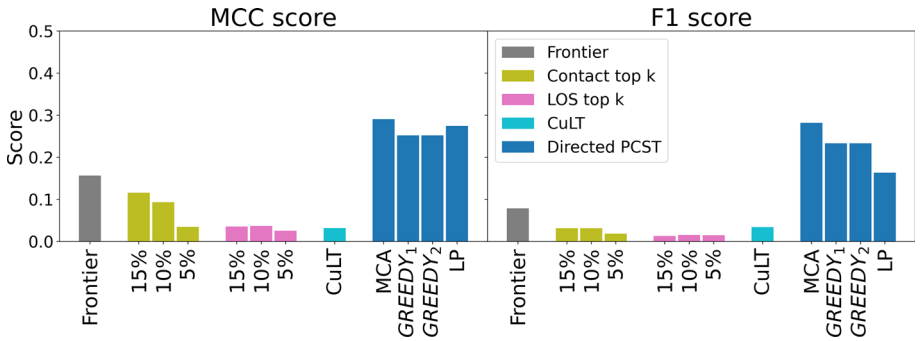


Fig. 4 Performance of all the methods in the synthetic data as measured by MCC (left) and F_1-score (right). All of our proposed methods MCA, GREEDY$_1$, GREEDY$_2$, and LP (in blue) comprehensively outperform the baselines

Note that we tune hyper-parameters including α, β, and γ for each method and report the best performance. The final result is presented in Fig. 4. As shown in the figure, our proposed approaches significantly outperform all the baselines in terms of both MCC and F_1-score. The result implies that our approaches recover as many ground-truth asymptomatic cases as any method, while maintaining a very high accuracy. The high margin of the discrepancy between the performance of our approaches and the baselines could be attributed to the fact that our approach finds the right balance between the likelihood of node being exposed to the disease (via edge weights) and the likelihood node developing symptoms (via node weights). The superiority of our approach over *CuLT* highlights the importance of taking individual risks into account in detecting asymptomatic cases. This is a key takeaway from our results.

6.1.2 The effect of α on the performance

Next, we study the effect of the parameter α on the performance. Note that the smaller values of α give higher weight to the edge costs, while the larger values give higher importance to the node weights, forcing our algorithms to pick nodes with higher probabilities. In this experiment, we quantify the effect of varying α on the performance. To this end, we run our approaches on the synthetic graph and obtain a set of recovered asymptomatic cases for $\alpha = 0$. We then repeat this process for different values of α. We compute MCC and the F_1-score for the inferred asymptomatic cases for each value of α. The result is presented in Fig. 5. We observe that for $\alpha = 0$, when the node probabilities have no effect on the solution, the performance is poor (as expected). On the other hand, for positive values of α, the performance is much better for all our methods. An immediate takeaway from this result is that individual risks are an important aspect of disease-spread. As we further increase α to values greater than 1, for the three methods that return an integral solution, there is a slight but gradual degradation in performance as more and more emphasis is placed on the node weights. The LP-based solution is much more sensitive to α and degrades relatively quickly as α increases. We suspect that the flexibility of being able to return a fractional solution allows the LP to disregard the constraint (solution to be a tree) and this degrades performance as α increases and there are more negative weight edges in the underlying graph. This experiment demonstrates that for small positive values of α where the importance of node weights and edge weights are balanced, we achieve the best performance, highlighting the importance of incorporating individual risks in detecting asymptomatic cases.

