Finding the Most Likely Infection Path in Networks with Limited Information

David Rey[a,*], Lauren Gardner[a,b], S. Travis Waller[a,b]

[a]*School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW, 2052, Australia*
[b]*NICTA, Sydney, NSW, 2052, Australia*

Abstract

In this paper we address the problem of identifying the most likely infection pattern responsible for the spread of a disease in a network. In particular, we focus on the scenario where limited information (*i.e.* infection reports) is available during an ongoing outbreak. For this problem we propose a maximum likelihood model and present an integer programming formulation. The objective of the model is to identify the most likely directed tree that spans to a set of known infected nodes, subject to path feasibility constraints. We propose a new solution method based on a three step heuristic which (1) reduces the initial graph using a polynomial time algorithm designed to find a subset of feasible infection paths, (2) ranks these paths using lower and upper bounds on their likelihood and generates multiple subgraphs containing a variable level of information, and (3) solves an exact mixed integer linear programming reformulation of the maximum likelihood model. Simulated contagion episodes are used to evaluate the performance of our solution method. Our results show that the approach is computationaly efficient and is able to reconstruct a significant proportion of the outbreak, even in the context of low levels of information availability.

Keywords: network optimization, integer programming, contagion processes, length constrained shortest path, k shortest paths

1. Introduction

Infectious diseases pose an increasing risk to humans due to a growing world population, increasingly dense urban environments, and highly connected global transport systems which together provide the necessary groundwork for a global epidemic. Over the past decades copious research efforts have contributed to the development of contagion models for predicting the expected spreading behaviour of infectious diseases by exploiting population demographics, human travel patterns, social interactions and properties of the disease itself. These models are used to assess various disease prevention, intervention and response strategies. The same models however, are unable to reconstruct the contagion process of an ongoing outbreak in order to reveal the spatiotemporal transmission patterns within a given population. We address this gap in the literature with the development of an optimization based solution method designed to identify the most likely infection path of a disease in a network. In particular, we focus on the scenario in which only a limited amount of the infection-related information is available, *i.e.* we assume that the infection status of only a subset of the population is known, which is often the case in real-world outbreaks.

In this paper, we propose a new combinatorial approach to quantify the likelihood of possible disease transmission patterns in networks. Our solution method utilizes the network topology (*e.g.* nodes, links), estimated disease parameters (*e.g.* transmission probabilities, infectious period) and available infection reports (*e.g.* node status information, time of infection) to evaluate a region that has been exposed to infection. Our objective is to find the set of links spanning to all known infected nodes which maximizes the likelihood of the infection pattern. We formulate this maximum likelihood problem as an Integer Program (IP) and

*Corresponding author
Email addresses: d.rey@unsw.edu.au (David Rey), l.gardner@unsw.edu.au (Lauren Gardner), s.waller@unsw.edu.au (S. Travis Waller)

propose a tailored heuristic algorithm to reduce the solution search space. From a mathematical perspective, the IP considered in this study is closely related to the hop-constrained Steiner tree problem, however we address the case in which the entire tree is not constrained but specific Origin-Destination (OD) constraints arise therein. The solution method developed is based on a k best paths ranking strategy which attempts to find feasible infection paths between known infected nodes and then solve an exact reformulation of the IP into a Mixed-Integer Linear Program (MILP) on the subgraph induced by the collection of such paths.

The performance of the solution method is measured by quantifying its ability to accurately identify the observed infection pattern. The solution method is shown to be able to reveal the set of links and nodes most likely responsible for having spread the disease. Furthermore, the methodology can account for missing infection information, enabling epidemiologists to better understand and anticipate disease transmission patterns during an ongoing outbreak and offer insights into the success of outbreak control measures.

We start by reviewing the literature on the modelling of contagion processes and on the optimization problems that unfold in our approach (Section 2). We then introduce the epidemiological context of this research as well as the mathematical formulation of the maximum likelihood model (Section 3). The solution method is presented in three steps (Section 4): (1) the mathematical properties of feasible infection paths are assessed and a polynomial time subgraph generation algorithm is presented; (2) a path-ranking strategy is developed and (3) an exact MILP reformulation of the initial IP is proposed. Network topology and disease parameters are introduced in the validation framework (Section 5) and the results obtained after the implementation of our solution method are then presented (Section 6). Finally, the contributions of this research is discussed and summarized (Section 7).

2. State of the Art

In this section we review the literature on epidemic modelling and discuss the position of this work with regards to network optimization problems.

2.1. Contagion Processes Modelling

Dynamic contagion processes impact copious network systems, and are therefore the focus of various studies within the emerging field of network science. In addition to the transmission of infectious disease through communities and biological systems [2, 25], the spread of information, ideas and opinions via social networks can also be modelled as a contagion process [6, 20]; as well as the global spread of computer viruses on the Internet network [26, 3]; power grid failures in electricity markets [31, 23]; and the collapse of financial systems [34].

In efforts to predict expected disease spreading behavior and characteristics, epidemiological models span from extremely generalized and simplified analytical models to increasingly in-depth stochastic agent based simulation tools. Analytical models are used to quantify the statistical properties of epidemic patterns [35, 8]; however, they are unable to capture certain behavioral aspects of the dynamics of disease spreading, and often lack detailed information about the network structure. In contrast, agent based simulation models can be used to replicate possible spreading scenarios, predict average spreading behavior, and analyze various intervention strategies for a given network and disease while capturing a greater degree of detail, but in turn require a highly detailed set of input data [7, 9, 11, 10, 13, 28]. The most recent and comprehensive models provide a greater degree of realism, but are difficult to implement within the short time frames in which real time control decisions must be made. Large scale simulation models can also be computationally taxing because multiple runs are required to accurately predict expected outcomes.

There currently exists a gap in the literature which calls for scenario specific disease prediction models. Most contagion models predict future potential outbreak scenarios based on system-wide information; however, they are not able to reconstruct the contagion process of an ongoing outbreak to reveal information about the current state of the network. Recent advances in disease modeling have begun addressing this issue. For example, there are models which use genetic sequencing data to analytically infer the geographic history of a given virus's migration [9, 1, 38, 27]. Often this approach involves first enumerating all possible

evolutionary trees, then assigning posterior probabilities based on specifics of the respective virus' mutation rates. Additionally the infection trees only include locations where samples were available. Jombart et al. [22] proposed a novel approach to reconstruct the spatiotemporal dynamics of outbreaks from sequence data by inferring ancestries directly between strains of an outbreak using their genotype and collection date. The "infectious" links were selected such that the number of mutations between nodes is minimized. This study is motivated by the need to track viruses through space and time in order to aid in the implementation of real-time containment strategies. Often the required genetic data and mutation based statistical properties are unavailable, or impossible to gather within the required time-frame. The proposed approach relies instead on available infection reports, network topology and disease properties to infer the spatiotemporal path of infection through a network.

2.2. Related Network Optimization Problems

Using infection data to reconstruct the infection tree of a contagion process can be approached with mathematical optimization techniques. Indeed, identifying the most likely infection links responsible for the spread of a disease in a network is closelly related to finding the maximum cost spanning tree [19] to all infected individuals, where the cost on the links represents the likelihood of the link to have spread the disease. Gardner et al. [14] proposed an efficient solution method for this problem when all the infected individuals are known which is based on an optimum branching strategy. In the context of epidemic modelling, it is seldom that the status of all infected individuals is known, hence it is critical to address the scenario where only partial information on the population is available. This more general problem can be represented as a Steiner Tree Problem (STP) [21], where Steiner nodes represent individuals whose infection status is unknown, which is NP-Hard [15].

Infection patterns of disease spreading processes are dependent on the parameters of the disease (exposed and infectious periods, transmission probabilities), therefore feasibility challenges arise in the search for valid infection paths. As such, the reconstruction of contagion processes with limited information is structurally similar to a constrained STP. The ressource constrained STP has been introduced by Rosenwein and Wong [29]; in their formulation, the authors attribute a ressource for each link in the network and ask for the minimum cost Steiner tree such that the total amount of ressources used in the tree does not exceed a given threshold. The authors discuss the efficiency a Lagrangean relaxation versus a Lagrangean decomposition of the problem. Voss [37] focused on the hop-constrained STP, which can be seen as special case of the ressource constrained STP where every ressource is set to a unit cost, and presented a dynamic tabu search heuristic. Several new techniques have emerged to address the ressource-constrained and the diameter-constrained STP [17, 33, 18], in these formulations the ressource constraint is imposed on the entire tree. In contrast, we consider the scenario where the hop-distance (length) of every path originating from the root node to a leaf of the tree is constrained independently.

Our objective is to find the Most Likely Infection Tree (MLIT) that spans to all known infected nodes in a network where the available information (*e.g.* node status and time of infection) enforces hop-distance constraints on infection paths. Fajardo and Gardner [12] designed a heuristic approach to solve a relaxed version of the MLIT. We extend that line of research by introducing a new solution method. In the next section we present the epidemiological context of this study and the mathematical formulation of the maximum likelihood model.

3. Problem Formulation

We start by presenting the epidemiological context of this research. To reproduce the spread of infectious diseases in networks, we use a generic compartmental model. We next introduce the mathematical notation and define the *level of information* in the network before detailing the problem formulation.

3.1. Epidemiological Model

The Susceptible-Exposed-Infectious-Recovered (SEIR) model [2] is a well established stochastic simulation model used in the epidemiological literature to model the progress of an epidemic in a large population. The SEIR model considers a fixed population of individuals which can be broken into four compartments:

- S (Susceptible): the individuals are susceptible to the disease and have never been infected.
- E (Exposed): the individuals have been infected but are not yet contagious. We assume that individuals remain in this compartment for a period of L timesteps.
- I (Infectious): the individuals have been infected and are contagious. We assume that individuals remain in this compartment for a period of D timesteps.
- R (Recovered): the individuals have been infected and are now recovered or removed. These individuals are not capable of spreading the disease.

The flow of the SEIR model can then be represented as

$$S \to E \to I \to R \tag{3.1}$$

Both L and D are parameters in the model expressed in units of time and are assumed to be disease-specific. At each time step, every infectious individual attempts to infect its neighbors in the network. Note that the SEIR model imposes that individuals can only be infected by a single other individual, hence the topology of the infection pattern induced by this model is a *tree*. The maximum number of infection trials between any two individuals is bounded by parameter D which represents the infectious period. In this study, we assume that an individual cannot be infected and become infectious at the same time step. Our objective is to reconstruct the spreading pattern of an outbreak in a network where only limited information is available, *i.e.* where not all the infected individuals are known. This approach is motivated by the more realistic setting in which only a subset of infected individuals report to public health authorities, whether due to limited medical accessibility or asymptomatic cases.

