Published in final edited form as:

Math Biosci. 2012 December ; 240(2): 77–84. doi:10.1016/j.mbs.2012.07.002.

Finding the probability of infection in an SIR network is NP-Hard

Michael Shapiro[1] and
Department of Pathology, Tufts University, Boston, MA 02111

Edgar Delgado-Eckert[2]
Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology Zurich (ETH Zürich), Basel, Switzerland and Swiss Institute of Bioinformatics. Current affiliation: University Children's Hospital (UKBB), University of Basel, Spitalstr. 33, Postfach 4031, Basel, Switzerland

Edgar Delgado-Eckert: edgar.delgado-eckert@unibas.ch

Abstract

It is the purpose of this article to review results that have long been known to communications network engineers and have direct application to epidemiology on networks. A common approach in epidemiology is to study the transmission of a disease in a population where each individual is initially susceptible (S), may become infective (I) and then removed or recovered (R) and plays no further epidemiological role. Much of the recent work gives explicit consideration to the network of social interactions or disease-transmitting contacts and attendant probability of transmission for each interacting pair. The state of such a network is an assignment of the values $\{S, I, R\}$ to its members. Given such a network, an initial state and a particular susceptible individual, we would like to compute their probability of becoming infected in the course of an epidemic. It turns out that this and related problems are **NP**-hard. In particular, it belongs in a class of problems for which no efficient algorithms for their solution are known. Moreover, finding an efficient algorithm for the solution of any problem in this class would entail a major breakthrough in theoretical computer science.

Keywords

epidemics; SIR networks; contact network; network reliability; NP-hard

1. Introduction

Mathematical modelling of epidemics is often traced to the celebrated SIR model of Kermack and McKendrick [1]. This model posits a population of constant size whose members fall into one of three classes: susceptible (S), infective (I) and removed (R). Approximating these as continuous and assuming well-mixing, i.e., each individual is in equal contact with and equally likely to infect each other individual, allows for an approximate description of the infection dynamics using ordinary differential equations (ODE).

[1] The first author wishes to acknowledge support from NIH Grant K25 AI079404-02.

[2] The second author is corresponding author.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Clearly, as it has been argued by many in theoretical [2, 3, 4] as well as experimental studies [5], the well-mixing assumption is not an accurate representation of real contact patterns. Thus, much recent work has focused on the role of the network of disease-transmitting contacts. (Reviewed in [6]. See also, [7, 8]. For a comparison of well-mixed and network-based models, see [9].) Indeed, Kermack's and McKendrick's ODE model arises as the limiting case as the number of people goes to infinity of a simplistic network model in which each individual has an equal chance of infecting every other. However, real-world social contact networks exhibit complex patterns of interconnection between individuals. Further, the probability of transmitting disease from one individual to another depends on the nature, frequency and duration of the contact as well as the immune competence of the target individual. This leads to a modelling formalism of social networks as a probabilistic graph $\mathcal{G} = (G, Pr)$. Here G is the graph $G = (V, E)$, each vertex $u \in V$ is an individual, each edge $e = (u, v) \in E$ records the fact that u might infect v and $Pr : E \to [0, 1]$ gives the probability that u infects v if u becomes infective while v is susceptible. In this formalism, $\mathcal{G}$ is a fixed graph $G = (V, E)$ with labelling Pr.

This relatively new modelling paradigm has triggered an enormous amount of research in theoretical epidemiology. The field has greatly benefited from approaches that range from applications of bond percolation theory and other techniques from statistical physics [10, 11, 3, 7, 12, 13, 14, 15, 16, 17, 18, 19] to large scale simulation endeavours [4, 20, 21, 22, 23]. Given that this mathematical formalism seems accurate and powerful to describe the spread of infectious diseases, the natural question arises as to whether calculations performed within this formalism can be used in practical situations to make useful predictions. Such calculations are based on potentially measurable parameters such as network topology and transmission probabilities [24]. For instance, one could attempt to calculate the probability that, given a social contact network $\mathcal{G}$, an epidemic starting with a set P of infectives results in the infection of an initially susceptible individual u. Are there any computational limitations when trying to calculate such magnitudes? If yes, how limiting are they? Fortunately, to address the computational issues associated with this and similar calculations, we don't need to start from scratch, given that network engineers have already studied since the 1970s problems that are essentially the same.

In the era of electronically digitalized information and digital computers, communications networks have become the biggest and count among the most important networks. The size of these networks is exponentially increasing. For instance, the size of the Internet shows exponential growth since its creation in the early nineties (http://www.isc.org/). As the components of such networks are subject to failure, engineers face the problem of designing, constructing and operating networks that meet the required standards of reliability. Of particular interest is the estimation of how reliable a given network is in performing its function, provided some knowledge about the reliability of its components is available. In many cases, the functionality of the network can be expressed as the ability of its topology to support the network's operation. In other words, the network is functional if and only if certain connectivity properties are fulfilled. Consider a network of computers which use this network to transmit messages. Let us suppose that each of these computers is reliable, but that each communication link has some chance of failure when called upon to transmit a message. We then encounter the same formalism explained above for social networks. A communications network is given by $\mathcal{G} = (G, Pr)$. Here $G = (V, E)$ where each vertex $u \in V$ is a computer, each edge $e = (u, v) \in E$ is a communication link and $Pr : E \to [0, 1]$ is the reliability of the communication link from u to v. One might ask, given a communications network $\mathcal{G}$, a set of computers P and a computer $u \notin P$, if the computers in P all send a message, what is the chance it will reach u? We will see that this is the same problem we stated above in the context of epidemics on social contact networks.

It has long been known in the communications network literature that this problem is computationally intractable. A standard benchmark of computational complexity is the class of **NP**-complete problems. This class has the following properties:

- At present, no algorithm for an **NP**-complete problem is known to have a running time which is bounded by a polynomial. Indeed, many algorithms for **NP**-complete problems have exponential running time. It is unknown whether any **NP**-complete problem can be solved by an algorithm with polynomial running time.
- If any problem in this class can be solved by an algorithm whose running time is bounded by a polynomial, then every problem in this class can be solved by an algorithm whose running time is bounded by a polynomial.

In view of the second, it is considered unlikely that any **NP**-complete problem has a polynomial time solution. The communication among computers problem (and hence the epidemiology problem) listed above is known to be as hard as any **NP**-complete problem. Such problems are termed **NP**-hard. This is not the first problem in network epidemiology known to be **NP**-hard. Previously known examples include the following: Given a social contact network and limited resources

- What is the optimal strategy for vaccinating a limited number of individuals?
- What is the optimal strategy for quarantining a limited number of individuals?
- What is the optimal strategy for placement of a limited number of sensors for monitoring the course of an epidemic?

(See [25, 26, 27, 28].) These problems involve the search for an optimum among subsets of the vertices or edges of the given social contact network. It might be hoped that finding the probability of infection of a single individual would be computationally less demanding. As the engineers have taught us, this is not so. While this result has been recently reported in the physics and operations research community [29] and has recently been applied to the spread of influence on social networks [30], it seems almost unknown among epidemiologists.