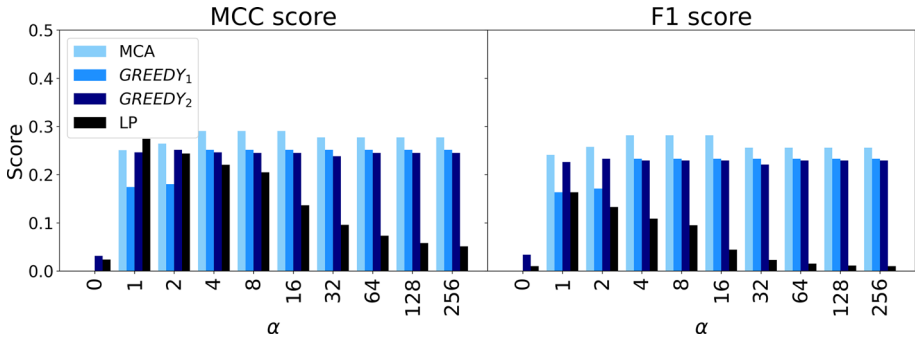


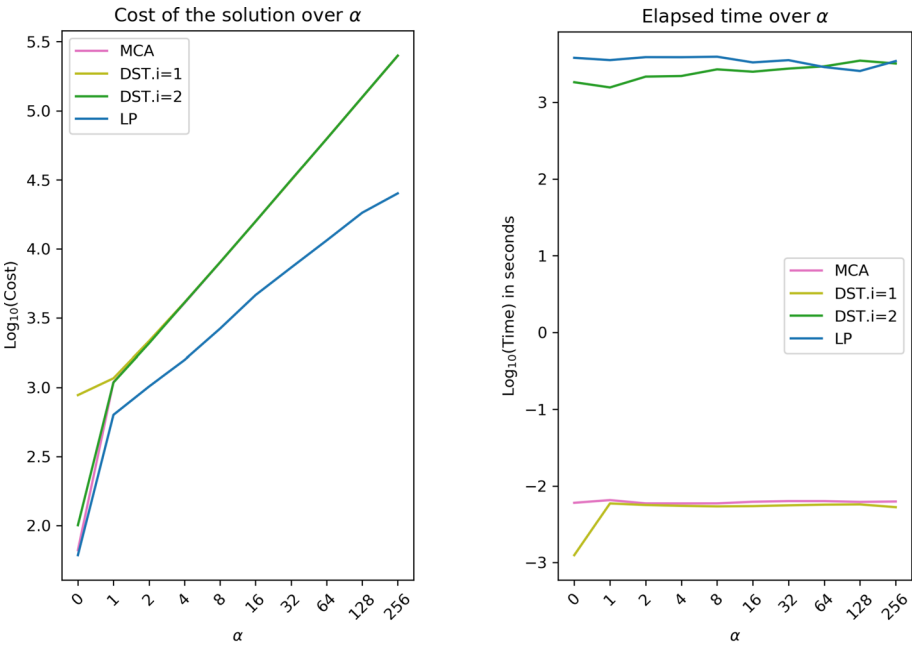
Fig. 5 The effect of varying α on the performance measured by MCC (left) and F_1-score (right).We see a sharp increase in the performance from $\alpha = 0$ to $\alpha = 1$, followed by a minor increase, and then a gradual decrease as α increases

6.2 Scalability and accuracy trade-off

As a step toward running our experiments on a larger time-expanded network, we next study the trade-off between scalability and the solution cost (summation of edge weights) of our proposed approaches on the synthetic data. In Fig. 6, we summarize the performance, in terms of the objective function cost our algorithms are minimizing and the time (in seconds) for each of our four methods. As expected, the optimal LP solution achieves the lowest cost, since its solution cost is guaranteed to be a lower bound on the cost of any integral directed Steiner tree (see Theorem 1). A pleasant surprise is that MCA returns a solution that is just a little bit larger than the LP solution for $\alpha = 0$, implying that the MCA returns a near-optimal solution. For larger α, MCA produces slightly cheaper trees than GREEDY$_1$ and GREEDY$_2$. The cost of the tree returned by GREEDY$_2$ is reasonable (within two times OPT for $\alpha = 0$), while GREEDY$_1$ returns a tree with relatively larger cost for $\alpha = 0$. With regard to running time, the LP-based solution has a large running time making it unscalable to large graphs. This is because even though an LP can be solved in polynomial time, the size of the flow-based LP (see Sect. 3.3) is much larger than the size of the underlying graph because there is a flow variable $f_{s,e}$ for every edge e and every terminal s. GREEDY$_2$ also has a high running time, which is too expensive for large graphs. On the other hand, our other two heuristics GREEDY$_1$ and MCA take a fraction of a second to compute the solution. Note that the MCA algorithm was made possible because we ensured that the time-expanded graph is a DAG.

7 Application: CDI case prediction

Next, we apply our method for asymptomatic case detection to the important task of predicting symptomatic CDI cases on actual hospital data (Fig. 7). Specifically, we use a measure of exposure to detected asymptomatic CDI cases as an additional feature in a CDI prediction model. Since we do have "ground-truth" CDI cases in our hospital data, we are able to compare our approach to other proposed methods for CDI prediction. Predicting CDI cases early is important task for many clinical reasons. For example, it can be used for doing early and more targeted testing of patients and initiating additional cleaning procedures at targeted locations so as to reduce CDI spread.



(a) The cost of the solution over α **(b)** Elapsed time of the solution over α

Fig. 6 The cost of the solution and the elapsed time in seconds for each method over α. Each value to its corresponding α is averaged over the space of β and γ. **a** The cost of MCA is near optimal when $\alpha = 0$, and MCA produces slightly cheaper trees than GREEDY$_1$ and GREEDY$_2$ across α. **b** MCA and GREEDY$_1$ take a fraction of a second to compute solution

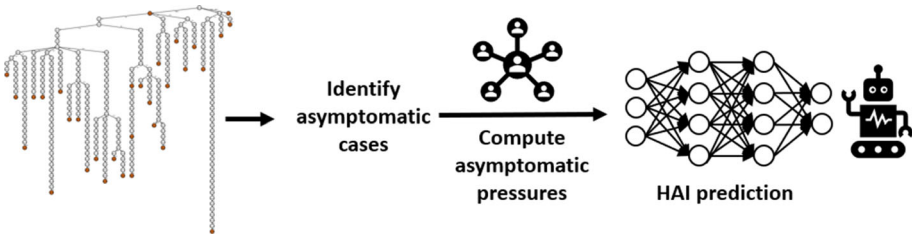


Fig. 7 This schematic shows our approach to applying the solution from ASYMPTOMATIC CASE DETECTION problem toward HAI case prediction

We expand the 1-month time window to 3 months and obtain a time-expanded network with more than 60.9K nodes and 1.6M edges (see Table 1 for the statistics of the network). Given the size of the this graph, and given our results on the cost–time trade-off, we only use GREEDY$_1$ and MCA as algorithms for DIRECTED PCST.