3.2. Notation and Information Availability

We assume a network composed by a directed weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where every node $i \in \mathcal{N}$ represents an individual; every link $(i, j) \in \mathcal{A}$ is a relationship among two individuals and the weights on links correspond to disease transmission probabilities. Namely, p_{ij} is the probability that individual i infects j. Using the terminology of the SEIR compartmental model, the state (e.g. infected, recovered) of a subset of individuals $\mathcal{I} \subseteq \mathcal{N}$ is assumed known, and for the known infected individuals the time of infection (i.e. timestamp) is also assumed known. These nodes are hereby referred to as *information* nodes. In this study, we assume that time can be discretized and therefore the timestamps of individuals are represented by integers. If an individual $i \in \mathcal{N}$ has been infected by the disease (e.g. i is in state E,I or R), its timestamp is represented by an integer $T_i \in \mathbb{Z}$. In turn, if i has never been infected, we set $T_i \to \infty$. In contrast, any node in $\mathcal{N} \backslash \mathcal{I}$ may or may not have been infected, this subset is referred to as the set of *zero-information* nodes.

Let $\mathcal{I}_i \subseteq \mathcal{I}$ be the set of infected information nodes and let $\mathcal{I}_n \subseteq \mathcal{I}$ be the set of non-infected information nodes ($\mathcal{I}_i \cup \mathcal{I}_n = \mathcal{I}$). Let $t_i \in \mathbb{Z}$ be a decision variable representing the timestamp of individual $i \in \mathcal{N}$; the value of t_i depends on the available information, namely

$$\forall i \in \mathcal{N}, \qquad t_i \equiv \begin{cases} T_i \in \mathbb{Z} & \text{if } i \in \mathcal{I}_i \\ T_i \to \infty & \text{if } i \in \mathcal{I}_n \\ \in \mathbb{Z} & \text{otherwise} \end{cases} \tag{3.2}$$

Hence if $i \in \mathcal{N} \setminus \mathcal{I}$, t_i is represented by an integer decision variable. We assume that no zero-information node may have been infected before the earliest known individual infected – hereby referred to as the root node – or after than the latest known infected individual and we define

$$T_{min} \equiv \min_{i \in \mathcal{I}_i} \{T_i\} \quad \text{and} \quad T_{max} \equiv \max_{i \in \mathcal{I}_i} \{T_i\} \tag{3.3}$$

T_{min} and T_{max} can be used to restrict the domain of variables t_i given that node i is infected. We next introduce the probabilistic inference model used to estimate the likelihood of an infection tree.

3.3. Maximum Likelihood Model

To find the MLIT of an outbreak in a network, we seek to determine the probability that a node has been infected by its neighbor during the spread of the disease. Given the epidemiological context presented in Section 3.1, we define a feasible infection link as follows.

Definition 1 (Feasible Infection Link). Let $(i, j) \in \mathcal{A}$, be a link of the network. (i, j) is a feasible infection link if and only if

$$L \leq t_j - t_i \leq L + D - 1 \tag{3.4}$$

Equation (3.4) states that node j may have been infected by i only if their timestamp difference is greater than L or if it is lower than $L + D - 1$, which corresponds to an interval of $D - 1$ time steps. Recall that we assume that that a node cannot be infected and infect an adjacent node at the same time step, hence we assume that $L \geq 1$. Let p_{ij} be the link transmission probability from individual i to j; the probability α_{ij} that j is infected by i is then

$$\alpha_{ij} = p_{ij}(1 - p_{ij})^{(t_j - t_i - L)^+} \tag{3.5}$$

where $(x)^+ \equiv \max\{x, 0\}$. $(t_j - t_i - L)^+$ is the number of unsuccessfull trials between nodes i and j given that eventually one trial is successfull. To account for the event that a node is not infected during the outbreak, we introduce the associated probability γ_{ij} as

$$\gamma_{ij} = (1 - p_{ij})^{\min\{D, (t_j - t_i - L + 1)^+\}} \tag{3.6}$$

where $\min\{D, (t_j - t_i - L + 1)^+\}$ is the maximum number of unsuccessfull infection trials between nodes i and j. In order to account for both probabilities in the model, we combine α_{ij} and γ_{ij} in a single expression. As α_{ij} and γ_{ij} are complementary, that is, only one of these events can occur; we introduce a binary decision variable x_{ij} to model this relationship. Let $\mathcal{T}^\star$ be the optimal infection tree, we define

$$\forall (i, j) \in \mathcal{A}, \qquad x_{ij} \equiv \begin{cases} 1 & \text{if } (i, j) \in \mathcal{T}^\star \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

x_{ij} are the main decision variables in the model as they define the resulting infection tree $\mathcal{T}^\star$. We define the likelihood of the infection tree as the product of probabilities α_{ij} and γ_{ij} over all the links in the network. Our objective is to maximize this likelihood, hence we introduce the likelihood function $\lambda_{ij}(x_{ij})$ defined as

$$\lambda_{ij}(x_{ij}) \equiv \alpha_{ij}^{x_{ij}} \gamma_{ij}^{(1 - x_{ij})} = p_{ij}(1 - p_{ij})^{x_{ij}(t_j - t_i - L)^+} (1 - p_{ij})^{(1 - x_{ij}) \min\{D, (t_j - t_i - L + 1)^+\}} \tag{3.8}$$

Objective function (3.9) computes the likelihood of an infection tree by estimating the impact of successfull and unsuccessfull trials among adjacent nodes. Let $r \in \mathcal{I}$ be the root of the infection tree and let $\Gamma^+(i)$ and $\Gamma^-(i)$ be the sets of successors and predecessors of node $i \in \mathcal{N}$ in graph $\mathcal{G}$ respectively. To find the MLIT of an outbreak, we propose the following IP.

Model 1 (IP for the MLIT).

$$\max \prod_{(i,j)\in A} p_{ij}(1-p_{ij})^{x_{ij}(t_j-t_i-L)^+}(1-p_{ij})^{(1-x_{ij})\min\{D,(t_j-t_i-L+1)^+\}} \tag{3.9}$$

subject to

$$x_{ij}=0 \text{ if } \begin{cases} t_j-t_i & \leq L-1 \\ & \text{or} \\ t_j-t_i & \geq D+L \end{cases} \qquad \forall (i,j)\in\mathcal{A} \tag{3.10}$$

$$\sum_{j\in\Gamma^+(s)} x_{rj} \geq 1 \tag{3.11}$$

$$\sum_{i\in\Gamma^-(j)} x_{ij} = 1 \qquad \forall j\in\mathcal{I}_i\setminus\{s\} \tag{3.12}$$

$$\sum_{i\in\Gamma^-(j)} x_{ij} = 0 \qquad \forall j\in\mathcal{I}_n \tag{3.13}$$

$$\sum_{i\in\Gamma^-(j)} x_{ij} \leq 1 \qquad \forall j\in\mathcal{N}\setminus\mathcal{I} \tag{3.14}$$

$$\sum_{i\in\Gamma^-(j)} x_{ij} \geq x_{jk} \qquad \forall j\in\mathcal{N}\setminus\{s\}, k\in\Gamma^+(j) \tag{3.15}$$

$$t_i = T_i \qquad \forall i\in\mathcal{I}_i \tag{3.16}$$

$$t_i = M \qquad \forall i\in\mathcal{I}_n \tag{3.17}$$

$$t_i \geq M\left(1-\sum_{j\in\Gamma^-(i)} x_{ji}\right) + T_{min}\sum_{j\in\Gamma^-(i)} x_{ji} \qquad \forall i\in\mathcal{N}\setminus\mathcal{I} \tag{3.18}$$

$$t_i \leq M\left(1-\sum_{j\in\Gamma^-(i)} x_{ji}\right) + T_{max}\sum_{j\in\Gamma^-(i)} x_{ji} \qquad \forall i\in\mathcal{N}\setminus\mathcal{I} \tag{3.19}$$

$x_{ij}\in\{0,1\}, t_i\in\mathbb{Z}$.

Objective function (3.9) maximizes the likelihood that the set of links (i,j) for which $x_{ij}=1$ infers the correct infection tree. Constraint (3.10) is the *feasibility constraint* and sets variables x_{ij} to zero if link (i,j) is not feasible. Constraints (3.11) to (3.15), hereby referred to as *flow constraints* ensure that $\mathcal{T}^\star$ is a rooted tree that spans from the root node r to at least all infected information nodes. Constraints (3.16) and (3.17) enforce that the timestamps of information nodes are fixed, where M is a large constant designed to reproduce an infinite infection time; whereas constraints (3.18) and (3.19) ensure that the timestamp of zero-information nodes is in the range $[T_{min}, T_{max}]$ only if the individual is infected. These constraints are hereby referred to as the *timestamps constraints*.

If every node of the network is an information node, that is $\mathcal{I}=\mathcal{N}$, solving Model 1 is equivalent to finding the most likely spanning tree to all infected individuals. In this scenario, all variables t_i are fixed, hence x_{ij} are the only decision variables. Otherwise, objective function (3.9) is nonlinear with respect to decision variables x_{ij} and $t_i, \forall i\in\mathcal{N}\setminus\mathcal{I}$ and the resulting model is an IP which is untractable when the size of the instances increases. It should be noted that instances can be preprocessed: non-infected information nodes $i\in\mathcal{I}_n$ as well as infeasible links between two infected information nodes $i,j\in\mathcal{I}_i$ can be removed from the network to improve computational performance. We next present a new solution method to find the MLIT in outbreaks with limited information.

Figure 1: Infection path between a pair of infected information nodes containing only zero-information nodes

4. Solution Method

In this section we present a new solution method to solve Model 1. Due to the potentially large number of Feasible Infection Path (FIP) between any two infected information nodes in networks with limited information, we propose to restrict the search to a set of good candidates. To do so, we stress the impact of the hop-distance on the likelihood of a FIP in the objective function. The hop-distance (length) between two nodes in an unweighted graph can be defined as the number of links in the Shortest Path (SP) connecting them. While the length of a path is not the only criterion to evaluate its likelihood, it is fair to assume that the longest FIP is most certainly not the most likely. This hypothesis is driven by the fact that the contribution of subset of links in the objective is determined by a product of probabilistic quantities. Hence an increase in the hop-distance of a FIP is likely to reduce the likelihood of the path in objective function (3.9). Our solution method is based on this heuristic approach and can be decomposed into three main steps: (1) finding (at most) K shortest FIP between the root node and every infected information node (hereby referred to as *OD pair*), (2) ranking the obtained infection paths using lower and upper bounds on path-likelihood and (3) solving an exact MILP-reformulation of Model 1 on a range of subgraphs containing a predetermined maximum number of FIP per OD pair.