This article is organized as follows: In Section 2 we give a very brief overview of the relevant concepts and methods in computational complexity. This provides the unacquainted reader with the basic tools for understanding the main message of this paper. Section 3 provides the elementary formal mathematical framework for studying SIR epidemics on networks, including the connection with percolation theory. In Section 4 we present a series of problems that have been studied in network engineering and demonstrate their structural isomorphism with certain problems concerning *SIR* epidemics on networks. Section 5 is devoted to studying the computational complexity of extended/generalized epidemiological problems. We finish in Section 6 with some concluding remarks.

2. Computational complexity

In this section we give a brief introduction to some of the major classes in the study of computational complexity. Perhaps chief among these is the class **NP**-complete which is used as a common benchmark for describing problems which are algorithmically soluble but computationally intractable. We start with a quick overview of the classes diagrammed in Figure 1[3] and then proceed to fill in some of the mathematical details. For those wishing a fuller account we recommend [31].

[3] We wish to thank Prof. Benjamin Hescott for advising us on the status of the question as to whether **#P**-complete exhausts the intersection of **NP**-hard and **#P**.

The classes **P**, **NP** and **#P** are defined by the running times of the algorithms that solve them and the capacities of machines these algorithms run on. The classes **NP**-complete and **#P**-complete play analogous roles in the classes **NP** and **#P**. They consist of the “hardest” problems of the class in the following sense: a solution for any **NP**-complete problem can be quickly transformed into a solution to any problem in **NP**. Accordingly, if there is an efficient solution to any **NP**-complete problem, there is an efficient solution to every problem in **NP**. (**#P**-complete problems play a similar role in **#P**.) It will be the work of this section to give more precise accounts of the terms problem, running time, transform and the classes of machines involved.

In describing the class **NP**-complete, it is useful to describe the class **P**, and necessary to describe the class **NP**.

The computational complexity of a problem Π is measured in terms of the running time necessary for an algorithm which solves Π. Defining these terms requires some preliminaries. First, note that a problem Π consists of a collection of instances, D_Π. Thus, “Determine whether 18 is composite” is an instance of the problem, “For any integer n, determine whether n is composite.” This is an example of a decision problem, that is, for each instance, the answer is either “yes” or “no”. A decision problem Π can be formalized as the pair (D_Π, Y_Π), where $Y_\Pi \subset D_\Pi$ consists of the yes instances. In this example, D_Π is the set of integers and Y_Π is the set of composite integers. We will refer to this problem as $\Pi_{composite}$.

Notice that each instance $\pi \in \Pi$ has a size, $\ell(\pi)$ and that the computational cost of solving the problem grows with the size of the problem. In this example, the size $\ell(n)$ of the instance n is the number of digits in n. If we then have an algorithm M which solves Π, we can consider the running time $r_M(\pi)$ required by M when applied to the instance π. This could be measured in elapsed time or in terms of the number of steps carried out by M in this computation. We would like to know the worst (i.e., longest) running time for all instances of a given size. Now depending on the problem, there is no guarantee that there is at least one instance of every size. When no instance of size n exists we will take the worst running time for instances of that size to be zero. We refer to the function which gives the worst running time at each size to be the running time of M defined as follows:

$$r_M(n)=\begin{cases} 0 & if\{\pi|\ell(\pi)=n\}=\varnothing \\ \max\{r_M(\pi)|\ell(\pi)=n\} & otherwise \end{cases}$$

The class **P** consists of those decision problems which can be solved with a polynomial running time. Stated formally, a decision problem Π belongs to the class **P** if there is an algorithm M which solves Π and a polynomial $p(n)$ such that $r_M(n) \le p(n)$. An example of a problem in the class **P** is Π_{mult} defined as follows: An instance of Π_{mult} is three integers, a, b and c. The size of an instance is the total number of digits in a, b and c. These constitute a yes instance if $a \times b = c$. It is not hard to convince oneself that the amount of computation necessary to decide whether $a \times b = c$ is bounded by a polynomial in the total number of digits.

The class **NP** consists of non-deterministic polynomial time problems. That is, a decision problem is **NP** if a machine which is allowed to guess can verify a yes instance in polynomial time. $\Pi_{composite}$ provides an example of a problem which is **NP**. Given an instance of $\Pi_{composite}$, i.e., an integer c, if c is, in fact, composite, a correct guess as to its factors a and b, can be verified in polynomial time by calling Π_{mult}. One can define this class in terms of the operation of non-deterministic Turing machines. See, for example, [32].

Clearly **P** ⊆ **NP**. In view of the perceived complexity of many problems in **NP**, it is generally believed that **P** ≠ **NP**, although no rigorous proof is known.

The class **NP**-complete consists of the hardest problems in **NP**. The problems in **NP**-complete have the following property: Suppose that Π_1 is **NP**-complete. Suppose that Π_2 is **NP**. Then there is an algorithm M which translates any instance π_2 of Π_2 into an instance π_1 of Π_1 such that π_1 is a yes instance of Π_1 if and only if π_2 is a yes instance of Π_2. Further, both the computational cost of translating π_2 into π_1 and the size $\ell(\pi_1)$ are bounded by a polynomial in $\ell(\pi_2)$. It follows that if any **NP**-complete problem can be solved (deterministically) in polynomial time, then every **NP** problem can be solved in polynomial time. Put another way, if any **NP**-complete problem can be solved in polynomial time, we will then have **P** = **NP**.

Hundreds of problems are known to be **NP**-complete [31]. These come from fields such as graph theory, number theory, scheduling, code optimization and many others. They are widely believed to be intrinsically intractable, but this remains an open question. Other problems which are not necessarily **NP**-complete (e.g., because they are not decision problems) are known to be at least as hard. This is because for such a problem, say Γ, there is an **NP**-complete problem Π that can be reduced to Γ, where the computational cost of this reduction is bounded by a polynomial in the length of the instance problem considered. Thus, Γ can be used to solve Π. These problems are called **NP**-hard. Since **NP**-complete problems transform to each other, all **NP**-complete problems can be solved by a reduction to any **NP**-hard problem. **NP**-hard problems are found in fields as diverse as epidemiology and origami [33].

Finally, we turn to the classes **#P** and **#P**-complete. The class **NP** consists of decision problems, e.g., $\Pi_{composite}$ which asks whether a given integer c can be written as the product of two factors, $a \times b = c$. The class **#P** consists of counting problems, e.g., given an integer c, how many different ways are there to write it as the product $a \times b = c$? Let us call this $\Pi^{\#}_{composite}$. Thus, for the instance 18 of $\Pi_{composite}$ the answer is “yes”, while for the instance 18 of $\Pi^{\#}_{composite}$, the answer is 4, as witnessed by the products 2 × 9 = 18, 3 × 6 = 18, 6 × 3 = 18 and 9 × 2 = 18.

We have seen that a problem Π is **NP** if there is a non-deterministic algorithm M which verifies the yes-instances of Π in polynomial time. M is allowed to make guesses. For each instance π of Π, we can ask how many different guesses M can make which will lead to the verification of π as a yes-instance. Of course, if π is not a yes-instance, this number is zero. Let us refer to this counting version of Π as $\Pi^{\#}$. The problem $\Pi^{\#}$ belongs to the class **#P** if for every yes-instance π, every computation verifying π has length bounded by a polynomial in $\ell(\pi)$.

