Low Complexity Method for Simulation of Epidemics Based on Dijkstra’s Algorithm

Davide Zorzenon, Fabio Molinari, Jörg Raisch

Abstract—Models of epidemics over networks have become popular, as they describe the impact of individual behavior on infection spread. However, they come with high computational complexity, which constitutes a problem in case large-scale scenarios are considered. This paper presents a discrete-time multi-agent SIR (Susceptible, Infected, Recovered) model that extends known results in literature. Based on that, using the novel notion of Contagion Graph, it proposes a graph-based method derived from Dijkstra’s algorithm that allows to decrease the computational complexity of a simulation. The Contagion Graph can be also employed as an approximation scheme describing the “mean behavior” of an epidemic over a network and requiring low computational power. Theoretical findings are confirmed by randomized large-scale simulation.

I. INTRODUCTION

Mathematical models of epidemics are essential tools for forecasting the spread of diseases. Recently, the COVID-19 pandemic has shown the importance of such instruments for planning control measures and assessing their impact [7], [2]. Among the frameworks available in literature, network-based models are particularly suitable for evaluating non-pharmaceutical interventions, e.g., social distancing, since they provide a natural representation of contact interactions between individuals (see, e.g., Fig. 1) [13], [16]. However, computational complexity is an issue when large-scale scenarios are considered [6]. Consequently, researchers have suggested several strategies aiming to reduce the computational burden while maintaining adequate levels of accuracy.

For instance, [8] introduces a dynamical discrete-time Susceptible, Infected, Recovered (SIR) model with Boolean algebra formalism, in which operations can be efficiently implemented. SIR models, in which individuals are assumed to get immune after recovering from the disease, have been widely used to describe infectious diseases [3], [10]. However, the limitation of discrete-time simulations comes from the synchronous updating, in which the state of the system is updated at regular time intervals. Since the time step has to be small enough to capture high frequency phenomena, the state is often refreshed even when no new events occur, causing a waste of computational resources. Event-based methods, as the Gillespie algorithm, overcome this problem by updating the state only when a new event occurs [14], [1].

In our paper, we first present a discrete-time SIR model of the disease evolution. Our model is a dynamical and stochastic generalization of the Boolean model described in [8], which also captures the possibility that individuals can be infected from outside the investigated network at an arbitrary time. Next, we introduce the so-called Contagion Graph, a graph obtained from the interaction network, from which we formally derive an event-driven procedure to simulate the model. We analytically show that the procedure, based on Dijkstra’s algorithm, significantly reduces the computational complexity, thus simulation time.

Another open issue in modeling epidemics over networks is the development of approximation schemes, whose aim is to describe the “mean behavior” of the disease evolution [11]. One of their advantages is the ability to provide insights of the stochastic process without the need of interpolating large amounts of simulation results [12]. We will show that the flexible nature of the Contagion Graph allows to formulate an approximate model based on statistical evidence. Both applications of the Contagion Graph are shown by employing randomized simulations.

The remainder of this paper is structured as follows. In Section II the dynamical system modeling the epidemic spread is presented. We compare our model to the one of [8] in Section III. Section IV presents the Contagion Graph, which allows for both complexity reduction during simulation and “mean behavior” analysis. Concluding remarks are given in Section V.
A. Notation
Throughout this paper, \( \mathbb{N}_0 \) denotes the set of nonnegative integers, \( \mathbb{N} \) the set of positive integers, and \( \mathbb{R} \) the set of real numbers. A graph is a pair \((\mathcal{N}, \mathcal{A})\) where \( \mathcal{N} \) is the set of nodes and \( \mathcal{A} \) is the set of arcs. If the graph is undirected, \( \mathcal{A} \) is the set of all two-element subsets \( \{i, j\} \) of \( \mathcal{N} \), so that there is an arc between node \( i \) and node \( j \). If the graph is directed, \( \mathcal{A} \) is the set of all pairs \((i, j)\), so that there is a directed arc from node \( i \) to node \( j \).

II. Problem Description

A. Multi-agent SIR model
Consider \( \mathcal{N} \), a set of agents labeled 1 through \( n \in \mathbb{N} \) describing the whole population in which an infection is spreading. Each agent (also referred to as individual) can interact with other agents at discrete-time steps (or iterations). Assume the underlying network topology modeling such interactions to be an undirected graph, i.e., at every iteration \( k \in \mathbb{N}_0 \), \( \mathcal{G}(k) := (\mathcal{N}, \mathcal{A}(k)) \). In particular, the set of neighbors of agent \( i \in \mathcal{N} \) at iteration \( k \in \mathbb{N}_0 \) is
\[
\mathcal{N}_i(k) := \{ j \in \mathcal{N} \mid \{j, i\} \in \mathcal{A}(k)\},
\]
and represents the set of all agents that can interact with agent \( i \) at iteration \( k \). Note that, by definition, self-arcs are not considered. The infection can spread from one individual (agent) to one or more of its neighbors in the graph \( \mathcal{G}(k) \), and this way through the whole population (set).