4.1. Graph Reduction Algorithm

We first formally extend the concept of link-feasibility to path-feasibility.

Definition 2 (Feasible Infection Path). Let $\mathcal{P}_{rs}$ be a path from node r to node s, path $\mathcal{P}_{rs}$ is a Feasible Infection Path (FIP) if

$$\forall (i,j) \in \mathcal{P}_{rs}, \qquad L \leq t_j - t_i \leq L + D - 1 \tag{4.1}$$

In the context of outbreaks with limited available information, a link from or to a zero-information node is potentially feasible, hence it is not straightforward to evaluate the feasibility of an infection path from or to a zero-information node. However the feasibility of an infection path between an OD pair, *i.e.* infected information nodes, can be decided efficiently. We first consider the elementary case where a subpath between two infected information nodes is composed of zero-information nodes only.

Proposition 1 (Subpath Feasibility). *Let $\mathcal{P}_{rs} = \{r, z_1, \ldots, z_l, s\}$ be a subpath between $r, s \in \mathcal{I}_i$ composed of zero-information nodes $z_i \in \mathcal{N} \setminus \mathcal{I}$ only, as depicted by Figure 1. $\mathcal{P}_{rs}$ is a FIP if and only if*

$$L|\mathcal{P}_{rs}| \leq T_s - T_r \leq (L + D - 1)|\mathcal{P}_{rs}| \tag{4.2}$$

where $|\mathcal{P}_{rs}|$ is the number of links in path $\mathcal{P}_{rs}$.

Proof. *If $\mathcal{P}_{rs}$ is a FIP, then by construction $\forall (i,j) \in \mathcal{P}_{rs}, L \leq t_j - t_i \leq L + D - 1$. Hence*

$$\underbrace{L + \cdots + L}_{|\mathcal{P}_{rs}| \text{ times}} \leq T_s - t_{z_l} + t_{z_l} + \cdots - t_{z_1} + t_{z_1} - T_r \leq \underbrace{(L + D - 1) + \cdots + (L + D - 1)}_{|\mathcal{P}_{rs}| \text{ times}} \tag{4.3}$$

which is equivalent to inequality (4.2). Reciprocally, since any link $(i,j) \in \mathcal{P}_{rs}$ is potentially feasible, inequality (4.2) can be broken into $|\mathcal{P}_{rs}|$ valid inequalities

$$L \leq t_{z_1} - T_r \leq L + D - 1 \tag{4.4}$$

$$\cdots$$

$$L \leq T_s - t_{z_l} \leq L + D - 1 \tag{4.5}$$

Hence every link in $\mathcal{P}_{rs}$ is a feasible infection link and $\mathcal{P}_{rs}$ is a FIP. □

Figure 2: Infection path between a pair of infected information nodes

To evaluate the feasibility of an infection occuring between any two infected information nodes, infection paths can be decomposed into subpaths containing zero-information nodes only.

Proposition 2 (Path Feasibility). *Let $\mathcal{P}_{rs}$ be a path with $r, s \in \mathcal{I}_i$ as depicted by Figure 2. $\mathcal{P}_{rs}$ is a FIP if and only if every subpath between any two infected information nodes in $\mathcal{P}_{rs}$ is a FIP.*

Proof. *If $\mathcal{P}_{rs}$ is a FIP, then by construction any link in $\mathcal{P}_{rs}$ is feasible, hence any subpath $\mathcal{P}_{i_j i_k} \subseteq \mathcal{P}_{rs}$ where $i_j, i_k \in \mathcal{I}_i$ is a FIP. Reciprocally, let $\{r, i_1, \dots, i_l, s\}$ be the set of infected information nodes contained in $\mathcal{P}_{rs}$. If the subpaths $\mathcal{P}_{ri_1}$, …, $\mathcal{P}_{i_l s}$ are FIP, then by Proposition 1 we have*

$$L|\mathcal{P}_{ri_1}| \leq T_{i_1} - T_r \leq (L + D - 1)|\mathcal{P}_{ri_1}| \tag{4.6}$$

$$\dots$$

$$L|\mathcal{P}_{i_l s}| \leq T_s - T_{i_l} \leq (L + D - 1)|\mathcal{P}_{i_l s}| \tag{4.7}$$

Summing these inequalities yields

$$L(|\mathcal{P}_{ri_1}| + \dots + |\mathcal{P}_{i_l s}|) \leq T_s - T_r \leq (L + D - 1)(|\mathcal{P}_{ri_1}| + \dots + |\mathcal{P}_{i_l s}|) \tag{4.8}$$

where by construction $|\mathcal{P}_{ri_1}| + \dots + |\mathcal{P}_{i_l s}| = |\mathcal{P}_{rs}|$. □

To reduce the size of graph $\mathcal{G}$, we propose to find at most K FIP per OD pair and regroup these paths into a subgraph. This graph reduction heuristic is inspired by the efficient algorithms available to solve the length (hop-distance) constrained SP [32] and the k loopless SP problem [39]. Namely, our approach works in two stages: (1) find the shortest FIP from the root to an infected information node and (2) find the next $K - 1$ shortest FIP for this OD pair if they exist – recall that only the existence of a single FIP per OD pair is ensured. The first stage can be achieved using a recursive procedure which re-labels nodes until they meet the feasibility constraint. The second stage requires to search for feasible deviations from the shortest FIP and can be done using a k-SP like procedure.

From Proposition 1, we know that an infection path between $r, s \in \mathcal{I}_i$ is feasible if and only if $L|\mathcal{P}_{rs}| \leq T_s - T_r \leq (L + D - 1)|\mathcal{P}_{rs}|$, hence the length of the candidate paths is bounded by

$$\underline{H}_{rs} = \left\lfloor \frac{T_s - T_r}{L + D - 1} \right\rfloor \leq |\mathcal{P}_{rs}| \leq \left\lceil \frac{T_s - T_r}{L} \right\rceil = \overline{H}_{rs} \tag{4.9}$$

$\underline{H}_{rs}$ and $\overline{H}_{rs}$ are the minimum and maximum feasible number of hops for path $\mathcal{P}_{rs}$, respectively. Hence to find the shortest FIP from r to s, we need to find the SP such that $|\mathcal{P}_{rs}| \geq \underline{H}_{rs}$. Saigal [32] introduced a dynamic programming algorithm – later revised by Rosseel [30] – to find the SP of a given length. Using his notation, let $C_{rs}(h)$ be the length of the SP from r to s with h hops and let $\mathcal{P}_{rs}(h)$ be this path. The procedure is initialized as

$$C_{rs}(1) = \begin{cases} 1 & \text{if } (r,s) \in \mathcal{A} \\ \infty & \text{otherwise} \end{cases} \tag{4.10}$$

$$\mathcal{P}_{rs}(1) = \begin{cases} \{r, s\} & \text{if } (r,s) \in \mathcal{A} \\ \emptyset & \text{otherwise} \end{cases} \tag{4.11}$$

and the recursive formulation (with unit link weights) for every $h \geq 2$ can be summarized as

$$C_{rs}(h+1) = \min_{j \in \Gamma^-(s)} \{C_{rs}(h)\} + 1 \tag{4.12}$$

$$\mathcal{P}_{rs}(h+1) = \mathcal{P}_{r\sigma} \cup \{s\} \quad \text{where} \quad \sigma = \arg\min_{j \in \Gamma^-(s)} \{C_{rs}(h)\} \tag{4.13}$$

This algorithm stops when h is equal to the desired number of hops and takes $\mathcal{O}(h|\mathcal{N}|^2)$ time. In our case, the recursive formulation needs to be evaluated at most $\overline{H}_{rs}$ times before the shortest FIP from r to s is found. To find the next shortest FIP, we use a procedure inspired by k-SP algorithms. Most k-SP algorithms start by finding the SP and then look among the possible deviations from this path to find next SP. This procedure is then iteratively applied until k paths have been found. Yen [39] algorithm uses two containers to store the candidate SPs found: we denote A_{rs}^{SP} the container containing the k SPs and B_{rs}^{SP} the one containing all the deviations from r to s that have been identified so far. We denote A_{rs}^{SFIP} the container containing the K shortest FIPs from r to s. The pseudo-code of the graph reduction heuristic is summarized in Algorithm 1. The K-SFIP algorithm first searches for the shortest FIP using the recursive procedure defined by (4.12) and (4.13). Once the shortest FIP has been found, the algorithm restricts the search for deviations to the paths contained in A_{rs}^{SFIP} and uses a k-SP like procedure to enumerate candidate paths for the current OD pair. This algorithm stops when the K shortest FIP are found or if the length of the current SP is strictly greater than the maximum feasible number of hops for this OD pair.

Algorithm 1: K-SFIP Algorithm

Data: A graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, a constant K
Result: A subgraph $\mathcal{G}_K = (\mathcal{N}_K, \mathcal{A}_K)$
$s \leftarrow \arg\min_{i \in \mathcal{I}_i} \{t_i\}$;
for $r \in \mathcal{I} \setminus \{s\}$ **do**
  Find 1-SFIP using the recursive procedure (4.12) and (4.13) and store it in A_{rs}^{SFIP};
  while $|A_{rs}^{SFIP}| < K \wedge T_s - T_r \geq |\mathcal{P}_{rs}|L$ **do**
    for *all deviations from the last path stored in* A_{rs}^{SFIP} **do**
      Find the SP and store it in B_{rs}^{SP};
    end
    Move the shortest deviation $\mathcal{P}_{rs}$ from B_{rs}^{SP} to A_{rs}^{SP};
    if $\underline{H}_{rs} \leq |\mathcal{P}_{rs}| \leq \overline{H}_{rs}$ **then** store $\mathcal{P}_{rs}$ in A_{rs}^{SFIP};
  end
end
$\mathcal{G}_K \leftarrow \bigcup_{s,r \in \mathcal{I}} (A_{rs}^{SFIP})$;

Theorem 1 (Correctness and time complexity of the K-SFIP algorithm). *The K-SFIP algorithm finds at least one FIP and at most the K shortest FIP from the root node to every infected information node in $\mathcal{O}(|\mathcal{I}_i|(\overline{H}|\mathcal{N}|^2 + K|\mathcal{N}|^3))$ time, where $\overline{H} = \max_{r,s \in \mathcal{I}_i} \{\overline{H}_{rs}\}$.*