Clearly, any counting problem can be turned into a decision problem by asking whether the count is non-zero. In this way, every **#P** problem can be turned into a problem in **NP**. However, **#P** problems may well be strictly harder than **NP** problems in the following sense. Even if it turns out that **P**=**NP**, there may still be problems in **#P** that can’t be solved in polynomial time.

Among the problems in **#P**, there are some which qualify as being the hardest in the sense that a solution for any one of these can be quickly transformed into a solution for any problem in **#P**. These problems are referred to as **#P**-complete. Among the **#P**-complete problems are some which are the counting versions of **NP**-complete problems. It follows from this that every **#P**-complete problem is **NP**-hard. To see this, let $\Pi^{\#}_1$ be any **#P**-

complete problem, and let $\Pi_2^\#$ be a **#P**-complete problem whose corresponding decision problem Π_2 is **NP**-complete. A solution to $\Pi_1^\#$ can be transformed into a solution to $\Pi_2^\#$, and this, in turn answers Π_2, hence any problem in **NP**. It follows that $\Pi_1^\#$ is **NP**-hard. As we shall see below, some problems in network epidemiology are **#P**-complete and thus **NP**-hard.

3. SIR epidemics on networks

We start by describing a network SIR model in which both the population (which is finite) and the transmission probabilities between individuals are constant with respect to time. A network in question is a graph $G = (V, E)$ where the vertices V represent individuals and the presence of an edge $(u, v) \in E$ indicates that u and v have contact which would enable disease transmission from u to v. This transmission is probabilistic. Each edge (u, v) is labeled with the probability that u transmits the disease to v if u becomes infected while v is susceptible. We make no assumption as to the topology of this network. In this model, disease transmission takes place at discrete times $t = 1, 2, 3 \ldots$.

A state of this system is the assignment of each individual to one of the classes S (susceptible), I (infected) or R (recovered). (Figure 2 shows the state of a network at time $t = 3$.) Only susceptibles can become infective. Infected individuals remain so for one time step and then become recovered. These latter are assumed immune to further infection and play no further role in the spread of disease. The edges determine who can infect whom and consequently which states can follow a given state. The transmission probabilities on the edges determine the probability that any one of these states follows the given state. An epidemic is a sequence of states each of which is a possible successor of the previous state. We assume that the initial state consists of susceptibles and at least one infected, but make no assumption as to the number of people who become infected in the course of the epidemic. Given an initial state, we can speak of the probability that an epidemic evolves through a given sequence of states and the probability that it arrives at a particular state. Let us formalize this.

As above, a social contact network is a pair $\mathcal{G} = (G, Pr)$ where $G = (V, E)$ is the graph with vertex set V and edge set E. Each edge has the form (u, v) with $u, v \in V$ and $u \neq v$. We make no assumption on the topology of this graph. The function Pr assigns a probability to each edge, that is $Pr : E \to [0, 1]$. The states of $\mathcal{G}$ are given by[4]

$$\mathrm{St}(\mathcal{G})=\{\phi|\phi:V\to\{S,I,R\}\}.$$

Given states ϕ_1 and ϕ_2, the state ϕ_2 is a possible (immediate) successor of ϕ_1 if it satisfies the following conditions for each $u, v \in V$:

1. If $\phi_1(u) = R$, then $\phi_2(u) = R$. (Recovered individuals stay recovered.)
2. If $\phi_1(u) = I$, then $\phi_2(u) = R$. (Infected individuals recover in one step.)
3. If $\phi_1(u) = S$, then $\phi_2(u) \in \{S, I\}$. (Susceptible individuals either stay susceptible or become infected.)
4. If $\phi_2(u) = I$, then $\phi_1(u) = S$ and there is a vertex $q \in V\backslash\{u\}$ and an edge (q, u) with $\phi_1(q) = I$. (Infected individuals were susceptible and were infected by a neighbour.)

[4] In particular, a state ϕ can be seen as a subset of the Cartesian product $V \times \{S, I, R\}$, and therefore, it is meaningful to speak of the probability of a state or of a collection of states.

The requirement that individuals recover in exactly one time-step might appear to be a drastic oversimplification. However, the formalism is rich enough to accommodate patterns of latency and extended periods of infectivity. This can be done by replacing the individual represented by vertex u by a sequence of vertices u_1, u_2, … representing u on day 1, u on day 2, etc. See, e.g., [34].

An epidemic Φ is a sequence of states $\phi_1, \ldots, \phi_n$ where ϕ_{i+1} is a possible successor of ϕ_i for $i = 1, \ldots, n - 1$. We will assume that the initial state ϕ_1 consists of infectives and susceptibles. The length of this epidemic is $\ell(\Phi) = n$. Since individuals recover after one step and recovered individuals cannot be reinfected, infection must be transmitted or die out. As a consequence, no epidemic can be longer than the longest self-avoiding path in $G = (V, E)$, for otherwise, it must infect some vertex twice, a contradiction. If we assume that each edge transmits or fails to transmit independently, then it is not hard to compute the probability that a susceptible individual is infected by its infected neighbours. This, in turn, allows one to compute the probability that a state ϕ_1 is followed by a particular successor state ϕ_2. Let us denote this probability by $Pr(\phi_2 \mid \phi_1)$. This system enjoys the Markov property, that is, the probability of a given state depends only on the previous state. Thus given an initial state Φ_1, the probability of the epidemic $\Phi = \phi_1, \ldots, \phi_n$, is

$$Pr(\Phi|\phi_1)=\prod_{i=2}^{n}Pr(\phi_i|\phi_{i-1}).$$

The probability that u becomes infected at the n^{th} step in the course of an epidemic starting with ϕ_1 is

$$Pr(\phi_n(u)=I|\phi_1)=\sum_{\substack{\{\phi_1,\ldots,\phi_n|\\ \phi_n(u)=I\}}}Pr(\phi_1,\ldots,\phi_n|\phi_1).$$

Abusing notation, we denote the probability that u becomes infected in the course of some epidemic starting with ϕ_1 by

$$Pr(u|\phi_1)=\sum_{j=1}^{n}Pr(\phi_j(u)=I|\phi_1).$$

In this sum we consider each of these epidemics only up to the time at which u becomes infected. Because of this, no epidemic appearing in this sum is an initial sub-epidemic of another. Accordingly, these are disjoint cases and we can take the total probability to be the sum of the individual probabilities.

We will be interested in initial states ϕ_1 consisting only of infectives and susceptibles. In this case, we can identify ϕ_1 with the set of infectives $P=\phi_1^{-1}(I)$. This gives the notation $Pr(u \mid P)$.

Let us formalize the problem $\Pi_{epidemic}$ of finding $Pr(u \mid P)$. An instance π of this problem consists of

- A graph $G = (V, E)$.

- A labelling[5] $Pr : E \to [0, 1] \cap \mathbb{Q}$.
- An initial infective set $P \subset V$.
- An individual $u \in V \setminus P$.

A solution to π is the value $Pr(u \mid P)$.

We take $\ell(\Pi) = |V|$.