Here, we first run all the methods to infer asymptomatic cases in the 3-month time-expanded network. Then, we leverage the inferred asymptomatic cases to predict the symptomatic CDI cases. To do so, we train a neural network with two types of data: (i) standard risk factors for CDI (e.g., [14]) and (ii) additional features, that we call *asymptomatic pressures*, that measure the exposure to the newly identified asymptomatic CDI cases. Note that the additional features in (ii) are generated from the solutions of our approaches and

Table 1 Network statistics of the time-expanded network

Statistics	G_S (1 month)	G_S (3 months)
$\|V\|$	20,948	60,903
$\|E\|$	546,743	1,656,866
k_{in} (mean; s.t.dev; max)	26.1; 16.5; 117	27.2; 16.3; 116
k_{out} (mean; s.t.dev; max)	26.1; 145.5; 20947	27.2; 247.2; 60902

[a] k_{in} and k_{out} denote in-degree and out-degree, respectively

the baselines. For each method, we investigate whether adding exposure features to asymptomatic CDI cases improves performance of the neural model in predicting symptomatic cases. We define four exposure measures to asymptomatic C. diff carriers as the following:

- Unit sum asymptomatic C. diff pressure (SAP_{unit}): cumulative daily exposure to asymptomatic C. diff carriers in the same unit from admission date up to the date of the instance
- Room sum asymptomatic C. diff pressure (SAP_{room}): cumulative daily exposure to asymptomatic C. diff carriers in the same room from admission date up to the date of the instance
- Unit mean asymptomatic C. diff pressure (MAP_{unit}): $\frac{SAP_{unit}}{LOS}$
- Room mean asymptomatic C. diff pressure (MAP_{room}): $\frac{SAP_{room}}{LOS}$

then set the *asymptomatic pressures* as the vector of these exposure measures $[SAP_{unit}, SAP_{room}, MAP_{unit}, MAP_{room}]$.

Since we have the ground-truth symptomatic cases, we use the area under ROC curve (AUC) to quantify the effectiveness of each method of predicting symptomatic cases. We split the data temporally into equally sized training and test sets. We further split training set into training (80%) and validation (20%) sets. We train Multi-layer perceptron model (MLP. 2-layer; 16 neurons in the hidden layer; ReLU activation; drop out 0.5; Adam optimizer; learning rate 0.01; 200 epochs; early stopping if validation loss stop decreasing for three consecutive epochs). We repeat this process five times (though small repetition, we observe low s.t.dev.) and then report the mean AUC score on the test set.

Since the methods differ from each other only in asymptomatic pressures, their performance on symptomatic case prediction can be viewed as a proxy measure of how accurate the choice of asymptomatic cases was.

Figure 8 shows the CDI case detection results. The horizontal dashed line is the baseline performance that do not use asymptomatic pressures as features. One would expect adding extra information regarding exposure to the asymptomatic cases would only improve the performance. Hence, we interpret a method to have detected potential asymptomatic carriers correctly, if the performance of the symptomatic cases classification surpasses the dashed line after adding additional features measuring the exposure to inferred asymptomatic cases. We note that all the natural baselines described earlier actually deteriorate the performance, as these baselines are not able to identify the asymptomatic carriers correctly. Furthermore, we compare our method against two alternative "extremes" for identifying asymptomatic cases: (i) *CuLT*: which uses a low-cost directed Steiner tree, while ignoring node weights completely and (ii) *Features only*: which uses node weights representing individual risks, while ignoring edges completely. For the *CuLT* method, adding asymptomatic pressures degrades performance relative to the baseline. The *Features only* method is helped by the use of asymptomatic pressures, but not as much as our method. The results in Fig. 8 show that our proposed approach via the DIRECTED PCST problem (in blue) performs better than

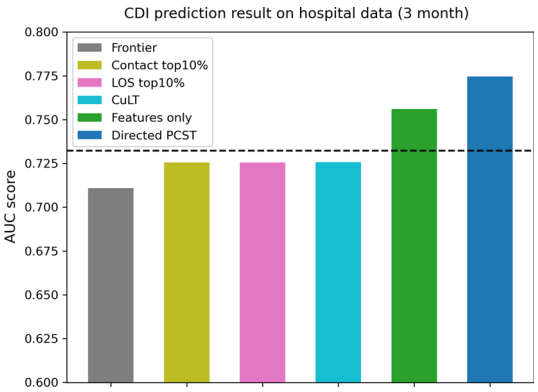


Fig. 8 Performance of the CDI prediction task measured by AUC. Our proposed approach DIRECTED PCST (in blue) outperforms the other methods as well as not using asymptomatic pressure features (dashed line) (color figure online)

all the baselines and improves over the horizontal dashed line in terms of the AUC on the symptomatic cases prediction task. These results indicate that our approach is indeed able to infer likely asymptomatic cases in real outbreaks even when the "ground-truth" data on asymptomatic cases are not available. The success in this task serves as a further, though indirect, evidence of the fact that our approach detects asymptomatic cases accurately.

8 Application: source detection

As demonstrated by the results presented earlier, our approaches outperform the baselines in detecting asymptomatic cases, implying that the cascades reconstructed by our approaches are more accurate. In this section, we leverage the reconstructed cascades to detect the sources (patient zeroes) of HAI outbreaks. We hypothesize that since our approach leads to more accurate cascades, it will also lead to more accurate sources.