Proof. *The K-SFIP algorithm starts by finding the shortest FIP from r to s using the recursive formulation defined by* (4.12) *and* (4.13). *Since we assume the existence of at least one FIP from the root node r to all infected information nodes $s \in \mathcal{I}_i$ and since the time complexity of Saigal [32] recursive algorithm is $\mathcal{O}(h|\mathcal{N}|^2)$, the shortest FIP from r to s can be found in $\mathcal{O}(\overline{H}_{rs}|\mathcal{N}|^2)$ time. To find the next shortest FIP, the algorithm searches among deviations from the last shortest FIP stored in A_{rs}^{SFIP}. The shortest deviation is moved from B_{rs}^{SP} to A_{rs}^{SP} and, if this path is a FIP, it is the next shortest FIP and this path is moved to A_{rs}^{SFIP}. The search for K shortest FIP terminates when K (shortest) FIP from r to s have been found or if the length of the current k-SP times the exposed period is greater than the timestamp difference for the current OD pair, that is, if $T_s - T_r < |\mathcal{P}_{rs}^k|L$. In this case, all the remaining paths for the current OD pair are infeasible. Observe that all the paths found after the shortest FIP has been found are either feasible if $T_s - T_r \geq |\mathcal{P}_{rs}^k|L$ or infeasible. Hence the while loop is executed at most $K - 1$ times. Since the time complexity of Yen [39] k-SP algorithm is $\mathcal{O}(k|\mathcal{N}|^3)$ and at most K shortest FIP for all the pairs from r to every node in $\mathcal{I}_i \setminus \{r\}$ are seeked, the time complexity of the K-SFIP algorithm is $\mathcal{O}(|\mathcal{I}_i|(\overline{H}|\mathcal{N}|^2 + K|\mathcal{N}|^3))$.* □

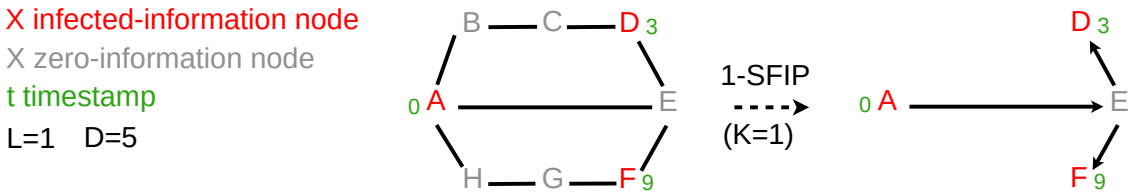


Figure 3: An outbreak scenario where the execution of the K-SFIP algorithm with $K = 1$ does not produce a subgraph containing a feasible infection tree.

Algorithm 1 provide a method to significantly reduce the search space. This heuristic however, does not ensure that the obtained subgraph contains a feasible infection tree. This is illustrated in Example 1.

Example 1. *Consider the network depicted by Figure 3. In this outbreak scenario, only three of the individuals are information nodes (red nodes). If Algorithm 1 is executed with $K = 1$, the search for the shortest FIP from root A to destination node D returns path $\{A, E, D\}$ and the search for the shortest FIP to destination node F returns path $\{A, E, F\}$. Solving Model 1 on the subgraph composed by both paths produces an infeasible solution because the contraints on node E are conflicting. Indeed, path $\{A, E, D\}$ imposes that $t_E \in [1, 2]$ and path $\{A, E, F\}$ imposes that $t_E \in [4, 8]$.* □

To increase the chances that the subgraph produced by the K-SFIP algorithm contains a feasible infection tree, it is necessary to increase the value of K. Searching for a large number of K shortest FIP leads to the generation of larger subgraphs and increases the probability of obtaining a feasible infection tree, however it may also introduce extra FIP which are unlikely to have spread the disease, hence resulting in a overload of nodes and links that may burden the resolution of Model 1. In the next section, we derive lower and upper bounds on the likelihood of a FIP in the objective function of Model 1. This method can be used to rank the K shortest FIP obtained after executing Algorithm 1 and generate multiple subgraphs with a predetermined maximum number of FIP per OD pair.

4.2. FIP Ranking Strategy

In order to derive a ranking strategy to sort FIPs, we observe that the likelihood of a FIP between any two infected information nodes can be estimated using bounds on the disease transmission probabilities. We first consider the case of a subpath composed by zero-information nodes only.

Proposition 3 (Lower and Upper Bounds on Subpath-likelihood). *Let $\mathcal{P}_{rs} = \{r, z_1, \ldots, z_l, s\}$ where $r, s \in \mathcal{I}_i$ and $z_1, \ldots, z_l \in \mathcal{N} \setminus \mathcal{I}$ be a FIP. If $\mathcal{P}_{rs}$ belongs to the infection tree, that is if $\forall (i, j) \in \mathcal{P}_{rs}, x_{ij} = 1$, the likelihood of $\mathcal{P}_{rs}$ is*

$$\prod_{(i,j)\in\mathcal{P}_{rs}} \lambda_{ij}(x_{ij} = 1) = \prod_{(i,j)\in\mathcal{P}_{rs}} \alpha_{ij} \tag{4.14}$$

and is bounded by

$$\underbrace{(\underline{p}_{rs})^{|\mathcal{P}_{rs}|}(1 - \overline{p}_{rs})^{(T_s - T_r - L|\mathcal{P}_{rs}|)}}_{l_{rs}} \leq \prod_{(i,j)\in\mathcal{P}_{rs}} \alpha_{ij} \leq \underbrace{(\overline{p}_{rs})^{|\mathcal{P}_{rs}|}(1 - \underline{p}_{rs})^{(T_s - T_r - L|\mathcal{P}_{rs}|)}}_{u_{rs}} \tag{4.15}$$

where

$$\underline{p}_{rs} = \min_{(i,j)\in\mathcal{P}_{rs}} \{p_{ij}\} \qquad \text{and} \qquad \overline{p}_{rs} = \max_{(i,j)\in\mathcal{P}_{rs}} \{p_{ij}\} \tag{4.16}$$

Proof. *The cost of $\mathcal{P}_{rs}$ in Objective function 3.9 is*

$$\prod_{(i,j)\in\mathcal{P}_{rs}} \lambda_{ij}(x_{ij} = 1) = \prod_{(i,j)\in\mathcal{P}_{rs}} \alpha_{ij} = p_{rm_1}(1 - p_{rz_1})^{(t_{z_1} - T_r - L)} \cdots p_{z_l s}(1 - p_{z_l s})^{(T_s - t_{z_l} - L)} \tag{4.17}$$

Defining $\underline{p}_{rs}$ and $\overline{p}_{rs}$ as in the proposition, we have

$$(\underline{p}_{rs})^{|\mathcal{P}_{rs}|} \leq p_{rz_1} \cdots p_{z_l s} \leq (\overline{p}_{rs})^{|\mathcal{P}_{rs}|} \tag{4.18}$$

Similarly, we have

$$(1-\overline{p}_{rs})^{(T_s-T_r-L|\mathcal{P}_{rs}|)} \leq (1-p_{rz_1})^{(t_{z_1}-T_r-L)} \cdots (1-p_{z_l s})^{(T_s-t_{z_l}-L)} \leq (1-\underline{p}_{rs})^{(T_s-T_r-L|\mathcal{P}_{rs}|)} \tag{4.19}$$

Since all the quantities are positive, combining the inequalities (4.18) *and* (4.19) *we obtain inequality* (4.15). □

Observe that inequality (4.15) does not depend on the timestamps of the zero-information nodes $t_{z_1}, \ldots, t_{z_l}$ but only on the lengty of path $\mathcal{P}_{rs}$, on the timestamps at its extremities and on the pairwise link probabilities; all of which are parameters in the model. Consequently, we can efficiently compute lower and upper bounds on the likelihood of each feasible subpath between any two infected information nodes in $\mathcal{T}^\star$. Proposition 3 can be extended to path-likelihood.

Proposition 4 (Lower and Upper Bounds on Path-likelihood). *Let $\mathcal{P}_{rs}$ be a FIP and let $\{r, i_1, \ldots, i_l, s\}$ be the set of infected information nodes contained in $\mathcal{P}_{rs}$. If $\mathcal{P}_{rs}$ belongs to the infection tree, that is if $\forall (i,j) \in \mathcal{P}_{rs}, x_{ij} = 1$, the likelihood of $\mathcal{P}_{rs}$ is bounded by*

$$\underbrace{l_{ri_1} \ldots l_{i_l s}}_{L_{rs}} \leq \prod_{(i,j)\in\mathcal{P}_{rs}} \alpha_{ij} \leq \underbrace{u_{ri_1} \ldots u_{i_l s}}_{U_{rs}} \tag{4.20}$$

where for all $i_j, i_k \in \mathcal{I}_i$, l_{i_j,i_k} and u_{i_j,i_k} are defined as in Proposition 3.

Proof. *$\mathcal{P}_{rs}$ can be decomposed into a set of subpaths between two consecutive information nodes, and Proposition 4 states that the likelihood of each of them is bounded by l_{i_j,i_k} and u_{i_j,i_k}. Since the bounds on the likelihood of a subpath are positive quantities, these bounds can be combined to obtain inequality* (4.20). □

Proposition 4 provides a mean to sort the FIPs obtained for each OD pair as the bounds L_{rs} and U_{rs} define a range of possible likelihoods for every FIP in objective function 3.9. We propose to use the mid-range value of the interval $[L_{rs}, U_{rs}]$ to sort the K (or less) shortest FIP obtained for every OD pair after the execution of Algorithm 1. The mid-range of the FIP $\mathcal{P}_{rs}$ is $(L_{rs}+U_{rs})/2$. Observe that if all the transmission probabilites are equal, then $L_{rs} = U_{rs}$ and the exact likelihood of every FIP in objective can be determined. Let S^w_{rs} be the set of the w FIP from r to s with the highest mid-range value, where $1 \leq w \leq K$; we define the subgraphs $\mathcal{G}_w$ as

$$\mathcal{G}_w \equiv \bigcup_{r,s\in\mathcal{I}} (S^w_{rs}) \tag{4.21}$$

Each subgraph $\mathcal{G}_w$ hence contains at most w FIP per OD pair and since the paths have been sorted in decreasing mid-range values we have: $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \cdots \subseteq \mathcal{G}_K$. Example 2 illustrates how FIP are sorted using the mid-range value ranking strategy.