The epidemiological viewpoint we have just described follows the evolution of probabilities over time. If we ignore the order of events, we come to the simpler viewpoint of percolation. Percolation methods have been used in epidemiology. (See, for example, [36, 37, 34, 3, 12, 38]. The latter two contain extensive references.) Since an individual is only infected for one time step in the course of any epidemic, an edge can transmit at most once in the course of an epidemic. This allows us to consider a random variable that takes as values subgraphs of G. Given $\mathcal{G}$, we take $\mathbb{G}$ to be the random variable which takes values in $\{G' = (V, E') \mid E' \subseteq E\}$. The probability that $\mathbb{G}$ takes the value G' is given by

$$Pr(G') = \left(\prod_{e \in E'} Pr(e)\right)\left(\prod_{e \notin E'} (1 - Pr(e))\right). \quad (*)$$

We may think of E' as determining whether $e = (u, v)$ transmits in the course of an epidemic if that epidemic has a state ϕ with $\phi(u) = I$ and $\phi(q) = S$. Given a path τ in G, we will abuse notation by writing $\tau \subset G$ and $e \in \tau$ for the edges of τ. Given a path τ, the probability that it appears in $G' = (V, E')$ is

$$Pr\{G' | \tau \subset G'\} = \prod_{e \in \tau} Pr(e).$$

The following theorem is equivalent to observation (3) of [34, Lemma 3.5, pp 720–21]. The reference [34] uses $\zeta \in \{0, 1\}^E$ to induce a subgraph $G' = (V, \zeta^{-1}(1))$ of $G = (V, E)$.

Theorem 1

Suppose $\mathcal{G}$ is social contact network. Then

$$Pr(u|P) = Pr\{G' | G' \text{ contains a path from } P \text{ to } u\}.$$

Corollary 1

$Pr(u \mid P)$ is a finite sum of terms of the form (*). In particular, it is a polynomial in the values $Pr(e)$ with integer coefficients and degree at most $|E|$.

This Theorem and its Corollary provide the link between epidemiology and communications networks. We will see that in the case where Pr is taken to have the single value p the

[5] There are technical issues here concerning the values of these probabilities. To avoid these issues they are usually assumed to be rational numbers and bounds are placed on the sizes of their denominators. For details, see [35]. Since $\mathbb{Q}$ is dense in $\mathbb{R}$, this is not a limitation on the possible probability values relevant in real applications.

polynomial of Corollary 1 corresponds to the reliability polynomial as described below (see also [35]).

4. NP-hard problems on communications networks: Consequences for epidemiological calculations

We assume that a communications network consists of a set of computers, each of which is reliable and a set of communication links each of which has a known likelihood of failure and that the communication links function or fail independently. There is no loss of generality in regarding each node as infallible, since a fallible computer can be modelled as a pair of nodes with a fallible link connecting its input to its output. Once again, we can formalize this as $\mathcal{G} = (G, Pr)$, where $G = (V, E)$ represents installed capacity (V being the set of computers and E the set of communication links), $Pr : E \to [0, 1]$ the reliability of each link and $\mathbb{G}$ is the random variable assuming values in $\{G' = (V, E') \mid E' \subseteq E\}$. Each $G' = (V, E')$ is the subnetwork of functioning links left after the failure of the edges $e \in E \backslash E'$. Successful transmission of a message on this network depends on the connectivity of the subgraph realized by $\mathbb{G}$. Network engineers focus on several kinds of connectivity. We first examine two of the simplest.

The two-terminal reliability problem is defined as the calculation of the probability that there is at least one correctly functioning path in the network connecting a predefined source node to a predefined target node. An instance π of $\Pi_{two\ terminal}$ consists of the following:

- A graph $G = (V, E)$.
- A labelling $Pr : E \to [0, 1] \cap \mathbb{Q}$.
- A source terminal $u \in V$.
- A target terminal $v \in V \backslash \{u\}$.

A solution to π is the value $Pr(v \mid u)$.

By Corollary 1 this value is given by an integer polynomial in the values $Pr(e)$ for $e \in E$. If we restrict to the case where every edge takes the same value, this becomes an integer polynomial in one variable called the reliability polynomial. Thus a related problem is the following:

An instance π of $\Pi_{rel\ poly}$ is

- A graph $G = (V, E)$.
- A source terminal $u \in V$.
- A target terminal $v \in V \backslash \{u\}$.

A solution to π is the coefficients of the reliability polynomial.

A number of additional network reliability problems have been studied (see [35], an excellent introduction to this field). These include

- k terminal reliability. This requires that k chosen terminals are mutually pair wise connected.
- Broadcasting, also known as all terminal reliability: This requires that all terminals are pair wise connected.

Note that while we have defined the reliability polynomial using the two-terminal problem. By Corollary 1 these also give polynomials in a similar way.

Naturally, in addition to the network reliability problems presented above, many other reasonable problems can be defined or could arise from practical applications. Formally, once a model $\mathcal{G} = (G, Pr)$ of the network has been chosen, a general mechanism to define a reliability problem is the following: A network operation is specified by defining a set $Op(G) \subseteq \{G' = (V, E') \mid E' \subseteq E\}$ of states considered to be functional. The set $Op(G)$ is sometimes called a stochastic binary system; the elements of $Op(G)$ are termed pathsets. Specifying the pathsets for G determines the whole stochastic binary system, and therefore defines the network operation. The reliability problem consists of finding the probability $Pr(Op(G))$ that the probabilistic graph $\mathbb{G}$ assumes values in the set $Op(G)$.

A first naive algorithm to solve a network reliability problem formulated in this general manner is to enumerate all states of $\mathbb{G}$ (i.e., the cardinality of the set $\{G' = (V, E') \mid E' \subseteq E\}$), determine whether a given state is a pathset or not using some predesigned recognition procedure[6], and sum the occurrence probabilities of each pathset. Due to the statistical independence assumed, the probability of occurrence of a pathset is simply the product of the operation probabilities of the edges in the pathset and the failure probabilities of the edges not present in the pathset. Complete state enumeration requires the generation of all $2^{|E|}$ states of $\mathbb{G}$, implying that the running time of this algorithm would exponentially depend on the number of links in the network.

A substantial amount of effort has been put into finding more efficient algorithms for exact calculation of network reliability problems (see [35]). However, efficient exact solutions seem unlikely as witnessed by [39, Theorem 1, p 158]:

Theorem 2

The problems $\Pi_{two\ terminal}$ and $\Pi_{rel\ poly}$ are **NP**-hard.

These problems belong to the class **#P**-complete [40, 41, 42, 43, 44, 35] and are thus **NP**-hard (see Section 2 above).

Corollary 2

The problem $\Pi_{epidemic}$ is **NP**-hard.

To see this, notice that every instance of $\Pi_{two\ terminal}$ is an instance of $\Pi_{epidemic}$, namely, an instance in which P consists of a single vertex.

More generally, despite dedicated efforts, no algorithm of polynomial running time has been found that allows for the exact calculation of the probability $Pr(Op(G))$ of a given set of pathsets $Op(G)$, unless very specific assumptions are made on the topology of the underlying probabilistic network ([35, 45]). We consider it an open question as to which (if any) of these more general network reliability problems (defined through the choice of a suitable stochastic binary system $Op(G)$) correspond to epidemiological problems.