As in the problem of detecting asymptomatic cases, we do not have "ground-truth" sources of real world HAI outbreaks. Hence, we perform our experiments on the synthetic data described in Sect. 4.2. To this end, we first prepare the time-expanded network G_S as described in Sect. 2. As mentioned earlier, the resulting network has more than 20.9K nodes and 0.5M edges. We then run the biased SIS simulation on G_S starting from multiple seeds SEED_{GT} in a manner similar to the one described in Sect. 4.2. We ensure that the seeds SEED_{GT} are selected from the first 10 days. As a result of the simulation, we obtain a set of observed symptomatic cases $\mathcal{S}_{GT}$. The source detection problem asks us to infer the patient zeroes, SEED_{GT}, given the underlying contact network G_S and the set of observed symptomatic infections $\mathcal{S}_{GT}$.

We run our algorithms and baselines to obtain the directed prize-collecting Steiner tree representing the reconstructed cascade. Note that our solution tree is actually a forest with all individual trees connected to the dummy root node. For our approach (and for the Steiner tree-based baseline *CuLT*), we interpret the nodes directly connected to the dummy root node as the sources SEED_m detected by our approach. Note that the nodes in SEED_m are "roots" for the individual trees. The contact top k and LOS top k baselines are defined in a manner similar to the one in Sect. 6. Frontier baseline is defined to be a set of nodes 1 day prior to having contact with the terminal nodes. The only additional constraint we pose on these baselines is that the sources have to be selected from the first 10 days. Finally, for the features only baseline, we just run our approaches with a large node weight of $\alpha = 1M$.

In this particular application, we discard the time-stamp of the nodes identified to be the sources and compute success on based only on the node id. We leave the problem of

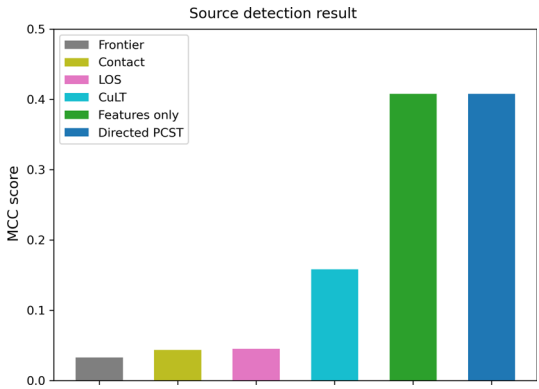


Fig. 9 Performance of the source detection task measured by MCC. Our proposed approach DIRECTED PCST (in blue) as well as the features only method that uses $\alpha = 1M$ (in green) outperforms the other methods in retrieving the ground-truth source nodes that initiated the infection cascades (color figure online)

detecting both the ids and the time-stamp of the sources to future work. For each method m, we measure the overlap between the sources SEED_m it detects and the hidden ground-truth sources SEED_{GT}. We quantify the overlap between SEED_m and SEED_{GT}, and we compute *Matthews correlation coefficient* (MCC) between the two. We tune hyper-parameters including α, β, and γ for each method and report the best performance. We repeat the process for all four variants of our approach and report the best result.

Figure 9 summarizes the performance of all the methods. The key take away from the plot is that our approach (in blue) outperforms all the baselines. The surprisingly competitive performance of the *features only* baseline is explained by the fact the sources too tend to be "high-risk" nodes. The fact that our approach significantly outperforms all other baselines reinforces our prior belief that cascades with accurate asymptomatic infections lead to more accurate sources as well. It is not surprising that the simple heuristics, *Frontier*, *Contact top k*, and *LOS top k*, have the poorest performance as in previous experiments. On the other hand, *CuLT* is much competitive in source detection task than in asymptomatic case detection task (Fig. 4). This observation alludes that a common ancestor connecting the observed symptomatic cases $\mathcal{S}_{GT}$ via the most likely transmission path is a reasonable guess for the source. However, as shown by our approach, a much better performance can be achieved by taking individual risks into account.

9 Case studies

We perform case studies on the UIHC data to demonstrate that the asymptomatic cases inferred by our algorithms are clinically meaningful. Because it is faster than our other algorithms while producing a solution of comparable quality, we use the MCA algorithm for our experiments. We use parameters $\alpha = 2$, $\beta = 2$, and $\gamma = 0$. Figure 10 shows the solution tree returned by MCA for the DIRECTED PCST problem. There were 97 CDI cases in the period of 3 months and our solution partitioned these into 38 outbreaks. One of these outbreaks is the "giant" outbreak, shown in Fig. 10 as emanating from the leftmost child of the "dummy" root. There are four minor outbreaks, also shown in Fig. 10. The remaining 33 outbreaks are just isolated cases and not depicted in the figure. In the figure, the intermediate nodes connecting the terminals to the root are inferred to be the asymptomatic cases.

Upon exploring the data in further detail, we discovered an inferred asymptomatic case (node in blue in the highlighted sub-tree) who had visited the UIHC for a major surgery for a disease unrelated to CDI. This patient was transferred into the UIHC from an acute-