Example 2. *Consider the network depicted by Figure 4 in which link transmission probabilities (in blue) are detailed. In this outbreak scenario, we assume that two FIP from root node A to information node D have been found by Algorithm 1 (executed with $K \geq 2$), namely 1-SFIP $= \{A, E, D\}$ of size 2 and 2-SFIP $= \{A, B, C, D\}$ of size 3. Computing the lower and upper bounds on the likelihood of 1-SFIP as defined in Proposition 3 gives*

$$l_{AD}^{1-SFIP} = (0.1)^2 \cdot (1-0.2)^{(3-0-1\times 2)} = 0.008 \tag{4.22}$$

$$u_{AD}^{1-SFIP} = (0.2)^2 \cdot (1-0.1)^{(3-0-1\times 2)} = 0.036 \tag{4.23}$$

and since all link probabilities of 2-SFIP are equal we have

$$l_{AD}^{2-SFIP} = u_{AD}^{2-SFIP} = (0.3)^3 \cdot (1-0.3)^{(3-0-1\times 3)} = 0.027 \tag{4.24}$$

Therefore the mid-range value of 1-SFIP is 0.022 *and the one of 2-SFIP is* 0.027. *Accordingly, 2-SFIP is ranked as the most likely FIP from A to D.* □

To find the MLIT, we propose to solve Model 1 for every subgraph $\mathcal{G}_w$ generated. Note that K such subgraphs can be obtained by a single run of Algorithm 1. We next present an exact reformulation of Model 1 into a MILP that will be used to find the MLIT on every subgraph $\mathcal{G}_w$.

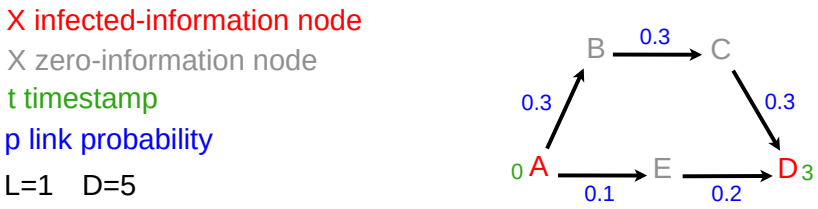


Figure 4: An outbreak scenario where the FIP ranking strategy finds that the SP is not the most likely.

4.3. Exact MILP Reformulation

In order to provide an efficient formulation to solve the MLIT, we introduce auxiliary decision variables and constraints to linearize objective function (3.9) and the feasibility constraint (3.10) in Model 1 with respect to decisions variables x_{ij} and t_i. It is a common practice among optimization techniques to consider the logarithm of likelihood functions instead of its proper expression. Applying this technique to (3.9) we obtain the following objective function

$$\max \sum_{(i,j)\in A} x_{ij}(t_j - t_i - L)^+ \log(p_{ij}(1-p_{ij})) + (1-x_{ij}) \min\{D, (t_j - t_i - L)^+\} \log(1-p_{ij}) \tag{4.25}$$

To achieve a linear formulation of objective function (4.25) we consider the relative timestamps of each link in the network. Let $\Delta t_{ij} \in \mathbb{Z}$ be the relative timestamp of nodes i and j defined as

$$\forall (i,j) \in \mathcal{A}, \qquad \Delta t_{ij} \equiv t_j - t_i \tag{4.26}$$

To represent the feasibility of an infection link in the model, we introduce a binary decision variable y_{ij} defined as

$$\forall (i,j) \in \mathcal{A}, \qquad y_{ij} \equiv \begin{cases} 1 & \text{if } L \le t_j - t_i \le L + D - 1 \quad (4.27) \\ 0 & \text{otherwise} \quad (4.28) \end{cases}$$

If $y_{ij} = 0$, then link (i,j) can be discarded from the infection tree, hence $x_{ij} = 0$. This relationship between variables x_{ij} and y_{ij} can be modelled using the next constraint

$$\forall (i,j) \in \mathcal{A}, \qquad x_{ij} \le y_{ij} \tag{4.29}$$

and variable y_{ij} can be introduced in the model through the constraints

$$\forall (i,j) \in \mathcal{A}, \qquad \begin{cases} \Delta t_{ij} \le y_{ij}(L + D - 1) + (1 - y_{ij})M & (4.30) \\ \Delta t_{ij} \ge y_{ij} L - (1 - y_{ij})M & (4.31) \end{cases}$$

where M is a large constant. Introducing variable Δt_{ij} in (4.25) we obtain the following expression

$$\max \sum_{(i,j)\in A} B_1 \log(p_{ij}(1-p_{ij})) + B_2 \log(1-p_{ij}) \tag{4.32}$$

where

$$B_1 = x_{ij}(\Delta t_{ij} - L) \tag{4.33}$$

$$B_2 = (1 - x_{ij}) \min\{D, (\Delta t_{ij} - L + 1)^+\} \tag{4.34}$$

To linearize cost function (4.25) we have to linearize B_1 and B_2. B_1 can be linearized by introducing a continuous decision variable $\rho_{ij}^{x,\Delta t}$ defined as

$$\forall (i,j) \in \mathcal{A}, \qquad \rho_{ij}^{x,\Delta t} \equiv x_{ij} \Delta t_{ij} \tag{4.35}$$

and the following set of constraints

$$\forall (i,j) \in \mathcal{A}, \qquad C(x_{ij}, \Delta t_{ij}) \equiv \begin{cases} \rho_{ij}^{x,\Delta t} \leq \Delta t_{ij} + (1 - x_{ij})M & (4.36) \\ \rho_{ij}^{x,\Delta t} \geq \Delta t_{ij} - (1 - x_{ij})M & (4.37) \\ \rho_{ij}^{x,\Delta t} \leq x_{ij} M & (4.38) \\ \rho_{ij}^{x,\Delta t} \geq -x_{ij} M & (4.39) \end{cases}$$

$C(x_{ij}, \Delta t_{ij})$ ensure that variable $\rho_{ij}^{x,\Delta t}$ behaves as the product $x_{ij} \cdot \Delta t_{ij}$. If $x_{ij} = 1$, then the first two constraints ensure that $\rho_{ij}^{x,\Delta t} = \Delta t_{ij}$. If $x_{ij} = 0$, then the first two constraints ensure that $\rho_{ij}^{x,\Delta t}$ is lower than a positive value and greater than a negative value and the last two constraints ensure that $\rho_{ij}^{x,\Delta t} = 0$. This exact reformulation is possible mainly because Δt_{ij} is a bounded variable; note that this reformulation is also possible if Δt_{ij} is a continuous variable [24]. In the remainder of this paper, the notation $C(bin, int)$ is used to design the set of constraints used to reformulate the decision variable product $bin \cdot int$, where bin is a binary variable and int is a bounded integer variable.

B_1 can hence be expressed linearly as

$$B_1 = \rho_{ij}^{x,\Delta t} - x_{ij} L \tag{4.40}$$

To linearize B_2 we first introduce an auxiliary decision variable. Let $m_{ij} \in \mathbb{Z}^+$ be defined as

$$\forall (i,j) \in \mathcal{A}, \qquad m_{ij} \equiv \min\{D, (\Delta t_{ij} - L + 1)^+\} \tag{4.41}$$

Since the model aims to maximize either B_1 or B_2, variable m_{ij} can be introduced in the model using the following constraints

$$\begin{aligned} m_{ij} &\leq D & (4.42) \\ m_{ij} &\leq l_{ij} & (4.43) \\ l_{ij} &\geq \Delta t_{ij} - L + 1 & (4.44) \\ l_{ij} &\geq 0 & (4.45) \end{aligned}$$

where l_{ij} is a continuous decision variable defined as

$$\forall (i,j) \in \mathcal{A}, \qquad l_{ij} \equiv (\Delta t_{ij} - L + 1)^+ \tag{4.46}$$

m_{ij} is upper bounded by the infectious period D and is positive, hence B_2 can be expressed linearly as

$$B_2 = m_{ij} - \rho_{ij}^{x,m} \tag{4.47}$$

where $\rho_{ij}^{x,m}$ is a continuous decision variable defined as

$$\forall (i,j) \in \mathcal{A}, \qquad \rho_{ij}^{x,m} \equiv x_{ij} m_{ij} \tag{4.48}$$

through constraint set $C(x_{ij}, m_{ij})$. The resulting model is presented below

Model 2 (MILP for the MLIT).

$$\max \sum_{(i,j)\in A} \left(\rho_{ij}^{x,\Delta t} - x_{ij}L\right)\log(p_{ij}(1-p_{ij})) + (m_{ij} - \rho_{ij}^{x,m})\log(1-p_{ij}) \tag{4.49}$$

subject to

$$\left.\begin{aligned}
\rho_{ij}^{x,\Delta t} &\equiv C(x_{ij}, \Delta t_{ij}) \\
\rho_{ij}^{x,m} &\equiv C(x_{ij}, m_{ij}) \\
\Delta t_{ij} &= t_j - t_i \\
\Delta t_{ij} &\leq y_{ij}(L + D - 1) + (1 - y_{ij})M \\
\Delta t_{ij} &\geq y_{ij}L - (1 - y_{ij})M \\
m_{ij} &\leq D \\
m_{ij} &\leq l_{ij} \\
l_{ij} &\geq \Delta t_{ij} - L + 1 \\
l_{ij} &\geq 0 \\
x_{ij} &\leq y_{ij}
\end{aligned}\right\} \quad \forall (i,j) \in \mathcal{A} \tag{4.50}$$

Flow constraints (3.11) –(3.15)

Timestamps constraints (3.16) –(3.19)

$x_{ij}, y_{ij} \in \{0,1\}, t_i, \Delta t_{ij} \in \mathbb{Z}, m_{ij} \in \mathbb{Z}^+, l_{ij} \in \mathbb{R}$.

Model 2 is an exact reformulation of Model 1 that can be solved using commercial MILP optimization software. We use Model 2 in our solution method to find the MLIT on the subgraphs induced by Algorithm 1. In the next section, we present the validation framework used to measure the performance of our approach.

5. Validation Framework

To validate our approach we (1) simulate outbreak scenarios on a randomly generated network using the SEIR compartmental model (presented in Section 3.1), (2) extract information on a subset of the nodes corresponding to a specified level of information availability which is to be used as input for the model, (3) implement the solution methodology, and (4) evaluate the performance of the solution method based on its ability to accurately identify the actual infection pattern.

To create the network structures we use a random graph generator which relies on a preferential attachment rule, resulting in networks with power law node degree distribution. Various studies have found that power law networks are representative of many real world networks, including social contact networks [4, 16]. Power law networks have a hub and spoke type structure with few highly connected nodes, (known as super spreaders in the context of contagion problems), while most nodes have a very low degree. The level of heterogeneity depends on the power law exponent. In this work we present results from a power law network with an exponent of 3, which is representative of many real world network structures [5]. The networks are generated according to the method developed in [36].