5. NP-hardness of extended problems in epidemiology

Epidemic on networks with time-varying transmission probabilities

As we have seen in the previous section, the seemingly simple problem of finding an individual's chances of infection is **NP**-hard. This is even so in the case where the set of initial infectives is a single individual.

[6] Such recognition procedures generally boil down to path-finding or spanning tree methods, which are efficient (i.e., of polynomial running time) and well-know procedures in algorithmic graph theory and computer science.

We can generalize $\Pi_{epidemic}$ by allowing transmission probabilities to vary over time. We have seen that the length of any epidemic is at most the length of the longest self-avoiding path in G. Consequently, time-varying transmission probabilities can be encoded as

$$Pr{:}E \times \{1, \ldots, |E|\} \rightarrow [0, 1].$$

In this case, percolation methods no longer apply. However, every instance of $\Pi_{epidemic}$ can be mapped into an instance of this extended problem. Thus, the time-varying version of this problem is **NP**-hard.

Epidemic on networks with disease latency

One might also generalize $\Pi_{epidemic}$ to allow patterns of latency and extended periods of infectivity[7]. We will take $\mathcal{I}$ to be a sequence of distinct states, $\{I_1, I_2, \ldots, I_N\}$. We assume that for each stage I_i there is an infectivity μ_i and a probability of recovery ρ_i. We take $\rho_N = 1$. We now consider a social contact network $\mathcal{G}$ and infectivity pattern $\mathcal{I}$. We refer to this as an $S\,\mathcal{I}\,R$ network. The states of this network are

$$\{\phi | \phi{:}V \rightarrow \{S\} \cup \mathscr{I} \cup \{R\}\}.$$

We modify the definition of possible successor states so that the allowable transitions are from S to I_1, from I_i to I_{i+1} for $i = 1$, & mldr;, $N - 1$ and from I_i to R for $i = 1, \ldots, N$. If $\phi(u) = I_i$, u transitions to state R with probability ρ_i and to state I_{i+1} with probability $1 - \rho_i$. If $e = (u, v) \in E$ and $\phi(u) = I_i$, and $\phi(v) = S$, then u infects v with probability $Pr_{\mathcal{I}}(e, i) = \mu_i Pr(e)$. We assume that $\mathcal{I}$ is non-trivial in the sense that there is i with $\mu_i \neq 0$ and $\rho_j \neq 1$. This ensures that at least one state is capable of transmitting infection and that an infected individual has a positive probability of reaching such an infective state. As before, under the assumption that transmissions and recoveries happen independently, we can develop an expression for $Pr_{\mathcal{I}}(u \mid P)$.

Fix $\mathcal{I}$. An instance of $\Pi_{\mathcal{I}}$ is an instance of $\Pi_{\ epidemic}$.

A solution to $\Pi_{\mathcal{I}}$ is the value $Pr_{\mathcal{I}}(u \mid P)$

Theorem 3—Given an non-trivial infectivity pattern $\mathcal{I}$, $\Pi_{\mathcal{I}}$ is **NP**-hard.

Lemma 1—Given $\mathcal{G} = (G, Pr)$ and $\mathcal{I} = \{I_1, \ldots, I_N\}$, there is $\mathcal{G}' = (G, Pr')$ so that for each $P \subset V$ and $u \notin P$, $Pr_{\mathcal{I}}(u \mid P) = Pr'(u \mid P)$.

Proof: Consider an edge $e = (u, v)$. Suppose that $\phi_1(u) = I_i$ and $\phi_1(v) = S$. What are the chances that v remains uninfected by u? (We assume for the moment that v is not infected by some other neighbour during the next N steps.) We take $\mu = Pr(e)$. Let us denote by ν_i the probability that u remains infected for i steps, but not $i + 1$ steps. We then have

[7] For a more general version of this see [34].

$$\nu_i = \rho_i \prod_{j=1}^{i} (1 - \rho_j).$$

The probability that v remains uninfected by u is

$$\tau_{\mathscr{I}}(\mu) = \sum_{i=1}^{N} \nu_j \prod_{j=1}^{i} (1 - \mu_j \mu)$$

We now define $\mathscr{G}' = (G, Pr')$ by taking

$$Pr'(e) = 1 - \tau_{\mathscr{I}}(Pr(e)).$$

This does what is required.

Proof (Proof of Theorem 3): We will show that $\Pi_{rel\ poly}$ is polynomially reducible to $\Pi_{\mathscr{I}}$.

Fix $\mathscr{I}$ to be a non-trivial pattern of infectivity. Suppose we are given an instance $\boldsymbol{\pi}$ of $\Pi_{\text{rel poly}}$. This consists of a graph $G = (V, E)$ and source and target vertices u and v. Suppose also that we have a polynomial time algorithm for solving $\Pi_{\mathscr{I}}$. Let M be the degree of the reliability polynomial we are trying to find. This is bounded by the length of the longest self-avoiding path in G. We choose $M + 1$ arbitrary probabilities $p_0, \ldots, p_{M+1}$. These give us $M + 1$ instances of $\Pi_{\mathscr{I}}$ by taking $\mathscr{G}_i = (G; Pr_i)$, where Pr_i takes the constant value p_i. By the previous lemma, solving these $M + 1$ instances of $\Pi_{\mathscr{I}}$ solves $M + 1$ distinct instances of $\Pi_{epidemic}$ which consist of the graph G and differing constant functions Pr_i'. These $M + 1$ values give us $M + 1$ independent linear equations whose unknowns are the coefficients of the reliability polynomial. Solving for these is a polynomial time problem.

Expected number of total infections

One might hope that while computing an individual's probability of infection is **NP**-hard, there might be a way to compute the expected number of infections. This, too, is **NP**-hard. Let us formalize this.

An instance $\boldsymbol{\pi}$ of $\Pi_{expected}$ is

- A graph $G = (V, E)$.
- A labelling $Pr : E \to [0, 1] \cap \mathbb{Q}$.
- An initial infective set $P \subset V$.

A solution to $\boldsymbol{\pi}$ is the expected number of infections,

$$\sum_{u \in V} Pr(u|P).$$

The following theorem was proved in [29, Theorem 1, p 483]. For the sake of completeness, we provide a proof here.

Theorem 4—Π_{expected} is **NP**-hard.

Proof: We will show that $\Pi_{epidemic}$ can be polynomially reduced to $\Pi_{expected}$. Suppose we are given an instance π of $\Pi_{epidemic}$. Let $\tilde{\pi}$ be the instance of $\Pi_{epidemic}$ which is formed from π by appending a single edge from u to $v \notin V$ and assigning $Pr(u, v) = 1$. It is clear that the expected number of infections in $\tilde{\pi}$ differs from the number of expected infections in $\tilde{\pi}$ by exactly $Pr(u \mid P)$. Thus, if we had a polynomial time algorithm for finding the expected number of infections, we could find the probability of any individual becoming infected.

The fact that $\Pi_{rel\ poly}$ is **NP**-hard suggests that the difficulty lies not in the probabilities Pr but in the topology of G. One problem which we have not addressed here is the question of calculating the probability of infection in an SIR network where the graph $G_t = (V_t, E_t)$ changes over time due to stochastic births and deaths. It seems likely that this will also provide a source of **NP**-hard problems. However, this requires a reformulation of the underlying problem.