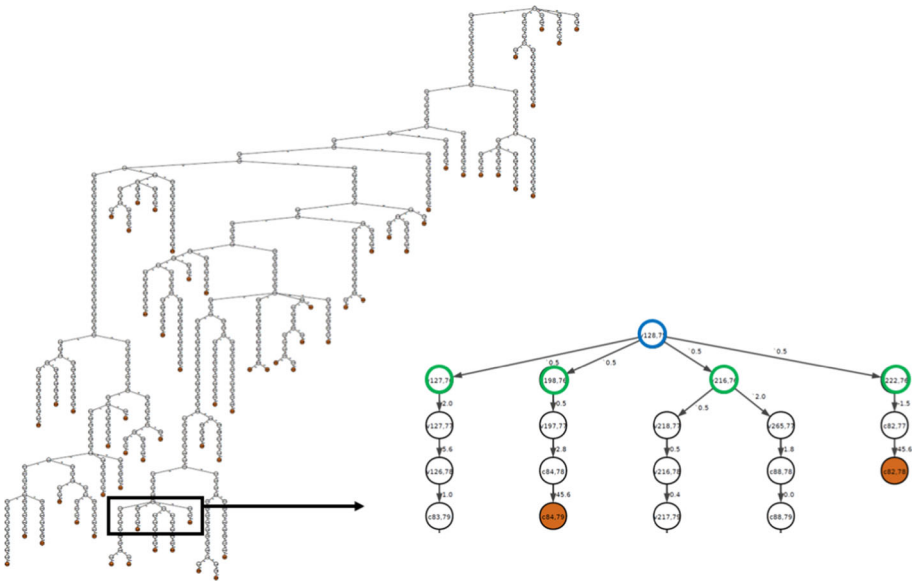


Fig. 10 A solution tree obtained by running DIRECTED PCST algorithms on UIHC data. Nodes in orange denote terminals. The highlighted sub-tree shows an inferred asymptomatic case (blue node) possibly infecting its four child nodes (color figure online)

care hospital providing inpatient medical care. Since exposure to healthcare settings is an important risk factor for CDI, it is likely that this asymptomatic patient was exposed to C. diff there prior to transferring to the UIHC. About two weeks later, in the afternoon, this patient was transferred to a large room with roughly 20 beds, stayed there for an hour, and then transferred again to another room. Surprisingly, all the four child nodes (nodes in green) had visited the same patient room on that day. Two of them had overlapped in time in the room with the blue node, and the remaining two patients had visited the patient room later in the afternoon. These four child nodes may have contracted C. diff in that patient room.

In Fig. 11, we focus on another inferred asymptomatic case (node in blue in the highlighted portion of the tree) that may have spread the pathogen to its four child nodes (nodes in green). This asymptomatic patient was a newborn baby in the hospital with extreme mortality and severity status and was treated in the pediatrics ICU. We suspect this patient was exposed to C. diff there. After about a month, the baby was transferred to the operating room (OR) early in the morning and then transferred to the neonatal intensive care unit (NICU) in the afternoon. On the same day, another baby (one of the green nodes) had close contact with the green node. She was transferred to OR in the morning and then to NICU in the afternoon. The rest of the child nodes also had visited the OR. These child nodes may have contracted C. diff while in the same unit (OR or NICU) with the blue node.

Next, we perform a case study with respect to the parameter γ. Figure 12 shows the effect of γ on the solution tree of the DIRECTED PCST problem on the hospital data. We can visually see that using smaller value of γ gives the solution more freedom branch out. The top most tree ($\gamma = 0$) detects 33 CDI cases and 5 asymptomatic cases coming from community; the middle tree ($\gamma = 16$) detects 19 CDI cases and 1 asymptomatic case from community; and the bottom most tree ($\gamma = 128$) detects 17 community CDI cases. Most of these community cases were transferred from the hospital emergency room, an outside acute hospital, or from

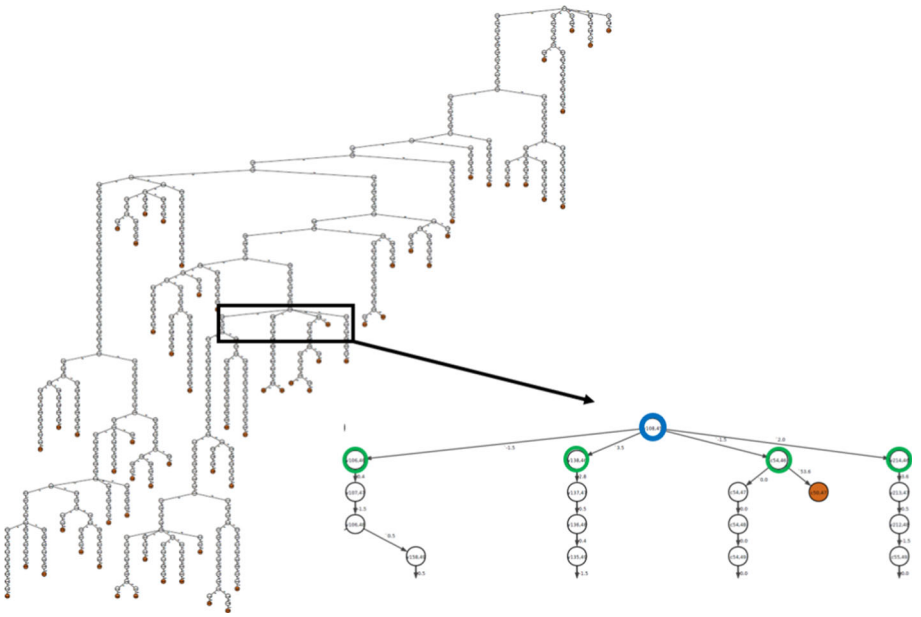


Fig. 11 A solution tree obtained by running DIRECTED PCST algorithms on UIHC data. Nodes in orange denote terminals. The highlighted portion of the solution tree shows an inferred asymptomatic case (blue node), possibly infecting its four child nodes (color figure online)