Due to the heterogeneous structure of power law networks, contagion processes can result in a wide range of scenarios, even for the same set of disease parameters and source of infection. Given the stochastic nature of the contagion process and the fact that the model performance can vary drastically based on the specific contagion process which evolves, the expected performance of the model is evaluated by averaging over multiple samples (*i.e* simulated contagion episodes). For each sample the model performance is based on how accurately it predicts the actual paths of infection (which are extracted from simulation outputs). In this case study, we use the following parameters values:

- Exposed period $L = 1$

- Infectious period $D = 5$
- Observation date: 15 time steps
- Disease transmission probabilities $\forall (i,j) \in \mathcal{A}, p_{ij} \in [0.1, 0.3]$
- Maximum number of FIP per OD pair: $K = 5$

Both exposed and infectious periods are expressed in units of time and the outbreak data is extracted after 15 simulation steps (recall that the contagion process herein is assumed to evolve with a discrete time step). The disease transmission probabilities are randomly distributed uniformly across the links within the range $[0.1, 0.3]$. These bounds result in a wide range of outbreak scenarios, ranging from 0% to 75% of the network becoming infected, representing the highly stochastic nature or real-world outbreaks. To reproduce the limited availability of information, we randomly select a subset of the nodes to be information nodes.

The model performance is measured by comparing the set of nodes and links involved in the spread of the disease in the SEIR simulation-based scenario (hereby referred to as actual outbreak) with those identified by the MLIT obtained by the solution method. Of specific interest in this work is the performance of the solution method for different levels of information availability *i.e.* size of the information subset. Recall that the status and the timestamps of zero-information nodes is unknown from the perspective of the solution method. For the evaluation of each sample we define a set of metrics to quantify the performance of the solution method. The metrics are determined by comparing the actual infection tree obtained at the chosen observation date (*i.e.* 15 simulation steps), with the MLIT obtained by the solution method. We consider two types of metrics:

- Link metrics; which compare the observed network links that have spread the disease (extracted from the SEIR simulation) with the links identified by the MLIT.
- Node metrics; which compare the observed comportmental of the nodes (extracted from the SEIR simulation) with the comportmental status of the nodes as specified by the MLIT.

The following steps are used to compute the performance metrics: for a given network $\mathcal{G}$, disease parameters D and L and a range of link transmission probabilities p_{ij}, and a predetermined number K of FIP per OD. For the levels of information: 50%, 60%, 70%, 80%, 90% and 100%, do

1. Randomly assign uniformly distributed link transmission probabilities in the range $[0.1, 0.3]$
2. Randomly introduce an infected individual into the network (root node)
3. Simulate an outbreak for 15 time steps using the SEIR model
4. Extract the observed infection tree $\mathcal{T}^{obs}$ from the simulation to use for evaluating the performance of the solution method
   (a) The full set of links in $\mathcal{T}^{obs}$
   (b) The full set of infected nodes in $\mathcal{T}^{obs}$ and their timestamps
5. Randomly select a subset $\mathcal{I}$ of nodes according to the level of information and include the source node in the subset $\mathcal{I}$
6. Extract the following (required) information from the simulation to use as input for the solution method
   (a) The set of information nodes $\mathcal{I}$
   (b) The timestamps of all information nodes $T_i, \forall \in \mathcal{I}$
7. Implement the solution method and store the MLIT for each subgraph $\mathcal{G}_w$, where $1 \leq w \leq K$
8. Identify the percentage of correctly identified links and nodes in the MLIT and determine
   (a) *Link Metric LM* – compares $\mathcal{T}^{\star}$ and $\mathcal{T}^{obs}$; *e.g.* if every link in $\mathcal{T}^{obs}$ is in $\mathcal{T}^{\star}$, then $LM = 100\%$.
   (b) *Subgraph Metric 1 SM1* – compares $\mathcal{G}_w$ and $\mathcal{G}$; *e.g.* if every link in $\mathcal{G}$ is in $\mathcal{G}_w$, then $SM1 = 100\%$.
   (c) *Subgraph Metric 2 SM2* – compares $\mathcal{G}_w$ and $\mathcal{T}^{obs}$; *e.g.* if every link in $\mathcal{T}^{obs}$ is in $\mathcal{G}_w$, then $SM2 = 100\%$.
   (d) *Zero-information node status metric ZISM* – compares the status of node in $\mathcal{N} \setminus \mathcal{I}$ with the observed comportmental status of nodes; *e.g.* if the status of every node in $\mathcal{N} \setminus \mathcal{I}$ is correctly identified, then $ZISM = 100\%$.

(e) *Zero-information node timestamp metric* $ZITM$ – compares the timestamp of node in $\mathcal{N} \setminus \mathcal{I}$ with the observed timestamp of nodes; *e.g.* if the timestamp of every node in $\mathcal{N} \setminus \mathcal{I}$ is correctly identified, then $ZITM = 100\%$.
(f) *Infected nodes metric* INM – compares the status of node in $\mathcal{N} \setminus \mathcal{I}$ such that $t_i \neq M$ with the nodes in $\mathcal{T}^{obs}$.
(g) *Non-infected nodes metric* $NINM$ – compares the status of node in $\mathcal{N} \setminus \mathcal{I}$ such that $t_i = M$ with the nodes not in $\mathcal{T}^{obs}$.

9. Repeat steps 1 to 8 500 times and extract statistics on the performance of the solution method.

The procedure outlined above returns the expected performance of our solution method, which is how accurately the MLIT represents the actual spreading scenario, for a specified network structure, level of information and set of disease parameters. Given the stochastic nature of the contagion process and the fact that the model performance will vary based on the size of the outbreak and the specific contagion process which occurs, the results are averaged over 500 samples to generate an expected performance. The computed metrics reflect the ability of the model to appropriately identify infected nodes and infection spreading links, while also penalizing the model for over-infecting the network.

The solution method is implemented with $K = 5$ and for different values of w: namely 1, 3 and 5. Recall that the graph $\mathcal{G}_1$ obtained for $w = 1$ is a subgraph of the subgraph obtained for higher values of w, that is $\mathcal{G}_1 \subseteq \mathcal{G}_3 \subseteq \mathcal{G}_5 \subseteq \mathcal{G}$. To measure the performance of the K-SFIP algorithm, we also implement Model 2 on graph $\mathcal{G}$ (data series $w = N/A$). The simulation of the SEIR compartmental model and the K-SFIP algorithm are implemented in C++ on a Windows 64-bit machine with 8Gb of RAM and the optimization problems (MILP instances) are solved using the CPLEX v12.5 commercial package (C++ API) with a time limit of one minute and an integrality gap of $1e-5$.

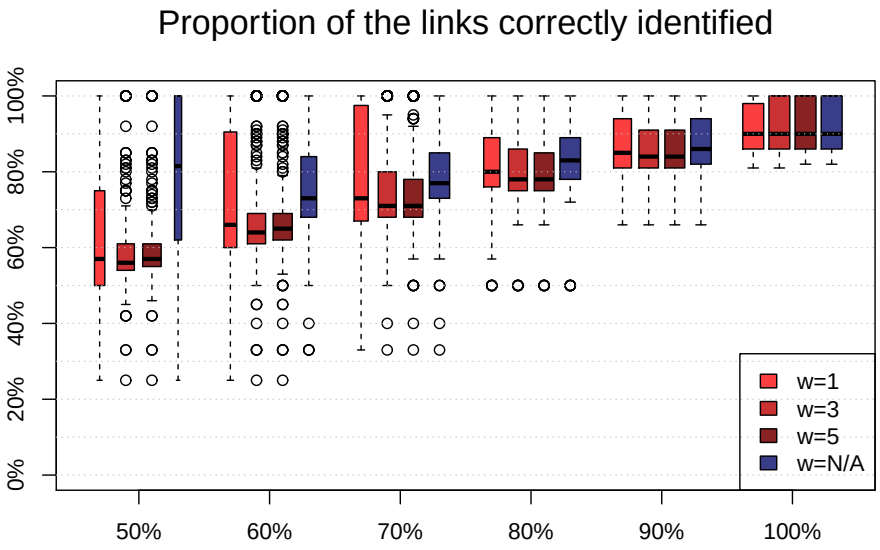
In the next section, we discuss the results obtained for a range of information availability levels on a randomly generated power-law network of 1000 nodes and 3122 links.

6. Results

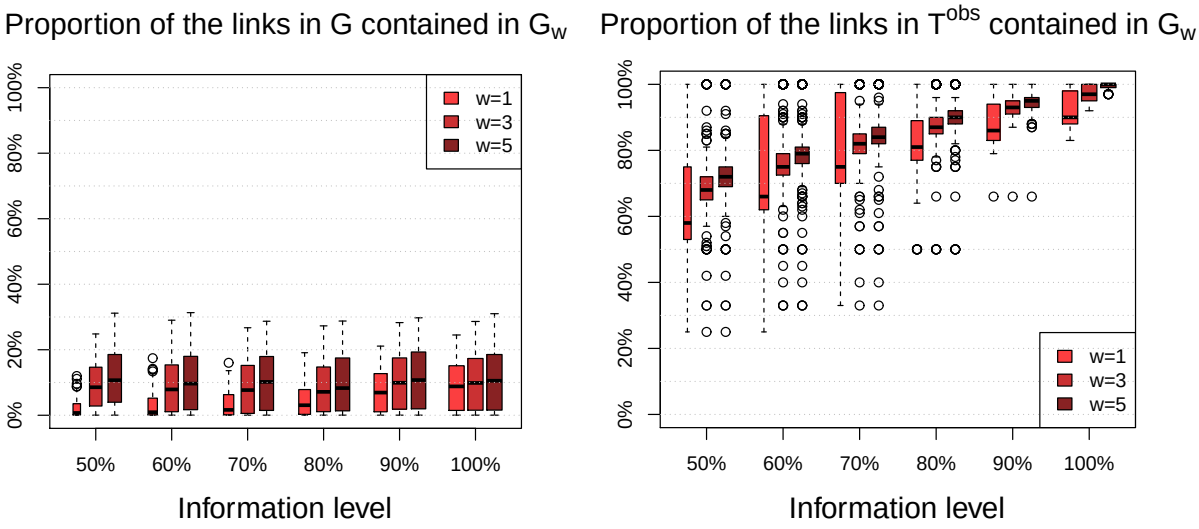
Figure 5 summarizes the link metrics, while Figure 6 and 7 summarize the node metrics. It should be noted that the width of the boxplots is proportional to the sample size. The data plotted corresponds to feasible solutions, *i.e.* every infeasible sample is not represented on the charts (see Table 1 for a detailed analysis on the proportion of feasible solutions).