6. Discussion and conclusions

It has been the purpose of this paper to draw the attention of network epidemiologists to results in communications network reliability which shed light on questions regarding the computational aspects of epidemiology of *SIR* networks.

Theorem 2 and Theorem 4 tell us that generally, in the absence of a major break-through in computer science we cannot expect to be able to compute exact probabilities of infection or expected number of infection in large social contact networks. As [31] points out, problems do not go away simply because we have deemed them **NP**-hard.

Since the network engineers have been here before us, it is tempting to ask whether their solutions will work for epidemiologists. While we consider the case open, the prospects seem mixed. Network engineers are often in the position of being able to choose the class of networks under consideration. As opposed to scale-free [8, 46] and small-world network structures [47, 48, 8], which frequently arise from a self-organization process during the spontaneous growth of a network, engineered or purposefully designed networks show rather different structures. Some of the classes that allow efficient calculations (exact or approximate) include trees, full graphs, series-parallel graphs [35], and channel graphs [45]. (See Figure 3.) Unfortunately, these classes of networks seem unrealistic as models of social contact networks.

Network engineers have turned to Monte Carlo simulation for the calculation of estimates of network reliability. We would like to give pointers into their literature [49, 50, 51, 52, 53, 54, 55, 29]. This approach has received increased attention in the last decade due to the power of modern computers and computing clusters. While Monte Carlo simulation only calculates an unbiased point estimator for reliability probabilities, increasing the number of simulated samples causes these estimates to converge to the actual value.

The fact that efficient and precise algorithms for computing infection probabilities are out of reach (see Theorems 2, 3 and 4) has real-world consequences. Designing a response to an emerging epidemic can depend on determining the kind of epidemiological probabilities we have been discussing [56]. The effectiveness of interventions during an emerging epidemic often crucially depends on timely implementation. Our results and those of [25] and [26] place an emphasis on the search for efficient and quick methods that give good approximations when applied to real-world social networks.

References

1. Kermack W, McKendrick A. A contribution to the mathematical theory of epidemics. Proc Roy Soc Lond A. 1927; 115:700–721.
2. Bansal, Shweta; Grenfell, BT.; Meyers; Ancel, L. When individual behaviour matters: homogeneous and network models in epidemiology. Journal of The Royal Society Interface. 2007; 4(16):879–891. arXiv: http://rsif.royalsocietypublishing.org/content/4/16/879.full.pdf+html. URL http://rsif.royalsocietypublishing.org/content/4/16/879.abstract. 10.1098/rsif.2007.1100
3. Newman MEJ. Spread of epidemic disease on networks. Phys Rev E. 2002; 66(1):016128.10.1103/PhysRevE.66.016128
4. Eubank S, Guclu H, Kumar VSA, Marathe MV, Srinivasan A, Toroczkai Z, Wang N. Modelling disease outbreaks in realistic urban social networks. Nature. 2004; 429(6988):180–184. URL http://www.ncbi.nlm.nih.gov/pubmed/15141212. [PubMed: 15141212]
5. Cauchemez S, Bhattarai A, Marchbanks TL, Fagan RP, Ostroff S, Ferguson NM, Swerdlow D. the Pennsylvania H1N1 working group, Role of social networks in shaping disease transmission during a community outbreak of 2009 h1n1 pandemic influenza. Proceedings of the National Academy of Sciences. 2011; 108(7):2825–2830. arXiv: http://www.pnas.org/content/108/7/2825.full.pdf+html. URL http://www.pnas.org/content/108/7/2825.abstract. 10.1073/pnas.1008895108
6. Keeling MJ, Eames KT. Networks and epidemic models. Journal of The Royal Society Interface. 2005; 2(4):295–307. arXiv:http://rsif.royalsocietypublishing.org/content/2/4/295.full.pdf+html. URL http://rsif.royalsocietypublishing.org/content/2/4/295.abstract. 10.1098/rsif.2005.0051
7. Newman MEJ. The structure and function of complex networks. SIAM Review. 2003; 45:167–256.
8. Eubank S. Network based models of infectious disease spread. Jpn J Infect Dis. 2005; 58(6):S9–13. [PubMed: 16377861]
9. Keeling M. The implications of network structure for epidemic dynamics. Theoretical Population Biology. 2005; 67(1):1–8. URL http://www.sciencedirect.com/science/article/B6WXD-4F4NYDN-1/2/58290fd5aa19b724c74dda6a1aa296d2. 10.1016/j.tpb.2004.08.002 [PubMed: 15649519]
10. Allard A, Noël P-A, Dubé LJ, Pourbohloul B. Heterogeneous bond percolation on multitype networks with an application to epidemic dynamics. Phys Rev E. 2009; 79(3):036113.10.1103/PhysRevE.79.036113
11. Miller JC. Percolation and epidemics in random clustered networks. Phys Rev E. 2009; 80(2):020901.10.1103/PhysRevE.80.020901
12. Kenah E, Robbins JM. Second look at the spread of epidemics on networks. Phys Rev E. 2007; 76:036113. URL. 10.1103/PhysRevE.76.036113
13. Moreno Y, Pastor-Satorras R, Vespignani A. Epidemic outbreaks in complex heterogeneous networks. Eur Phys J B. 2002; 26(4):521–529. URL http://dx.doi.org/10.1140/epjb/e20020122. 10.1140/epjb/e20020122
14. Meyers LA, Newman M, Pourbohloul B. Predicting epidemics on directed contact networks. Journal of Theoretical Biology. 2006; 240(3):400–418. URL http://www.sciencedirect.com/science/article/pii/S0022519305004418. 10.1016/j.jtbi.2005.10.004 [PubMed: 16300796]
15. Volz E. Sir dynamics in random networks with heterogeneous connectivity. Journal of Mathematical Biology. 2008; 56:293–310. URL http://dx.doi.org/10.1007/s00285-007-0116-4. 10.1007/s00285-007-0116-4 [PubMed: 17668212]
16. Volz E, Meyers LA. Susceptible-infected-recovered epidemics in dynamic contact networks. Proceedings of the Royal Society B: Biological Sciences. 2007; 274(1628):2925–2934. arXiv:http://rspb.royalsocietypublishing.org/content/274/1628/2925.full.pdf+html. URL http://rspb.royalsocietypublishing.org/content/274/1628/2925.abstract. 10.1098/rspb.2007.1159
17. Pastor-Satorras R, Vespignani A. Epidemic spreading in scale-free networks. Phys Rev Lett. 2001; 86(14):3200–3203.10.1103/PhysRevLett.86.3200 [PubMed: 11290142]

18. House T, Keeling MJ. Insights from unifying modern approximations to infections on networks. Journal of The Royal Society Interface. 2011; 8(54):67–73. arXiv:http://rsif.royalsocietypublishing.org/content/8/54/67.full.pdf+html. URL http://rsif.royalsocietypublishing.org/content/8/54/67.abstract. 10.1098/rsif.2010.0179

19. Boguñá M, Pastor-Satorras R, Vespignani A. Absence of epidemic threshold in scale-free networks with degree correlations. Phys Rev Lett. 2003; 90(2):028701.10.1103/PhysRevLett.90.028701 [PubMed: 12570587]

20. Stroud P, Del Valle S, Sydoriak S, Riese J, Mniszewski S. Spatial dynamics of pandemic influenza in a massive artificial society. Journal of Artificial Societies and Social Simulation. 2007; 10(4):9. URL http://jasss.soc.surrey.ac.uk/10/4/9.html.