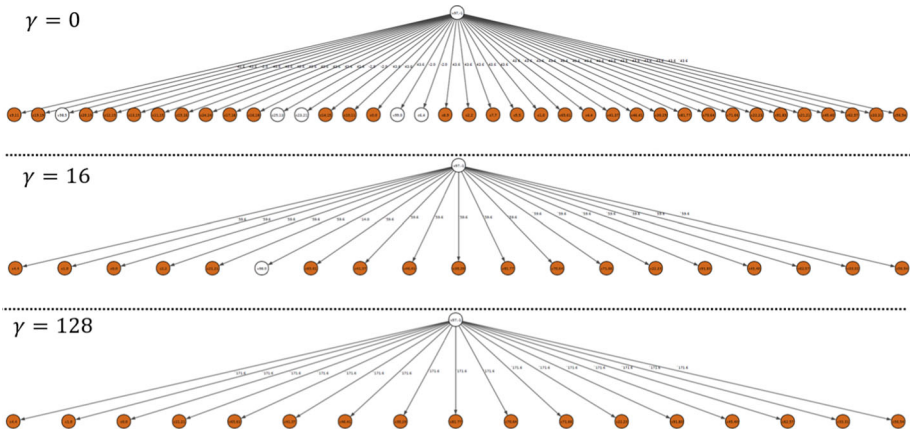


Fig. 12 Effect of γ on the solution trees of the DIRECTED PCST problem on UIHC data. Larger values of γ lead to solutions with fewer infection cascades. Fewer infection cascades imply more transmission within the hospital and fewer cases imported from the community

within the hospital clinic. Many of these patients had major to extreme mortality risk during their visit, where many of them were serviced in the internal medicine department.

10 Related work

In this section, we survey research in areas closely related to this work.

Epidemics over temporal network: Several approaches have outbreak detection [1, 30], infection prediction [25], disease modeling control [19], and epidemic surveillance [4] in temporal networks. See [25] for a survey of existing works in this space.

Symptomatic Cases Prediction: A line of related works have developed AI/data mining tools to predict symptomatic cases in both microscopic and macroscopic scales. The microscopic prediction focuses on predicting the individual nodes/people to be infected in near future. Some common approaches leveraged in the past include topological LSTM [38], temporal point processes [18], and random walk based embeddings [23]. On the other hand, work focusing on the macroscopic level predicts the total number of symptomatic cases in a large geographic area. Some common approaches include deep learning [2, 31], data assimilation [34], and empirical Bayesian approach [5]. A slightly related field vies to infer missing infections at a macroscopic level [8, 9, 39].

Cascade reconstruction over time: There has been much interest in reconstructing epidemic outbreaks over time. Farajtabar et al. [15] use two-stage framework that learns the diffusion model and identifies the source that maximizes the likelihood of observing the cascade as per the learned diffusion model, as a solution to the source identification which is a special case of missing infection problem. Sundareisan et al. [35] propose Netfill that recovers missing infections under SI model given snapshots of infections over time, using minimum description length principle [35]. There is also some work that leverages Steiner trees to infer missing infections. Rozenshtein et al. [33] use a Steiner tree-based approach to reconstruct epidemic cascades on a streaming contact network for SI-like model. Xiao et al. [42] solve a related problem of inferring missing infections in static networks and in a different paper. Xiao et al. [41] propose a sampling based approach for a robust cascade reconstruction. None of these approaches incorporate individual risks. Note that our problem of detecting asymptomatic cases is inherently different from the problem of inferring missing infections because individual attributes play a significant role in determining whether or not an individual is asymptomatically colonized. Makar et al. [24] present a latent state modeling approach to detect asymptomatic carriers. However, they assume that the underlying network is static and the disease does not spread through a chain of infections. In this paper, we solve the problem in a more general setting, where the underlying network is dynamic and disease spreads through a chain of asymptomatic cases.

11 Conclusion

This paper studies the problem of detecting asymptomatic cases in a temporal network in which outbreaks have occurred. We show that taking into account both individual risk and the likelihood of disease-flow along edges leads to improved detection. We formulate the asymptomatic case detection problem as a directed prize-collecting Steiner tree problem and show an approximation-preserving reduction from this problem to the directed Steiner tree problem and then use this reduction to obtain scalable algorithms for the directed prize-collecting Steiner tree problem. We then solve large instances of this problem on both synthetic data and actual hospital data and demonstrate that our detection methods outperform various baselines, including baselines that ignore either the individual risk or edge characteristics. We also demonstrate that the solutions returned by our approach are clinically meaningful by conducting several case studies.

We aim to take this work in two different directions. In an algorithmic direction, we plan on extending our approach to more general models of HAI spread, e.g., the pathogen load transfer model of [19]. In such models, cascades are no longer just trees and so we would have to optimize over more complicated cascade structures. In an applied direction, we aim to implement our methods in actual clinical settings as a way of aiding in low-cost surveillance by identifying patients or locations that need additional monitoring. Specifically, we would use our approach in real time to identify high-risk patients or locations (e.g., patient room, nurses' station) for additional surveillance.

Acknowledgements This project was funded by CDC MInD Healthcare Network grants U01CK000531 and U01CK000594 and NSF Grant 1955939. The authors acknowledge feedback from other University of Iowa CompEpi group members. This paper is an extended version of work published in ICDM 2021 [20]. The authors thank the anonymous ICDM 2021 reviewers for providing valuable feedback.