To this end, let \( x_i : \mathbb{N}_0 \rightarrow \{0, 1\} \) describe if an agent, say \( i \in \mathcal{N} \), is found to be infected at iteration \( k \in \mathbb{N}_0 \). In particular,
\[
x_i(k) = 1
\]
means that agent \( i \) is infected at iteration \( k \), otherwise \( x_i(k) = 0 \). In what follows, let \( \mathcal{N}_i(k) \subseteq \mathcal{N}_i(k) \) be defined as the subset of neighbors of agent \( i \) at iteration \( k \) that are infected. Formally,
\[
\mathcal{N}_i(k) := \{ j \in \mathcal{N} \mid \{j, i\} \in \mathcal{A}(k), x_j(k) = 1\}.
\]
If \( x_i(k) = 0 \), agent \( i \) could be either susceptible (namely, it has never got in contact with the infection) or recovered (namely, it has already got in contact with the infection, towards which it has developed immunity). Let variable \( y_i(k) \) capture whether an individual \( i \) is recovered at iteration \( k + 1 \), i.e.,
\[
y_i(k) = 1
\]
means that agent \( i \) is recovered at iteration \( k + 1 \), otherwise \( y_i(k) = 0 \), namely \( i \) is susceptible or infected at iteration \( k + 1 \). This clearly implies, \( \forall i \in \mathcal{N}, \forall k \in \mathbb{N}_0 \),
\[
y_i(k) = 1 \implies x_i(k + 1) = 0. \quad (1)
\]
It is also assumed that immunity lasts forever, i.e., \( \forall k \in \mathbb{N}_0 \),
\[
y_i(k + 1) \geq y_i(k). \quad (2)
\]
Let, \( \forall i \in \mathcal{N}, k_i \) define the first time step when agent \( i \) is found to be infected, i.e.,
\[
k_i := \inf\{ k \in \mathbb{N}_0 \mid x_i(k) = 1\}. \quad (3)
\]

In this paper, we use the convention that \( \inf \emptyset = \infty \); therefore, if agent \( i \) does not get in contact with the infection, then \( k_i = \infty \). Agent \( i \)'s recovery time is the number of iterations from \( k_i + 1 \) until agent \( i \) is recovered, and is denoted by \( R_i \in \mathbb{N} \). Formally, \( \forall i \in \mathcal{N}, R_i \) is the smallest integer such that
\[
y_i(k_i + R_i) = 1.
\]
The system is heterogeneous, i.e., agents may have different recovery times.

B. Dynamics
Consider a pair of agents, i.e., \( \{i, j\} \subset \mathcal{N} \), such that \( \{i, j\} \in \mathcal{A}(k) \) at a given iteration \( k \). Agent \( i \) is infected \( (x_i(k) = 1) \) whilst agent \( j \) is susceptible. The probability that agent \( i \) infects its neighbor agent \( j \) at time \( k \) is \( p_{ij}(k) \in [0, 1] \). Formally,
\[
P(x_j(k + 1) = 1 \mid x_i(k) = 0, x_i(k) = 1, y_j(k) = 0, \mathcal{N}_j(k) = \{i\}) = p_{ij}(k). \quad (5)
\]
We model this behavior by defining a random variable drawn out of a Bernoulli distribution with probability \( p_{ij}(k) \), i.e.,
\[
\forall k \in \mathbb{N}_0, \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i(k), \xi_{ij}(k) \sim \mathcal{B}(p_{ij}(k)),
\]
called contagion coefficient. We assume that, \( \forall \{i, j\} \in \mathcal{A}(k) \),
\[
\xi_{ij}(k) = \xi_{ji}(k). \quad \text{If } \xi_{ji}(k) = 1, \text{ an infected agent } i \text{ infects a}
\]

Fig. 2: Infection spreading with \( p_{ij} = 0.2 \) and \( R_i \in [3, 5] \).

Fig. 3: Infection spreading with \( p_{ij} = 0.2 \) and \( R_i \in [3, 30] \).

Fig. 4: Infection spreading with \( p_{ij} = 0.5 \) and \( R_i \in [3, 5] \).
susceptible neighbor agent \( j \) at iteration \( k \). Formally, \( \forall k \in \mathbb{N}_0, \forall \{i, j\} \in \mathcal{A}(k), \)

\[
\begin{align*}
\xi_{ij}(k) &= 1 \\
x_i(k) &= 1 \implies x_j(k+1) = 1. \\
y_j(k) &= 0
\end{align*}
\]

Consider, e.g., the case of an infection spreading across a population. Each contagion coefficient models whether two neighboring agents, at a given time step, have a contact, which would cause a contagion in one of the two is infected.

We also consider that the infection can hit an individual without being spread from one infected agent in the network, but rather from an external injection. To this end, let

\[ u_i(k) = 1 \]

denote that agent \( i \) gets in contact with the infection (from outside of the network) at time \( k \in \mathbb{N}_0 \). Traditional models considering closed populations (see, e.g., [8]) can be represented by having \( u_i(0) = 1 \), with agent \( i \) defined traditionally as patient zero. Note that, \( \forall i \in \mathcal{N}, \forall k \in \mathbb{N}_0, \)

\[
\begin{align*}
y_i(k) &= 0 \\
u_i(k) &= 1 \implies x_i(k+1) = 1. 
\end{align*}
\]

**Definition 1.** Agent \( i \) is closed towards external infection if, \( \forall k \in \mathbb{N}_0, u_i(k) = 0 \). Otherwise, agent \( i \) is called open towards external infections.