21. Mniszewski, SM.; Del Valle, SY.; Stroud, PD.; Riese, JM.; Sydoriak, SJ. Proceedings of the 2008 Spring simulation multiconference, SpringSim '08. Society for Computer Simulation International; San Diego, CA, USA: 2008. Episims simulation of a multi-component strategy for pandemic influenza; p. 556-563.URL http://portal.acm.org/citation.cfm?id=1400549.1400636

22. Valle, SD.; Kubicek, D.; Mniszewski, S.; Riese, J.; Romero, P.; Smith, J.; Stroud, P.; Sydoriak, S. Tech Rep LAUR-06-0666. Los Alamos National Laboratory; 2006. EpiSimS Los Angeles Case Study.

23. Mniszewski SM, Del Valle SY, Stroud PD, Riese JM, Sydoriak SJ. Pandemic simulation of antivirals + school closures: buying time until strain-specific vaccine is available. Comput Math Organ Theory. 2008; 14:209–221. URL http://portal.acm.org/citation.cfm?id=1394977.1394987. 10.1007/s10588-008-9027-1

24. Wallinga J, Teunis P, Kretzschmar M. Using data on social contacts to estimate age-specific transmission parameters for respiratory-spread infectious agents. American Journal of Epidemiology. 2006; 164(10):936–944. arXiv:http://aje.oxfordjournals.org/content/164/10/936.full.pdf+html. URL http://aje.oxfordjournals.org/content/164/10/936.abstract. 10.1093/aje/kwj317 [PubMed: 16968863]

25. Wang, N. Phd thesis. University of Maryland; College Park, MD, USA: 2005. Modeling and analysis of massive social networks. adviser: Srinivasan, Aravind

26. Eubank, S.; Kumar, VSA.; Marathe, MV.; Srinivasan, A.; Wang, N. Structural and algorithmic aspects of massive social networks. SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics; Philadelphia, PA, USA. 2004. p. 718-727.

27. Aspnes, J.; Chang, K.; Yampolskiy, A. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics; Philadelphia, PA, USA. 2005. p. 43-52.

28. Hayrapetyan, A.; Kempe, D.; Pál, M.; Svitkina, Z. Algorithms - ESA 2005. Springer-Verlag; 2005. Unbalanced graph cuts; p. 191-202.URL {http://dx.doi.org/10.1007/1156107119}

29. Laumanns M, Zenklusen R. Computational complexity of impact size estimation for spreading processes on networks. European Physical Journal B. 2009; 71(4):481–487. URL http://www.springerlink.com/index/10.1140/epjb/e2009-00344-7.

30. Wang, C.; Chen, W.; Wang, Y. Scalable influence maximization for independent cascade model in large-scale social networks; Data Mining and Knowledge Discovery. p. 1-32.URL http://dx.doi.org/10.1007/s10618-012-0262-1

31. Garey, MR.; Johnson, DS. A Guide to the Theory of NP-Completeness. W. H. Freeman & Co; New York, NY, USA: 1990. Computers and Intractability.

32. Hopcroft, JE.; Ullman, JD. Introduction To Automata Theory, Languages, And Computation. Addison-Wesley Longman Publishing Co., Inc; Boston, MA, USA: 1990.

33. Bern, M.; Hayes, B. The complexity of flat origami. Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Mathematics; 1996. p. 175-183.

34. Floyd W, Kay L, Shapiro M. Some elementary properties of sir networks or, can I get sick because you got vaccinated? Bull Math Biol. 2008; 70(3):713–727. URL http://www.ncbi.nlm.nih.gov/pubmed/18060461. 10.1007/s11538-007-9275-0 [PubMed: 18060461]

35. Colbourn, CJ. The Combinatorics of Network Reliability. Oxford University Press, Inc; New York, NY, USA: 1987.

36. Grassberger P. Critical behavior of the general epidemic process and dynamical percolation. Math Biosci. 1983; 63:205–213.

37. Sander LM, Warren CP, Sokolov IM, Simon C, Koopman J. Percolation on heterogeneous networks as a model for epidemics. Math Biosci. 2002; 180:293–205. [PubMed: 12387929]

38. Meyers LA. Contact network epidemiology: bond percolation applied to infectious disease prediction and control. Bull Amer Math Soc. 2007; 44:63–86.

39. Ball MO. Complexity of network reliability computations. Networks. 1980; 10:153–165.

40. Buzacott, JA. Tech Rep Working Paper No 76-0 16. Department of Industrial Engineering, University of Toronto; Aug. 1976 A recursive algorithm for finding the probability that a graph is disconnected.

41. Ball, MO. Phd thesis. Cornell University; 1977. Network reliability analysis: Algorithms and complexity.

42. Rosenthal A. Computing the reliability of complex networks. SIAM J Appl Math. 1977; 32:384–393.

43. Ball MO. Computing network reliability. Oper Res. 1979; 27:823–838.

44. Agrawal A, Barlow R. A survey of network reliability and domination theory. Oper Res. 1984; 32:478–492.

45. Harms, DD.; Kraetzl, M.; Colbourn, CJ.; Devitt, JS. Network Reliability: Experiments with a Symbolic Algebra Environment. CRC Press, Inc; Boca Raton, FL, USA: 1995.

46. May RM, Lloyd AL. Infection dynamics on scale-free networks. Phys Rev E. 2001; 64(6): 066112.10.1103/PhysRevE.64.066112

47. Watts DJ, Strogatz SH. Collective dynamics of ‘small-world’ networds. Nature. 1998; 393(4):440–443. [PubMed: 9623998]

48. Neuman MEJ, Watts DJ. Scaling and percolation in the small-world network model. Phys Rev E. 1999; 60:7332–7342.

49. Barlow, RE.; Proschan, F. Statistical theory of reliability and life testing. Holt, Rinehart and Winston, Inc; New York: 1975. probability models, International Series in Decision Processes, Series in Quantitative Methods for Decision Making

50. Gertsbakh, IB.; Shpungin, Y. Models of Network Reliability: Analysis, Combinatorics and Monte Carlo. CRC Press, Inc; Boca Raton, FL, USA: 2010.