The proportion of the links correctly identified by the solution method (see Figure 5a) is shown to increase with the information level and the value of w. The average number of links correctly identified by the solution method is more volatile for low levels of information than for scenarios where 100% of the node-level information is available. When a single FIP to every infected information node is sought and for an information level of only 50%, the expected proportion of links correctly identified by the solution method is concentrated between 50% and 70%. This range increases beyond 90% when all the information is available. On the aggregate, increasing the value of w beyond 3 only minimally improves performance, mainly reducing volatility. When the K-SFIP algorithm is disabled, that is, Model 2 is solved on graph $\mathcal{G}$ (data series $w = N/A$), the proportion of links correctly identified is improved by more than 10% for low information levels; however this gain depreciates for higher levels of information.

Figures 5b and 5c illustrate the proportion of the total links in graoh $\mathcal{G}$ contained in subgraph $\mathcal{G}_w$ (for $w = 1, 3, 5$) and the proportion of the links in the observed infection tree $\mathcal{T}^{obs}$ contained in subgraph $\mathcal{G}_w$ respectively. We observe that the collection of FIP almost always represents less than 20% of the initial link set. This figure increases with w but remains stable across all information levels. While only a fraction of the initial graph is included in subgraphs $\mathcal{G}_w$, they generally contain more than 60% of the infection tree for low levels of information and contain almost the whole infection tree when all the node-information is available. In fact, this figure increases significantly with the information level and is a proxy for the proportion of links correctly identified.



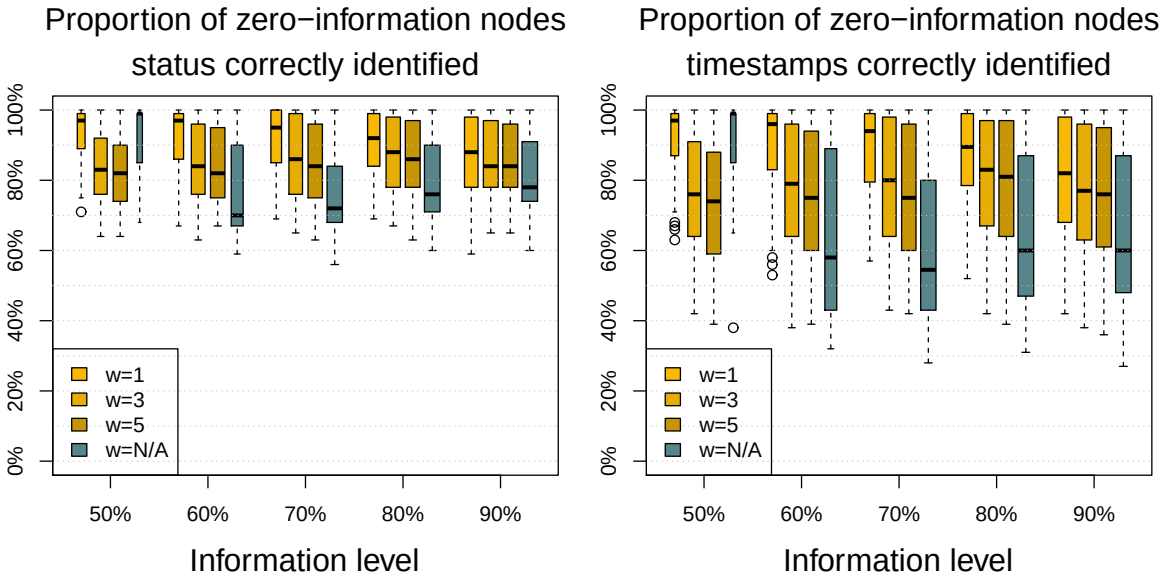
(a) LM



(b) $SM1$ (c) $SM2$

Figure 5: Performance of the solution method: link identification metric and K-SFIP algorithm

Figure 6 illustrates the performance of the solution method with regards to the identification of zero-information nodes. The proportion of zero-information nodes whose status has been correctly identified (see Figure 6a) is at least 80% when the K-SFIP algorithm is implemented, whereas solving Model 2 directly results in poorer performance. Here increasing the number of FIP per OD slightly deteriorates the performance of our approach with regards to this metric. We report a similar trend for the proportion of zero-information nodes whose timestamp have been correctly identified (see Figure 6b), although as expected the solution method is globally less efficient with this metric, especially when w increases. Solving the MLIT on subgraph $\mathcal{G}_1$, we observe that on average 70% to 90% of the timestamps are correctly identified when the instance is feasible. Note that the decrease of this figure with the information level is a consequence of the low proportion of feasible solutions for low levels of information. Using $\mathcal{G}_3$ or $\mathcal{G}_5$, the average performance is stable above 75% and slightly increases with the information level. Here also, poorer performance is observed when the MILP is solved on $\mathcal{G}$.



(a) $ZISM$ (b) $ZITM$

Figure 6: Performance of the solution method: Zero-information node Metric

Figure 7 extends the node level performance analysis by distinguishing between infected and non-infected zero-information nodes. The performance metrics are plotted against the proportion of the population infected to reveal the performance of the solution methodology relative to outbreak size. Each dot on a subfigure represents a sample (outbreak scenario); the x-axis corresponds to the proportion of individuals infected in a given outbreak scenario and the y-axis represents the corresponding performance of the node-level metrics for both infected (metric INM) and non-infected (metric $NINM$) zero-information nodes. Specifically, the percentage of nodes in each state correctly identified by the solution method are graphed. We observe that the proportion of infected individuals correctly identified is highly volatile for small outbreak scenarios, especially when less than 10% of the population is infected. Conversely, the proportion of non infected individuals correctly identified is stable and close to 100% in these contagion episodes. This trend shifts as the outbreak size increases: on average, the proportion of non infected individuals correctly identified decreases whereas the proportion of infected individuals correctly identified stabilizes around 50%. This trend is reversed when the K-SFIP algorithm is disabled, wherein more than 75% of the infected population is correctly identified by the model due to the additional number of possible paths generated. This is especially beneficial with larger outbreaks. However, a drop in the identification of non infected individuals occurs simultaneously.

Finally, Table 1 summarizes the computational performance of our solution method. For each set of instances (information level and subgraph) the mean runtime of the K-SFIP algorithm is given (recall that one execution of the K-SFIP algorithm provides all the subgraphs for a number of FIP per OD pair). These results clearly show that the heuristic is computationaly efficient, even in the context of low information availability. The performance of the MILP resolution is detailed in three columns: the proportion of feasible solutions, the proportion of optimal solutions and the mean runtime. Recall that a one minute time limit is imposed. We report that only 1/4 of the subgraphs $\mathcal{G}_1$ contain a feasible infection tree in the case where the information level is set to 50%, while only 1/8 of the instances are proved feasible when the heuristic is disabled (dataseries w=N/A). Globally, the number of feasible solutions increases with w, however this is not always the case for the number of optimal solutions (*e.g.* for low information levels). This trend highlights the trade-off between the number of FIP per OD pair requested and computational performance of the solution method. When all the node-level information is available, the solution method is computationaly efficent as all the instances are solved to optimality, with or without the use of the K-SFIP algorithm.

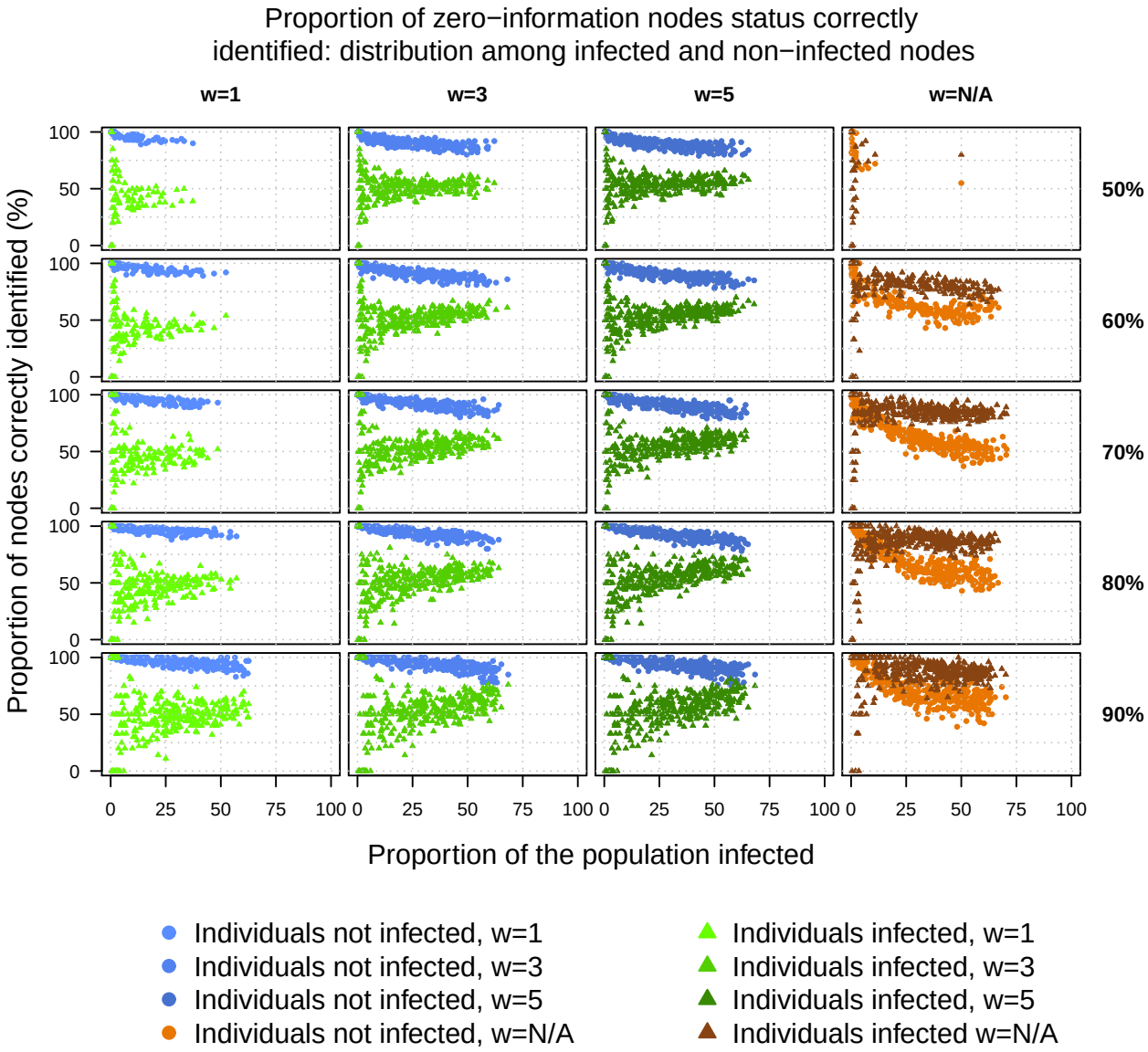


Figure 7: Performance of the solution method: INM and $NINM$

7. Conclusion

The problem addressed in this research is to infer a contagion pattern in a network utilizing partially available node-level information. The underlying problem is similar to a hop-constrained Steiner Tree problem but instead of constraining the entire tree, infection paths are constrained independently. We formulated the MLIT as an IP and proposed a three step solution method which relied on reducing the initial graph by finding at most K shortest FIP, ranking the obtained paths using bounds on their likelihood and solving an exact MILP reformulation of the initial IP on every subgraph with a predetermined number of FIP per OD pair. A polynomial time algorithm was developed for the graph reduction heuristic which was inspired by efficient procedures to solve the length constrained SP and the k-SP problems.