References

1. Adhikari B, Lewis B, Vullikanti A, Jiménez JM, Prakash BA (2019) Fast and near-optimal monitoring for healthcare acquired infection outbreaks. PLoS Comp Bio 15(9):e1007284
2. Adhikari B, Xu X, Ramakrishnan N and Prakash BA (2019) Epideep: exploiting embeddings for epidemic forecasting, In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 577–586
3. Archer A, Bateni M, Hajiaghayi M, Karloff H (2011) Improved approximation algorithms for prize-collecting Steiner tree and TSP. SICOMP 40(2):309–332
4. Bai Y, Yang B, Lin L, Herrera JL, Du Z, Holme P (2017) Optimizing sentinel surveillance in temporal network epidemiology. Sci Rep 7(1):1–10
5. Brooks LC, Farrow DC, Hyun S, Tibshirani RJ, Rosenfeld R (2015) Flexible modeling of epidemics with an empirical bayes framework. PLoS Comput Biol 11(8):e1004382
6. Buitrago-Garcia DC, Egli-Gany D, Counotte MJ, Hossmann S, Imeri H, Ipekci AM, Salanti G, Low N (2020) The role of asymptomatic sars-cov-2 infections: rapid living systematic review and meta-analysis. medRxiv
7. Charikar M, Chekuri C, Cheung T-Y, Dai Z, Goel A, Guha S, Li M (1999) Approximation algorithms for directed steiner problems. J Algorithms 33(1):73–91
8. Childs ML, Kain MP, Harris MJ, Kirk D, Couper L, Nova N, Delwel I, Ritchie J, Becker AD, Mordecai EA (1957) (2021), The impact of long-term non-pharmaceutical interventions on covid-19 epidemic dynamics and control: The value and limitations of early models. Proc R Soc B 288:20210811
9. Cui J, Haddadan A, Haque A A-U, Adhikari B, Vullikanti A, Prakash BA (2021) Information theoretic model selection for accurately estimating unreported covid-19 infections. medRxiv
10. Curtis D, Hlady C, Kanade G, Pemmaraju S, Polgreen P, Segre A (2013) Healthcare worker contact networks and the prevention of hospital-acquired infections. PLOS One 8(12):e79906
11. Curtis D, Hlady C, Pemmaraju S, Polgreen P, Segre A (2010) Modeling and estimating the spatial distribution of healthcare workers. In: 1st ACM International Conference on Health Informatics
12. Dubberke ER, Reske KA, Olsen MA, McMullen KM, Mayfield JL, McDonald LC, Fraser VJ (2007) Evaluation of clostridium difficile-associated disease pressure as a risk factor for c difficile-associated disease. Arch Int Med 167(10):1092–7
13. Dubberke ER, Reske KA, Seiler S, Hink T, Kwon JH, Burnham C-AD (2015) Risk factors for acquisition and loss of clostridium difficile colonization in hospitalized patients. Antimicrob Agents Chemother 59(8):4533–43
14. Dubberke ER, Yan Y, Reske KA, Butler AM, Doherty J, Pham V, Fraser VJ (2011) Development and validation of a clostridium difficile infection risk prediction model. ICHE 32(4):360–366
15. Farajtabar M, Rodriguez MG, Zamani M, Du N, Zha H, Song L (2015) Back to the past: Source identification in diffusion networks from partially observed cascades. AISTATS
16. Gabow HN, Galil Z, Spencer T, Tarjan RE (1986) Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. Combinatorica 6(2):109–122
17. Halperin E, Krauthgamer R (2003) Polylogarithmic inapproximability. STOC, pp 585–594