51. Fishman GS. A Monte Carlo sampling plan for estimating network reliability. Oper Res. 1986; 34(4):581–594. URL http://dx.doi.org/10.1287/opre.34.4.581. 10.1287/opre.34.4.581

52. Lomonosov M, Shpungin Y. Combinatorics of reliability Monte Carlo. Random Structures Algorithms. 1999; 14(4):329–343. URL http://dx.doi.org/10.1002/(SICI)1098-2418(199907)14:4〈329::AID-RSA3〉3.0.CO;2-X. 10.1002/(SICI)1098-2418(199907)14:4<329::AID-RSA3>3.0.CO;2-X

53. Cancela H, El Khadiri M. A recursive variance-reduction algorithm for estimating communication-network reliability, Reliability. IEEE Transactions on. 1995; 44(4):595–602.10.1109/24.475978

54. Hui KP, Bean N, Kraetzl M, Kroese D. The tree cut and merge algorithm for estimation of network reliability. Probab Engrg Inform Sci. 2003; 17(1):23–45. URL http://dx.doi.org/10.1017/S0269964803171021. 10.1017/S0269964803171021

55. Hui KP, Bean N, Kraetzl M, Kroese DP. The cross-entropy method for network reliability estimation. Ann Oper Res. 2005; 134:101–118. URL http://dx.doi.org/10.1007/s10479-005-5726-x. 10.1007/s10479-005-5726-x

56. Wallinga J, van Boven M, Lipsitch M. Optimizing infectious disease interventions during an emerging epidemic. PNAS. 2010; 107(2):923–928. [PubMed: 20080777]

Highlight

Communicaitons engineers have long studied transmission of messages in fallible networks. In its general form, exact solution of this problem is NP-hard and therefore presumed intractable. The communication problem is isomorphic to transmission of disease in social contact networks. In particular, exact computation of infection probabilities in social contact networks is also NP-hard.

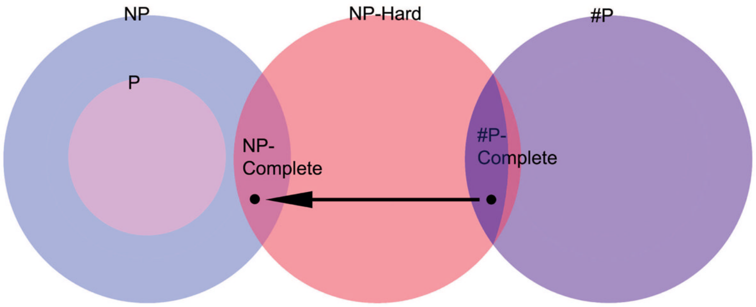

Figure 1.

The class **NP** consists of *decision problems*, problems for which the answer is “yes or “no”. Within this class is the class **P** of problems which can be solved with polynomial running time and the class **NP**-complete of problems which are as hard as any other problem in this class. It is not known whether **P** = **NP**. *This diagram is drawn under the assumption that these are not equal.* The class **#P** consists of *counting* problems, problems for which the answer is a count, a non-negative integer. Within this class is the class **#P**-complete of problems which are as hard as any in **#P**. Every counting problem can be turned into a decision problem, namely, “Is the count larger than zero?” In this way some **#P**-complete problems become **NP**-complete problems, and thus all **#P**-complete problems (and all **NP**-complete problems) belong to the class of **NP**-hard problems, those which are as hard as any **NP** problem. The sliver to the right of the **#P**-complete problems indicates the possibility that (**NP**-hard) ∩ (**#P**) $\subsetneq$ **#P**-complete. This is currently unknown.

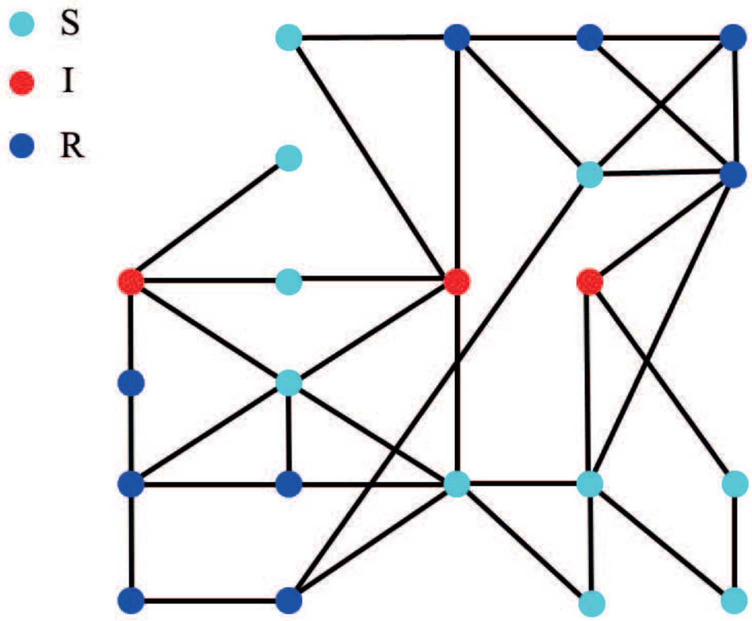

Figure 2.

The state of a social network on the third day of an epidemic which started at the lower left and upper right corners. Susceptibles are shown in cyan, infectives in red and recovereds in blue. Every infective is three edges from one of the two original infectives along some route of transmission. Each infective shares an edge with at least one recovered, namely the recovered that infected it. One susceptible will never become infected. The epidemic can continue for at most six more steps. We have omitted the probabilities on the edges.

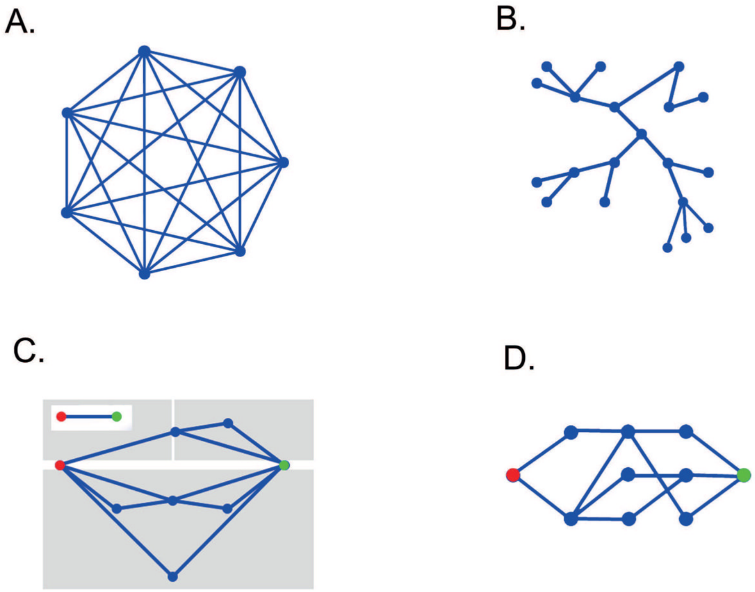

Figure 3.

A. The full graph on seven vertices. Every edge is connected to every other. The Kermack-McKendrick model arises as the limit of the full graph as the number of vertices goes to infinity. B. Trees are connected graphs without loops. C. Series parallel graphs have a distinguished source and target vertex. They are built up from the graph which consists of two vertices and one edge (inset) by repeatedly combining smaller series parallel graphs either in series or in parallel. The grey rectangles indicate the last two moves in assembling this graph. The reader may wish to check that the three subgraphs are series parallel graphs. D. Channel graphs also have a distinguished source and target vertex. The vertices are partitioned into subsets and this set of subsets is ordered. Here each subset is grouped vertically. The first and last of these subsets each consist of a single vertex. Each edge connects vertices from adjacent subsets. While each of these classes of graphs allows for efficient computation of epidemic probabilities, none of them seem likely to be useful models of real world social networks.