Once an agent, say \( i \in \mathcal{N} \), is infected, we know that it will recover in \( R_i \) iterations. We define a variable, referred to as infection stopwatch, formally defined as, \( \forall i \in \mathcal{N}, \forall k \in \mathbb{N}_0, \)

\[
s_i(k) := \begin{cases} 0 & \text{if } k \leq k_i \\ R_i + 1 & \text{if } k \geq k_i + R_i + 1 \\ k - k_i & \text{otherwise} \end{cases}
\]

Such a variable counts the time steps since agent \( i \) got infected. If no infection is developed at time \( k \), then this variable is 0. If agent \( i \) has already recovered at time \( k \), this variable equals \( R_i + 1 \).

With all such notions at hand, the dynamics of each agent \( i \in \mathcal{N} \) evolves according to the non-linear discrete-time system

\[
\begin{align*}
\begin{cases} x_i(k+1) = \varrho(R_i - s_i(k)) x_i(k) + \sum_{j \in \mathcal{N}(i)} \xi_{ij}(k) x_j(k) + u_i(k) \\ s_i(k+1) = s_i(k) + x_i(k) \end{cases}
\end{align*}
\]

where \( \varrho : \mathbb{R} \mapsto \{0, 1\} \) is the step function

\[
\varrho(\varnothing) = \begin{cases} 0 & \text{if } \varnothing \leq 0 \\ 1 & \text{if } \varnothing > 0 \end{cases}
\]

In what follows, let \( x(k) \) and \( s(k) \) be \( n \)-dimensional vectors stacking, respectively, the infection variables and the infection stopwatch variables of all agents at time step \( k \), i.e., \( \forall k \in \mathbb{N}_0, \forall i \in \{1, \ldots, n\}, \)

\[
[x(k)]_i = x_i(k), \quad [s(k)]_i = s_i(k).
\]

**Proposition 1.** System (9) guarantees that properties (1), (2), (4), (6)-(8) are satisfied, \( \forall i \in \mathcal{N}, \forall k \in \mathbb{N}_0. \)

**Proof.** The proof is omitted due to space limitation.

**Example 1.** Consider the static network \( (\mathcal{N}, \mathcal{A}) \), depicted in Figure 1 in which an infection is spreading with dynamics (9). Agent 1 gets in contact with the infection at time 0, i.e., \( u_1(k) = 0 \). We simulate three different scenarios in order to address the impact that different parameters have on the infection spreading.

Figure 2 let, \( \forall (j,i) \in \mathcal{A}, \forall k \in \mathbb{N}_0, p_{ij}(k) = 0.2 \) and \( \forall i \in \mathcal{N}, R_i \in [3, 5] \) (randomly extracted from this set). We can observe the infection dynamics in the figure. Some agents never get infected. In fact, after time \( k = 40 \), a subset remains susceptible and the infection disappears.

Figure 3 in this case, we increment the maximum recovery time to 30 iterations. This implies, as in the figure, that the peak of infection is wider and longer-lasting. Also, the whole population gets in contact with the disease.

Figure 4 on the other hand, if we increment the possibility of infecting another agent (namely, \( \forall (j,i) \in \mathcal{A}, \forall k \in \mathbb{N}_0, p_{ij}(k) = 0.5 \)), but we keep the maximum recovery time to 5 iterations, we obtain a fast infection spread, in which the whole population is infected, but the infection disappears after 30 steps.

Running each one of these simulation takes on average 0.95s on a machine mounted Intel i7 at 2.90GHz with Python 3.6. The running time increases with the simulation horizon, which must be set large enough to capture the entire disease evolution. As shown in the examples, an adequate choice of the simulation horizon greatly depends on parameters of individuals such as \( p_{ij}(k) \) and \( R_i \). This proves that, despite the benefits of agent-based models, the time needed for simulation diverges, not only with the network’s size, but also with the size of parameters of agents. As motivated by [11], this paper is concerned with finding a formal method to run multi-agent simulation on networks with inexpensive computational effort.

**Remark 1.** Note that \( x_i(k), y_i(k), k_i, \) and \( s_i(k) \) are random variables, functions of contagion coefficients.

## III. Model Dynamics with Boolean Algebra

In the following, we show that system (9) can be rewritten as a dynamical system in the Boolean algebra. This proves that our model is a stochastic dynamical generalization of [8, Eq. (3)]. In fact, our system is represented in the formalism of dynamical systems, whilst [8, Eq. (3)] has a state vector whose dimension increases with iterations. Moreover, in [8, Eq. (3)] contagion is deterministic, whereas in (9) it depends, at every time step \( k \in \mathbb{N}_0 \), on the stochastic realizations of variables \( \{\xi_{ij}(k)\}_{i,j \in \mathcal{A}(k)} \). Moreover, [8, Eq. (3)] does not consider infections originating outside the considered network, but only epidemics within a closed population.