The specific application of focus was disease outbreaks in social contact networks. As expected, the performance of the solution method was sensitive to the level of information in the network: the proportion of links correctly identified increased with information availability. With 50% of the infections reported, the solution method was able to identify about 60% of the contacts responsible for spreading the disease and this

Instance		K-SFIP algorithm	MILP performance		
Info. level	w	Mean runtime (s)	% Feasible solution	% Optimal solution	Mean runtime (s)
50%	1	4.2	25.9	25.9	1.2
	3		71.8	48.6	24.3
	5		83.7	42.0	33.8
	N/A	N/A	12.4	5.3	37.7
60%	1	5.3	39.0	39.0	1.2
	3		77.8	70.3	9.8
	5		86.0	65.0	18.7
	N/A	N/A	58.8	10.9	51.0
70%	1	6.2	48.0	48.0	1.2
	3		78.0	78.0	2.0
	5		87.5	85.0	5.0
	N/A	N/A	99.5	32.5	43.6
80%	1	5.9	61.5	61.5	1.1
	3		83.7	83.7	1.3
	5		89.9	89.9	1.4
	N/A	N/A	99.6	89.5	11.1
90%	1	5.1	75.6	75.6	1.2
	3		90.5	90.5	1.3
	5		93.6	93.6	1.3
	N/A	N/A	99.8	99.8	1.5
100%	1	1.5	100.0	100.0	1.2
	3		100.0	100.0	1.3
	5		100.0	100.0	1.3
	N/A	N/A	100.0	100.0	1.4

Table 1: Computational performance of the solution method. The mean runtime of the K-SFIP algorithm with $K = 5$ is given for each set of instances. The number of feasible and optimal solutions obtained after solving Model 2 (MILP) using the CPLEX v12.5 solver with a one minute time limit and the mean running time (averaged over the set of feasible instances) for each subgraph $\mathcal{G}_w$ as well as graph $\mathcal{G}$ is detailed. Note that every MILP instance solved without using the K-SFIP algorithm is known to be feasible, however CPLEX fails to prove feasibility after one minute of runtime.

figure was increased to 90% when all the information is available. On average, 50% to 80% of unreported infected individuals were identified depending on the level of information available. From a heuristic perspective, the K-SFIP algorithm was shown to be efficient as it was able to generate subgraphs containing 20% of the links of the original graph which account for 60% to 100% of the observed infection tree, depending to the information availability.

The proposed methodology provides a novel procedure for evaluating a region that has been exposed to infection. The performance of the solution method was evaluated as a function of information availability, and was shown to accurately identify a significant proportion of the nodes and links responsible for the spread of a disease in a network. Some direct extensions of research include testing the solution method in the context of different network topologies (*e.g.* alternative node degree distributions), evaluating different disease parameter ranges, as well as comparing the performance at different observation periods, and finally investigating alternative objective functions which could improve the probabilistic inference framework. Additional extensions of this research evolve from relaxing the current set of model assumptions (*e.g.* the source of infection is known).

Acknowledgments

NICTA is funded by the Australian Department of Communications and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] A.J., D., A., R., 2007. Beast: Bayesian evolutionary analysis by sampling trees. BMC Evolution Biology 7: 214 .

[2] Anderson, R., May, R., 1991. Infectious Diseases of Humans: Dynamics and Control. Oxford University Press.

[3] Balthrop, J., Forrest, S., Newman, M., Williamson, M., 2004. Email networks and the spread of computer viruses. Science, Vol 304(5670), pp 527-529 .

[4] Barabási, A.L., Albert, R., 1999. Emergence of scaling in random networks. Science, Vol 286, pp 509-512 .

[5] Clauset, A., Shalizi, C., Newman, M., 2009. Power-law distributions in empirical data. SIAM Review 51, 661–703. doi:10.1137/070710111.

[6] Coleman, J., Menzel, H., Katz, E., 1966. Medical Innovations: A Diffusion Study. Bobbs Merrill, New York.

[7] Cummings, D., Burke, D., Epstein, J.M., Singa, R., Chakravarty, S., 2002. Toward a Containment Strategy for Smallpox Bioterror: An Individual-Based Computational Approach. Brookings Institute Press.

[8] D., B., V., C., B., G., H., H., J.J., R., A., V., 2009. The modelling of global epidemics: Stochastic dynamics and predictability. Proceedings of the National Academies of Science USA, Vol 106, pp 21484-21489 .

[9] D.T., H., M., C.T., D.J., S., L., M., J.K., F., J., W., M.E.J., W., 2003. The construction and analysis of epidemic trees with reference to the 2001 uk foot-and-mouth outbreak. Proceedings of the Royal Society B 270: 121–127 .

[10] Dunham, J., 2005. An agent-based spatially explicit epidemiological model in mason. Journal of Artificial Societies and Social Simulation, 9(1): 3 .

[11] Eubank, S., Guclu, H., Kumar, V., Marathe, M., Srinivasan, A., Toroczkai, Z., Wang, N., 2004. Modeling disease outbreaks in realistic urban social networks. Nature, Vol 429, pp 180-184 .

[12] Fajardo, D., Gardner, L., 2012. Inferring contagion patterns in social contact networks with limited infection data. Networks and Spatial Economics doi:10.1007/s11067-013-9186-6.

[13] Ferguson, N., Cummings, D., Fraser, C., Cajka, J., Cooley, P., Burke, D., 2006. Strategies for mitigating an influenza pandemic. Nature, Vol 442, pp 448-452 .

[14] Gardner, L., Fajardo, D., Waller, S., 2012. Inferring contagion patterns in social contact networks using a maximum likelihood approach. Transportation Research Record, journal of the Transportation Research Board .

[15] Garey, M., Johnson, D., 1977. The Rectilinear Steiner Tree Problem is NP-Complete. SIAM Journal on Applied Mathematics 32, 826–834. doi:10.1137/0132071.

[16] Gonzales, M., Hidalgo, C., Barabási, A.L., 2008. Understanding individual human mobility patterns. Nature, Vol 453, pp 479-482 .

[17] Gouveia, L., Magnanti, T.L., 2003. Network flow models for designing diameter-constrained minimum-spanning and steiner trees. Networks 41, 159–173. doi:10.1002/net.10069.

[18] Gouveia, L., Simonetti, L., Uchoa, E., 2011. Modeling hop-constrained and diameter-constrained minimum spanning tree problems as steiner tree problems over layered graphs. Mathematical Programming 128, 123–148. doi:10.1007/s10107-009-0297-2.

[19] Graham, R.L., Hell, P., 1985. On the history of the minimum spanning tree problem. Annals of the History of Computing 7, 43–57. doi:10.1109/MAHC.1985.10011.

[20] Hasan, S., Ukkusuri, S., 2011. A contagion model for understanding the propagation of hurricane warning information. Transportation Research Part B, Vol 45, pp 1590-1605 .

[21] Hwang, F.K., Richards, D.S., 1992. Steiner tree problems. Networks 22, 55–89. doi:10.1002/net.3230220105.

[22] Jombart, T., Eggo, R.M., Dodd, P., Balloux, F., 2009. Spatiotemporal dynamics in the early stages of the 2009 a/h1n1 influenza pandemic. PLoS Currents Influenza. .

[23] Kinney, R., Crucitti, P., Albert, R., Latora, V., 2005. Modeling cascading failures in the north american power grid. The European Physics Journal B, Vol 46(1), pp 101-107 .

[24] Liberti, L., Cafieri, S., Tarissan, F., 2009. Reformulations in mathematical programming : A computational approach, in: Foundations of Computational Intelligence Volume 3 - Global Optimization. Springer.

[25] Murray, J., 2002. Mathematical Biology, 3rd edition. Springer.

[26] Newman, M., Forrest, S., Balthrop, J., 2002. Email networks and the spread of computer viruses. Physical Review E, Vol 66(3) .

[27] P., L., M., S., A., R., 2009. Reconstructing the initial global spread of a human influenza pandemic: A bayesian spatial-temporal model for the global spread of h1n1pdm. PLoS Currents Influenza .

[28] Roche, B., Drake, J., Rohani, P., 2011. An agent-based model to study the epidemiological and evolutionary dynamics of influenza viruses. BMC bioinformatics 12(1):87 .

[29] Rosenwein, M.B., Wong, R.T., 1995. A constrained steiner tree problem. European Journal of Operational Research .

[30] Rosseel, M., 1968. Comments on a paper by romesh saigal: "a constrained shortest route problem". Operations Research 16, pp. 1232–1234.

[31] Sachtjen, M., Carreras, B., Lynch, V., 2000. Disturbances in a power transmission system. Physical Review E, Vol 61(5), pp 4877-4882 .

[32] Saigal, R., 1968. A constrained shortest route problem. Operations Research 16, pp. 205–209.

[33] Santos, M., Drummond, L.M., Uchoa, E., 2010. A distributed dual ascent algorithm for the hop-constrained steiner tree problem. Operations Research Letters 38, 57 – 62. doi:http://dx.doi.org/10.1016/j.orl.2009.09.008.

[34] Sornette, D., 2003. Why Stock Markets Crash: Critical Events in Complex Financial Systems. Princeton University Press.

[35] V., C., A., B., M., B., A., V., 2006. The modelling of global epidemics: Stochastic dynamics and predictability. Bulletin of Mathematical Biology, Vol 68, pp 1893-1921 .

[36] Viger, F., Latapy, M., 2005. Efficient and simple generation of random simple connected graphs with prescribed degree sequence, in: Procedings of the 11th Conferance on Computing and Combinatorics (COCOON), 440–449.

[37] Voss, S., 1999. The steiner tree problem with hop constraints. Annals of Operations Research .

[38] Wallace, R., HoDac, H., Lathrop, R., Fitch, W., 2007. A statistical phylogeography of influenza a h5n1. Proceedings of the National Academies of Science USA, Vol 104(11), pp 4473-4478 .

[39] Yen, J.Y., 1971. Finding the k shortest loopless paths in a network. Management Science .