18. Islam MR, Muthiah S, Adhikari B, Prakash BA, Ramakrishnan N (2018) Deepdiffuse: predicting the'who'and'when'in cascades. In: 2018 IEEE International Conference on Data Mining (ICDM)', IEEE, pp 1055–1060
19. Jang H, Justice S, Polgreen PM, Segre AM, Sewell DK, Pemmaraju SV (2019) Evaluating architectural changes to alter pathogen dynamics in a dialysis unit. ASONAM
20. Jang H, Pai S, Adhikari B, Pemmaraju SV (2021) Risk-aware temporal cascade reconstruction to detect asymptomatic cases: For the cdc mind healthcare network. In: 2021 IEEE International Conference on Data Mining (ICDM), IEEE, pp 240–249
21. Kyne L, Warny M, Qamar A, Kelly CP (2000) Asymptomatic carriage of clostridium difficile and serum levels of igg antibody against toxin a. NEJM 342(6):390–397
22. Leekha S, Aronhalt KC, Sloan LM, Patel R, Orenstein R (2013) Asymptomatic clostridium difficile colonization in a tertiary care hospital: admission prevalence and risk factors. Am J Infect Control 41(5):390–393
23. Li C, Ma J, Guo X, Mei Q (2017) Deepcas: an end-to-end predictor of information cascades. In: Proceedings of the 26th international conference on World Wide Web, pp 577–586
24. Makar M, Guttag J, Wiens J (2018) Learning the probability of activation in the presence of latent spreaders. AAAI, 32
25. Masuda N, Holme P (2013) Predicting and controlling infectious disease epidemics using temporal networks. F1000prime reports, 5
26. Matthews B (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochim Biophys Acta (BBA) - Protein Struct 405(2):442–451
27. Nissle K, Kopf D, Rösler A (2016) Asymptomatic and yet c. difficile-toxin positive? prevalence and risk factors of carriers of toxigenic clostridium difficile among geriatric in-patients. BMC Geriatr. https://doi.org/10.1186/s12877-016-0358-3
28. Potasman I (2017) Asymptomatic infections: the hidden epidemic. Int J Clin Res Trials 2:118
29. Raghavan P, Tompson CD (1987) Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica 7(4):365–74
30. Reis BY, Kohane IS, Mandl KD (2007) An epidemiological network model for disease outbreak detection. PLoS Med 4(6):e210
31. Rodriguez A, Tabassum A, Cui J, Xie J, Ho J, Agarwal P, Adhikari B, Prakash BA, (2020) Deepcovid: an operational deep learning-driven framework for explainable real-time covid-19 forecasting. medRxiv
32. Rothvoß T (2011) Directed steiner tree and the lasserre hierarchy. CoRR
33. Rozenshtein P, Gionis A, Prakash BA, Vreeken J (2016) Reconstructing an epidemic over time. ACM SIGKDD pp. 1835–1844
34. Shaman J, Kohn M (2009) Absolute humidity modulates influenza survival, transmission, and seasonality. Proc Nat Acad Sci 106(9):3243–3248
35. Sundareisan S, Vreeken J, Prakash BA (2015) Hidden hazards: finding missing nodes in large graph epidemics. SDM pp 415–423
36. Tuncbag N, Braunstein A, Pagnani A, Huang SS, Chayes J, Borgs C, Zecchina R, Fraenkel E (2013) Simultaneous reconstruction of multiple signaling pathways via the prize-collecting steiner forest problem. J Comput Biol 20(2):124–36
37. U.S. Department of Health and Human Services (Jan 15, 2020 (accessed June 10, 2020)) , Health Care-Associated Infections
38. Wang J, Zheng VW, Liu Z, Chang K C-C (2017) Topological recurrent neural network for diffusion prediction. In: 2017 IEEE International Conference on Data Mining (ICDM) IEEE, pp 475–484
39. Wilder B, Charpignon M, Killian JA, Ou H-C, Mate A, Jabbari S, Perrault A, Desai AN, Tambe M, Majumder MS (2020) Modeling between-population variation in covid-19 dynamics in hubei, lombardy, and new york city. Proc Nat Acad Sci 117(41):25904–25910
40. Worby CJ, Jeyaratnam D, Robotham JV, Kypraios T, O'Neill PD, De Angelis D, French G, Cooper BS (2013) Estimating the effectiveness of isolation and decolonization measures in reducing transmission of methicillin-resistant staphylococcus aureus in hospital general wards. AJE 177(11):1306–1313
41. Xiao H, Aslay C, Gionis A (2018) Robust cascade reconstruction by steiner tree sampling. ICDM pp 637–646
42. Xiao H, Rozenshtein P, Tatti N, Gionis A (2018) Reconstructing a cascade from temporal observations. SDM pp 666–674
43. Yakob L, Riley TV, Paterson DL, Clements AC (2013) Clostridium difficile exposure as an insidious source of infection in healthcare settings: an epidemiological model'. BMC Infect Dis 13(376):1–8
44. Zelikovsky A (1997) A series of approximation algorithms for the acyclic directed steiner tree problem. Algorithmica 18(1):99-110

Hankyu Jang is a PhD student in the Computer Science Department at the University of Iowa, where he is advised by Prof. Alberto Segre and Prof. Sriram Pemmaraju. He is a member of the Computational Epidemiology research group. His PhD research is computational modeling and inference in healthcare-associated infections. He has industrial work experience, such as Applied Scientist Intern at Amazon and Machine Learning and Data Science Intern at American Family Insurance. Before his PhD, he did MS in Data Science from Indiana University Bloomington. His research interest, in general, is in developing and applying data mining and deep learning methods on dynamic graphs.

Shreyas Pai is a postdoctoral researcher in the Theoretical Computer Science Group at Aalto University. He received his PhD in Computer Science at the University of Iowa under the supervision of Prof. Sriram Pemmaraju in May 2021. During his PhD, he worked as a research assistant in the Computational Epidemiology Group where we try to understand and model the spread of hospital-acquired infections. His research interests generally lie in theoretical computer science, more specifically in Distributed algorithms, communication complexity, and combinatorial optimization.

Bijaya Adhikari is an Assistant Professor in the Department of Computer Science at the University of Iowa. He is also a member of the Interdisciplinary Computational Epidemiology Research Group at the University of Iowa. His research focuses on network mining problems and their applications to computational epidemiology. His research has been published at top-tier data mining venues and interdisciplinary venues such as PLoS Computational Biology and PNAS. Some examples of his prior works include learning embeddings of infection cascades, dynamic embeddings of historical influenza seasons for epidemic forecasting, and deep representation learning for COVID-19 forecasting.

Sriram Pemmaraju is a Professor in the Department of Computer Science at the University of Iowa and a member of the Computational Epidemiology Research Group at Iowa. His primary research interests are in Theoretical Computer Science, but he is also interested in applying algorithmic techniques to computational epidemiology problems, especially resource allocation and inference problems for infections that spread in healthcare settings.