**Remark 2.** Due to space limitation, in this section we consider (9) without the presence of \( u_i(k) \).
A. Fundamentals of Boolean Algebra

Consider two Boolean variables $a$ and $b$. We define, respectively, conjunction, disjunction and negation operations as, $\forall a \in \{0,1\}$, $\forall b \in \{0,1\}$,

$$a \lor b = \max(a, b), \quad a \land b = \min(a, b), \quad \neg a = 1 - a.$$ 

We extend the operations to matrices. To this end, consider three Boolean matrices, i.e., $A, B \in \{0,1\}^{n \times m}$, $C \in \{0,1\}^{m \times p}$. Disjunction, conjunction, and negation operations are defined as

$$[A \lor B]_{ij} = a_{ij} \lor b_{ij}, \quad i = 1 \ldots n, \quad j = 1 \ldots m,$$

$$[A \land C]_{ij} = \bigwedge_{h=1}^{m} a_{ih} \land b_{hj}, \quad i = 1 \ldots n, \quad j = 1 \ldots p,$$

$$[-A]_{ij} = \neg a_{ij}, \quad i = 1 \ldots n, \quad j = 1 \ldots m.$$ 

Furthermore, let us define the element-wise conjunction operation (Hadamard product in the Boolean algebra) $\odot$, i.e.,

$$[A \odot B]_{ij} = a_{ij} \land b_{ij}, \quad i = 1 \ldots n, \quad j = 1 \ldots m.$$ 

B. Infection Model in the Boolean Algebra

We define the following dynamical system in the Boolean algebra with $k \in \mathbb{N}_0$ the iteration index:

$$\begin{align*}
    x^h(k+1) &= (\Xi(k) \land x^h(k)) \odot (\neg y^h(k)) \\
y^h(k+1) &= y^h(k) \lor x^h(k-R+1)
\end{align*}$$

where

- $x^h(k) \in \{0,1\}^n$ is the Boolean infection vector, such that $x^h_i(k)$ is 1 if individual $i$ is infected at time $k$, 0 otherwise;
- $y^h(k) \in \{0,1\}^n$ is the Boolean recovery vector, such that $y^h_i(k)$ is 1 if individual $i$ is recovered at time $k$, 0 otherwise;
- $\Xi(k) \in \{0,1\}^n \times n$, such that

$$\Xi_{ij}(k) := \begin{cases} 
    \xi_{ij}(k) & \text{if } \{i,j\} \in A(k) \\
    1 & \text{if } i = j \\
    0 & \text{otherwise}
\end{cases}$$

- $x^h(k-R+1)$ is short hand notation for the $n$-dimensional vector of elements $[x^h(k-R+1)]_i = x^h_i(k-R+1), \quad i = 1 \ldots n$.

To initialize (10) we define $x^h_0(k) = 0, \forall k < 0$. Note that, if $y^h(0) = 0$, the second equation of system (10) can be written as

$$y^h(k+1) = \bigvee_{t=0}^{k-R+1} x^h(t).$$

In this case, and considering all contagion coefficients equal to one, i.e., $\forall i,j \in A(k)$, $\xi_{ij}(k) = 1$, it is immediate to see the equivalence of system (10) with [8, Eq. (3)].

Remark 3. The state of system (10) at iteration $k$ is

$$x^h_s(k) := [x^h_1(k), \ldots, x^h_i(k-R+1), \ldots, x^h_n(k), \ldots, x^h_{n}(k-R_n+1), y^h_1(k), \ldots, y^h_n(k)]^T \in \{0,1\}^{n_s},$$

with $n_s = n + \sum_{i \in \mathcal{N}} R_i$. Unlike [8, Eq.(3)], in system (10) the state dimension does not grow with time, but has dimension $n_s$.

Remark 4. The state-space cardinality for system (10) is $2^{n_s}$. It is immediate to show that system (9) has a lower state-space cardinality than system (10) if

$$\sum_{i \in \mathcal{N}} \log_2(R_i + 2) \leq \sum_{i \in \mathcal{N}} R_i.$$

In what follows, we prove that (10) can be seen as the Boolean formulation of (9).

Proposition 2. Assume, $\forall i \in \mathcal{N}$,

$$x_i(0) = x^h_i(0), \quad s_i(0) = y^h_i(0) = 0,$$

and, by definition,

$$x^h_i(k) = 0 \quad \forall k < 0.$$ 

Vectors $x(k)$ and $s(k)$, respectively $x^h(k)$ and $y^h(k)$, evolving according to (9), respectively (10), satisfy, $\forall k \in \mathbb{N}_0$, $\forall i \in \mathcal{N}$,

$$x_i(k) = x^h_i(k) \quad \text{and} \quad s_i(k) \geq R_i \iff y^h_i(k) = 1. \tag{11}$$

Proof. The proof follows by strong induction.

Let the induction statement be (11). The base case, for $k = 0$, is trivially verified in the hypothesis.

The inductive hypothesis is that, $\forall k \leq h, \quad h \in \mathbb{N}_0$, (11) is true. Thus, we need to prove that (11) must also be true for $k = h+1$.

First of all, let us note that, by the inductive hypothesis, $\forall i \in \mathcal{N}$,

$$\varrho(x_i(h) + \sum_{j \in \mathcal{N}_i(h)} \xi_{ij}(h)x_j(h)) = \left( x_i(h) \lor \bigvee_{j \in \mathcal{N}_i(h)} \xi_{ij}(h) \land x_j(h) \right) = [\Xi(h) \land x(h)]. \tag{12}$$

By the induction hypothesis and by definition of $\varrho(o)$, we have, $\forall i \in \mathcal{N}$,

$$\varrho(R_i - s_i(h)) = 0 \iff y^h_i(h) = 1,$$

equivalently, $\forall i \in \mathcal{N}$,

$$\varrho(R_i - s_i(h)) = \neg y^h_i(h). \tag{13}$$

By bringing together (12) and (13), we conclude that, by the inductive hypothesis, $\forall i \in \mathcal{N}$,

$$x_i(h+1) = x^h_i(h+1). \tag{14}$$

At this point, to conclude the proof, we need to prove that the induction hypothesis also implies, $\forall i \in \mathcal{N}$,

$$s_i(h+1) \geq R_i \iff y^h_i(h+1) = 1. \tag{15}$$

To this end, note that, by (9), $\forall i \in \mathcal{N}$,

$$s_i(h+1) \geq R_i \iff x_i(\ell - R_i + 1) = 1 \quad \text{for some } \ell \in \{R_i - 1, \ldots, h\}. \tag{16}$$
By the induction hypothesis, \( \forall i \in \mathcal{N}, \)
\[
x_i(\ell - R_i + 1) = x^b_i(\ell - R_i + 1) = 1 \implies y^b_i(h + 1) = 1, \quad (18)
\]
for some \( \ell \in \{R_i - 1, \ldots, h\} \). By bringing together (16), (17), and (18), under the induction hypothesis, (15) is verified. This proves (11) for \( k = h + 1 \), thus implying, by strong induction, that (11) always holds. This concludes the proof. \( \square \)

IV. Contagion Graph

As motivated in literature and seen in Example 1, a common issue for models of epidemics over networks is the computational burden. In this section, we investigate a possible way to obtain simulation results for system (9) with reduced computational complexity.

A. Contagion Graph as equivalent model

In order to ease the explanation, consider these two assumptions that will be relaxed in future work.

Assumption 1. The network topology is constant over time, i.e., \( \forall k \in \mathbb{N}_0, A(k) = A \).

Assumption 2. The contagion coefficients of each pair of agents is constant through time, i.e., \( \forall k \in \mathbb{N}_0, \forall \{j, i\} \in A, \xi_{ij}(k) = \xi_{ij} \).

Assumption 1 corresponds to the case in which neither restricting measures are taken in order to contain the contagion, nor new connections between individuals can be established. With Assumption 2, we consider the probability of interaction between any pair of individuals to be constant over time.

Consider \( k_i \) as defined in \( \tau \). Set \( \{k_i\}_{i \in \mathcal{N}} \) provides a full description of epidemics; indeed, if \( k_i \in \mathbb{N}_0 \), individual \( i \) gets infected at time \( k_i \) and recovers at time \( k_i + R_i + 1 \), while, if \( k_i = \infty \), agent \( i \) never gets infected.

Definition 2. The random variable \( \tau_{ij}(k) \) denotes, \( \forall \{i, j\} \in A \), the time-steps that agent \( j \in \mathcal{N}_i \) takes to infect its neighbor \( i \), assuming \( k_j = k \), and ignoring the effect of other individuals. Formally,
\[
\tau_{ij}(k) := 1 + \inf\{h \in \{k, \ldots, k + R_j - 1\} \mid \xi_{ij}(h) = 1\}. \quad (19)
\]

Definition 3. Given an agent \( i \), variable \( k^{ext}_i \in \mathbb{N}_0 \) denotes the time of a possible external infection, i.e.,
\[
k^{ext}_i := 1 + \inf\{k \in \mathbb{N}_0 \mid u_i(k) = 1\}. \quad (20)
\]

It is clear by definition that agents closed towards external infections have \( k^{ext}_i = \infty \).

Definition 4. Let \( \mathcal{N}^{ext} \subseteq \mathcal{N} \) denote the set of agents open towards external infections, i.e.,
\[
\mathcal{N}^{ext} := \{i \in \mathcal{N} \mid k^{ext}_i \in \mathbb{N}_0\}. \quad (21)
\]

Let now \( \tilde{\mathcal{N}} \) be a new set of nodes labeled with negative numbers, such that each node in \( \mathcal{N}^{ext} \) corresponds to one node in \( \tilde{\mathcal{N}} \), i.e.,
\[
\tilde{\mathcal{N}} := \{-i \mid i \in \mathcal{N}^{ext}\}.
\]

Lemma 1. Under Assumptions 1-2, \( \tau_{ij}(k) \) is a stationary process with probability distribution
\[
P(\tau_{ij} = \tau) = \begin{cases} p_{ij}(1 - p_{ij})^{\tau - 1} & \text{if } \tau \in \{1, \ldots, R_j\} \\ (1 - p_{ij})^{R_j} & \text{if } \tau = \infty \\ 0 & \text{otherwise} \end{cases}. \quad (22)
\]

Proof. If \( \tau \in \{1, \ldots, R_j\} \), then \( \tau_{ij}(k) = \tau \) corresponds to event \( \xi_{ij}(k) = \ldots = \xi_{ij}(k + \tau - 1) = 0, \xi_{ij}(k + \tau) = 1 \). Since \( \xi_{ij}(k) \) is a Bernoulli distributed random variable, in this case \( \tau_{ij}(k) \) assumes the geometric distribution \( Geo(p_{ij}) \), as in the first line of (22). Note that for \( \tau \in \mathbb{N} \setminus \{1, \ldots, R_j\} \), \( \tau_{ij}(k) = \tau \) has zero probability, since it would imply \( h > k + R_j - 1 \) in (19). Therefore, event \( \xi_{ij}(k) = \ldots = \xi_{ij}(k + R_j - 1) = 0 \), which has probability \( (1 - p_{ij})^{R_j} \), corresponds to
\[
\tau_{ij}(k) = 1 + \inf_{h \in \{0, \ldots, R_j - 1\} \cap (\mathbb{N} \setminus \{1, \ldots, R_j\})}(\xi_{ij}(k + h - 1)) = 1 + \inf_{h \in \mathbb{N}}(h = \infty).
\]

This concludes the proof. \( \square \)

Theorem 1. Under Assumptions 1-2, for any agent \( i \) closed towards external infections, we have
\[
k_i := \inf_{j \in \mathcal{N}_i} \left( k_j + \phi_{R_j} \left( \left\lfloor \log_{1-p_{ij}}(u_{ij}) \right\rfloor \right) \right), \quad (23)
\]
where
\[
\phi_{R_j}(\alpha) := \begin{cases} 0 & \text{if } \alpha \leq R_j \\ \infty & \text{otherwise} \end{cases}.
\]

and \( u_{ij} \sim \mathcal{U}(0, 1) \).

Proof. By incorporating (6) into (4), for \( i \) being an agent closed towards external infection, one has
\[
k_i = \inf_{j \in \mathcal{N}_i} \{k \in \mathbb{N}_0 \mid \xi_{ij}(k - 1) = 1, x_j(k - 1) = 1\}. \quad (24)
\]

Also by (3) and definition of recovery time, the latter is equivalent to
\[
k_i = \inf_{j \in \mathcal{N}_i} \{k \in \mathbb{N}_0 \mid \xi_{ji}(k - 1) = 1, 1 \leq k - k_j \leq R_j + 1\},
\]
that, by considering (19), can be rewritten as
\[
k_i := \inf_{j \in \mathcal{N}_i} \{k_j + \tau_{ij}(k_j)\}. \quad (25)
\]

By Lemma 1 being \( \tau_{ij} \) stationary,
\[
k_i := \inf_{j \in \mathcal{N}_i} \{k_j + \tau_{ij}\}. \quad (26)
\]
A realization of \( \tau_{ij} \), by (20), is determined by applying a threshold \( \phi_{R_j}(\cdot) \) on a realization of a random variable, say \( \tau_{ij} \), with geometric distribution, i.e., \( \tau_{ij} \sim Geo(p_{ij}) \).

\( ^1 \)The symbol \( \overset{d}{=} \) denotes equality in distribution.
An algorithm for sampling a geometric random variable in constant time is given in literature, see, e.g., [5, section X.2]. Given a realization of a uniformly distributed random variable \( u_{ij} \sim U(0,1) \), a sample of \( \tilde{\tau}_{ij} \) is computed exactly using formula \( \left\lceil \log_{1-p_{ij}} u_{ij} \right\rceil \), where \( \lceil \cdot \rceil \) is the ceiling function. By incorporating

\[
\tau_{ij} \overset{d}{=} \phi_{R_j} \left( \left\lceil \log_{1-p_{ij}} u_{ij} \right\rceil \right) \tag{23}
\]

into (22), the proof is concluded. \( \square \)

**Corollary 1.** Under Assumptions 1-2 for any agent \( i \) closed towards external infections, we have

\[
k_i \overset{d}{=} \inf \left\{ \inf_{j \in N_i} \left( k_j + \phi_{R_j} \left( \left\lceil \log_{1-p_{ij}} u_{ij} \right\rceil \right) \right), k_i^{ext} \right\} \tag{24}
\]

for \( u_{ij} \sim U(0,1) \).

**Proof.** Consider (22). In case of an agent open towards external infection, we could rewrite it as

\[
k_i \overset{d}{=} \inf \left\{ \inf_{j \in N_i} \left( k_j + \phi_{R_j} \left( \left\lceil \log_{1-p_{ij}} u_{ij} \right\rceil \right) \right) + \kappa_i^{ext}, \kappa_i^{ext} \right\} \tag{25}
\]

By incorporating (23) into the latter, the proof immediately follows. \( \square \)

We have gathered all notions needed for defining the Contagion Graph associated to network \((N, A)\) in which an infection is spreading across nodes with dynamics modeled by (9).

**Definition 5.** The Contagion Graph of network \((N, A)\) with infection dynamics (9) is a directed graph with random weights whose nodes are \( N \cup N^\circ \). The set of arcs is \( A_1 \cup A_2 \), where

\[
A_1 := \{(j, i) \mid \{i, j\} \in A, \tau_{ij} \in N_0 \}
\]

and

\[
A_2 := \{(-i, i) \mid i \in N^\circ \}.
\]

Arc weights are, \( \forall (j, i) \in A_1, \tau_{ij} \), and, \( \forall (-i, i) \in A_2, k_i^{ext} \). Weights \( \tau_{ij} \) are random variables whose distribution is as in (23).

**Example 2.** Consider a network of 5 agents with infection dynamics (9) as in Fig. 5a. The recovery time is supposed to be 3 time steps for each individual, i.e., \( \forall i \in N \), \( R_i = 3 \), and the time of infection from external sources are \( k_i^{ext} = 1 \), \( k_i^{1ext} = 3 \). From these parameters, it is possible to draw the Contagion Graph in Fig. 5b where nodes of set \( N = \{1, 2, 3, 4, 5\} \) are in white and the ones of set \( N^\circ = \{-1, -4\} \) are in red; note that weights \( \tau_{ij} \) are random variables, while \( k_i^{ext} \) are deterministic. Assuming that, \( \forall \{i, j\} \in A, p_{ij} = 0 \), a realization for random variables \( \tau_{ij} \) is computed as in (23). The corresponding realization of the Contagion Graph is shown in Fig. 5c. Here, the arcs that are missing from the Contagion Graph correspond to a realization \( \tau_{ij} = \infty \).

The Contagion Graph provides an efficient method to compute all \( \{k_i\}_{i \in N} \), as explained by the following result.

**Theorem 2.** For each agent \( i \in N \), the random variable \( k_i \) is equal in distribution to the minimum weighted path on the Contagion Graph from any node in \( N^\circ \) to \( i \).

**Proof.** Consider (25). This can be expanded as, \( \forall i \in N \),

\[
k_i \overset{d}{=} \inf \left\{ \inf_{j \in N_i} \{k_f + \tau_{fj} + \tau_{ij}, k_j^{ext} + \tau_{ij}\}, k_i^{ext}\right\}
\]

\[
\overset{d}{=} \inf \left\{ k_f + \tau_{fj} + \tau_{ij}, k_j^{ext} + \tau_{ij}, k_i^{ext}\right\}.
\]

One can see that \( k_i \) is equal in distribution to the minimum between the path from \(-i \in N^\circ \) to \( i \in N \), the path from \(-j \in N^\circ \) to \( i \in N \), and \( k_f \) plus the path from \( \ell \in N \) to \( i \in N^\circ \). By doing this recursively, one obtains

\[
k_i \overset{d}{=} \inf_{\{h_1, \ldots, h_L\} \subseteq N^\circ} \left\{ k_i^{ext} + \sum_{m=2}^L \tau_{h_m h_{m-1}} \right\}, \tag{26}
\]

which is, by Definition 2, the minimum path going from one node \(-h_1 \in N^\circ \) to \( i \in N \). \( \square \)

By this latter Theorem, in order to simulate the epidemics’ dynamics (9) over a network \((N, A)\), it is sufficient to obtain a realization of the Contagion Graph and compute (26) for each node. As it will be shown in Section IV-B, this allows to decrease the computational complexity if compared to running (9).

**Corollary 2.** For a realization of the Contagion Graph, we can compute the number of infected agents at every time \( k \in N_0 \) as

\[
\sum_{i \in N} x_i(k) = \sum_{i \in N} \varrho(k - k_i + 1) - \varrho(k - k_i - R_i).
\]

**Proof.** The proof is omitted due to space limitation. \( \square \)
Example 3 (Continuation of Example 2). Given the realization of Fig. 5c, we can determine the evolution of the disease spread by computing the paths of minimum weight from any node of $\hat{N}$ to any node of $N$. By solving the single-source shortest path problem from nodes $-1$ and $-4$, and then taking the path of minimum weight between the two, we get $k_1 = 1$, $k_2 = 4$, $k_3 = 7$, $k_4 = 3$, $k_5 = 4$.

B. Computational complexity

1) Dynamics: in order to compute the disease evolution using (9), we need $n(n+4)$ operations for every time step. Note that the number of time steps needed to capture the entire evolution depends on the size of parameters $p_{ij}$, $R_i$ and $k_i^{\text{ext}}$; hence, with this approach, which is the same described in [8], we can compute $k_i$ for all nodes only in weakly polynomial time complexity of $O(n^2T)$, where $T \in \mathbb{N}$ is the simulation horizon.

2) Contagion Graph: Obtaining a realization of the Contagion Graph requires at most a computational complexity equal to $O(n^2)$. In fact, drawing the Contagion Graph requires to compute (23) for all elements of set $\hat{N} \times \hat{N}$. By Theorem 2, for each agent $i \in \hat{N}$, $k_i$ can be computed by solving a single-source shortest path problem from every node in $\hat{N}$ to every node in $\hat{N}$. Since the weights of every Contagion Graph realization are non-negative, this can be done by applying Dijkstra's algorithm $|\hat{N}|$ times. Note that the complexity of one run of Dijkstra's algorithm equals $O(n^2)$. Hence, the overall complexity of employing the Contagion Graph for computing $\{k_i\}_{i \in \hat{N}}$ is $O(n^2 + n^2|\hat{N}|) = O(n^2|\hat{N}|)$, thus strongly polynomial in time, see, e.g., [4].

Given that, in general, $T \gg |\hat{N}|$, using the Contagion Graph greatly improves simulation performance.

Example 4 (Continues from Example 1). Running the same simulation as Fig. 2 with the Contagion Graph takes 0.05s on the same machine.

C. Contagion Graph as approximated model

The Contagion Graph can be also a computationally inexpensive method to assess the “mean behavior” of an epidemics over a network. We formulate the approach, described also in [9, section 2.1] in the case of static networks; future work will extend the same strategy to the dynamical case.

Consider an arc, say $(j, i)$, of the Contagion Graph. Its weight is a random variable (moreover, in case its realization is infinite, the arc is dropped). We aim at estimating the value of this arc weight by fitting some parameters. In fact, given a parameter $\beta \in [0, 1]$, the estimated arc weight $\hat{\tau}_{ij} \in \mathbb{N} \cup \{\infty\}$ is the minimum value greater than $\tau_{ij}$ with probability higher than $\beta$. Formally, $\forall (i, j) \in A_1$,

$$\hat{\tau}_{ij} := \inf\{\tau \in \mathbb{N} \mid P(\tau \geq \tau_{ij}) \geq \beta\}.$$  \hspace{1cm} (27)

Different choices of $\beta$ correspond to different approximations of $\tau_{ij}$. By (20) and (27), one can obtain an explicit formula for $\hat{\tau}_{ij}$, i.e., \[\hat{\tau}_{ij} = \phi_{R_{ij}} \left( \left\lceil \log_{1-p_{ij}} (1-\beta) \right\rceil \right).\]

We build a Contagion Graph with arcs having weights equal to $\hat{\tau}_{ij}$ (note that if the weight is infinite, the corresponding arc is dropped). Thus, we can obtain an estimation of set $\{k_i\}_{i \in \hat{N}}$ (depending on $\beta$), by solving the shortest path problem, as in Section IV-C. Clearly, these estimates are computed in the same time complexity of a single simulation.

Example 5. Consider the network in Fig. 3 in which $u_1(0) = 1$. We consider two different scenarios for the problem: (i) $p_{ij} = 0.8$ and $R_i \in [3, 20]$, (ii) $p_{ij} = 0.2$ and $R_i \in [3, 20]$. For each scenario, we run 400 Montecarlo simulations employing dynamics (9) and one realization of the Contagion Graph following the idea in Section IV-C with $\beta = 0.5$. We compare results of the Contagion Graph with results from the discrete-time simulation, thus showing that the Contagion Graph could be used as an approximated model. In fact, in Fig. 7 (scenario (i)), one can see that the Contagion Graph represents the "mean behavior" of the randomized simulations. By decreasing the infectivity (scenario (ii)), many more nodes will happen to be non-infected through simulations. This is the case of Fig. 8 in which the Contagion Graph captures the fact that some nodes are not expected to get infected, and these correspond to the nodes that are found to be non-infected most times in the Monte Carlo simulations.

V. CONCLUSION AND FUTURE WORK

This paper has considered an agent-based model of epidemics over complex networks and has provided a novel method for simulating epidemics with lower computational complexity. The proposed method is based on a graph-based formalization of the problem and can be also employed for estimating the mean behavior of the epidemic.

3Relaxing the condition of discrete $\tau_{ij}$, the ceiling function can be removed.
Future work will aim at extending the present work to the case of time-varying contact networks, and at developing fast control actions based on the **Contagion Graph** for containing epidemics on large-scale networks. By using available open data, we aim to employ our method to investigate COVID-19 scenarios.

**REFERENCES**

[1] Rehan Ahmad and Kevin S Xu. Continuous-time simulation of epidemic processes on dynamic interaction networks. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 143–152. Springer, 2019.

[2] Anass Bouchinita and Assiam Jebrane. A hybrid multi-scale model of covid-19 transmission dynamics to assess the potential of non-pharmaceutical interventions. *Chaos, Solitons & Fractals*, page 109941, 2020.

[3] Giuseppe C Calafiore, Carlo Novara, and Corrado Possieri. A modified sir model for the covid-19 contagion in italy. *arXiv preprint arXiv:2003.14391*, 2020.

[4] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[5] L Devroye. *Non-uniform random variate generation*. Springer-Verlag, 1986.

[6] Wei Duan, Zongchen Fan, Peng Zhang, Gang Guo, and Xiaogang Qiu. Mathematical and computational approaches to epidemic modeling: a comprehensive review. *Frontiers of Computer Science*, 9(5):806–826, 2015.

[7] Giulia Giordano, Franco Blanchini, Raffaele Bruno, Patrizio Colaneri, Alessandro Di Filippo, Angelo Di Matteo, Marta Colaneri, et al. A sidarthe model of covid-19 epidemic in italy. *arXiv preprint arXiv:2003.09861*, 2020.

[8] Andreas Koher, Hartmut HK Lentz, Philipp Hövel, and Igor M Sokolov. Infections on temporal networks - a matrix-based approach. *PLoS one*, 11(4):e0151209, 2016.

[9] Ronald Prescott Loui. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26(9):670–676, 1983.

[10] Dave Osthus, Kyle S Hickmann, Petruţa C Caragea, Dave Higdon, and Sara Y Del Valle. Forecasting seasonal influenza with a state-space sir model. *The annals of applied statistics*, 11(1):202, 2017.

[11] Lorenzo Pellis, Frank Ball, Shweta Bansal, Ken Eames, Thomas House, Valérie Isham, and Pieter Trapman. Eight challenges for network epidemic models. *Epidemics*, 10:58–62, 2015.

[12] Luis EC Rocha and Naoki Masuda. Individual-based approach to epidemic processes on arbitrary dynamic contact networks. *Scientific reports*, 6:31456, 2016.

[13] Eyob Teweldemedhin, Tshilidzi Marwala, and Conrad Mueller. Agent-based modelling: a case study in hiv epidemic. In *Fourth International Conference on Hybrid Intelligent Systems (HIS’04)*, pages 154–159. IEEE, 2004.

Fig. 7: Simulation results for Example 5-(i). The diameter of each circle is proportional to how many times \( k_i \) is found with that value in the Montecarlo simulations of dynamics (9). Few agents are found to be non-infected (in few simulations). The results provided by the Contagion Graph (black triangles) show that this approach provides a reliable estimation of the "mean behavior".

Fig. 8: Simulation results for Example 5-(ii). The decrease in infectivity results in many agents not being infected through the Montecarlo simulations (see the red circles with larger diameter than Example 5-(i)). This phenomenon is captured by the Contagion Graph.