

THE CONTAGION NUMBER: HOW FAST CAN A DISEASE SPREAD?

Misty Blessley¹, Randy Davila², Trevor Hale^{3*} and Ryan Pepper⁴

¹ Statistics, Operations, and Data Science Department, Temple University, USA

² Computational and Applied Mathematics Department, Rice University, USA

³ Information and Operations Management Department, Texas A&M University, USA

⁴ Mathematics and Statistics Department, University of Houston - Downtown, USA

Received: 23 November 2022;

Accepted: 30 January 2023;

Available online: 1 March 2023.

Original scientific paper

Abstract: *The overall purpose of this paper is to define a new metric on the spreadability of a disease. Herein, we define a variant of the well-known graph-theoretic burning number (BN) metric that we coin the contagion number (CN). We aver that the CN is a better metric to model disease spread than the BN as the CN concentrates on first time infections. This is important because the Centers for Disease Control and Prevention report that COVID-19 reinfections are rare. This paper delineates a novel methodology to solve for the CN of any tree, in polynomial time, which addresses how fast a disease could spread (i.e., a worst-cast analysis). We then employ Monte Carlo simulation to determine the average contagion number (ACN) (i.e., a most-likely analysis) of how fast a disease would spread. The latter is analyzed on scale-free graphs, which are specifically designed to model human social networks (sociograms). We test our method on some randomly generated scale-free graphs and our findings indicate the CN to be a robust, tractable (the BN is NP-hard even for a tree), and effective disease spread metric for decision makers. The contributions herein advance disease spread understanding and reveal the importance of the underlying network structure. Understanding disease spreadability informs public policy and the associated managerial allocation decisions.*

Key words: *Disease spread, graph theory, burning number, contagion number, COVID-19.*

1. Introduction

In early 2020, Earth was hit with a pandemic the likes of which it had not seen in 100 years. According to the World Health Organization (WHO), a pandemic is the worldwide spread of a new disease. This disease we speak of is none other than COVID-19, which stands for coronavirus disease of 2019, and is the illness caused by the SARS-CoV2 virus originally recognized as an “outbreak of pneumonia of unknown

* Corresponding author.

E-mail addresses: misty.blessley@temple.edu (M. Blessley), trevor.hale@tamu.edu (T. Hale *), rrd6@rice.edu (R. Davila), pepperr@uhd.edu (R. Pepper)

The contagion number of a graph. How fast can a disease spread?

etiology” in Wuhan City, Hubei Province, China, on December 31, 2019. (Centers for Disease Control and Prevention, 2021a., p. 3). COVID-19 exposed our global operational fragility, as could be seen in disruptions of necessities such as food, personal products, and medical supplies. Supply chain and operations personnel were asked to pivot on a dime and public policymakers faced tough decisions, in the quest to maintain social welfare.

An understanding of disease spread can bring clarity for decision makers (Keskinocak, 2021). For example, at one point allocating ventilators was reduced to deciding who gets to live and who gets to die (Medical News Today, April 8, 2020). “Allocations that maximize health benefits are rarely equitable, while equitable allocations may be significantly non-optimal,”(Duijzer et al., 2018, p.143), thus capturing the serious nature of exactly how scarce supplies should be allocated. When considering that, “...the primary reliance on the innovation capabilities of the developed world during pandemics does not present an optimal way forward.”(Adbi et al., 2019, p. 826), as a means of increasing supply, it can clearly be seen how understanding disease spread can prove beneficial.

Herein, we propose a related metric to the burning number of a graph, which we name the contagion number (CN), as a better metric for forecasting disease spread. The burning number is a well-known means of measuring the speed of contagion of a graph (Bonato et al., 2016), with contagion defined as a disease spread by close contact. Related to the burning number, our contagion number is defined herein as the smallest number of iterations of the burning number process that is required to infect a given sociogram of people or collection of networked communities. The last step in the burning number process is always to burn a single node, even if every node has already been burned (i.e., the burning number does not constrain a network node from getting the same disease/same variant more than once). The contagion number disregards this, and similar parts of the burning number, because we are concerned only about first-time infections, given the Centers for Disease Control and Prevention (CDC) finding that COVID-19 reinfections are “rare” despite making headlines, thus making our model appropriate for the study of COVID-19 and similar contagions (Centers for Disease Control and Prevention, 2021b).

Our contribution is mostly in answering the question of forecasting how fast a disease can spread (the inherent risk), but it also extends to the practical application of how best to strategically allocate limited resources (i.e., recommends dispensing strategies that can mitigate the impact of epidemic outbreak, therefore alleviate suffering). In particular, our contribution are as follows:

1. We demonstrate that a variant of the NP-hard burning number, coined herein as the contagion number, is a robust metric for forecasting disease spread. We then delineate a new methodology that solves for the contagion number in polynomial time, for any tree.
2. We also simulate for values of the ACN for random trees, as well as for scale-free graphs of size n with varying edge densities, because scale-free graphs are used to simulate social networks, containing preferential attachment (Barabási & Albert, 1999).

Taken together, decision makers can use the CN for worst case scenarios and the ACN for most-likely scenarios of gauging disease spreadability. The result are two indices that decision makers can gauge risk and further dialogue about public policies meant to uphold social welfare objectives in times of pandemic.

Our study is organized as follows. We begin with a brief literature review focused on the most prolific contemporary approach to modeling disease spread followed under COVID-19, before moving on to recent efforts with respect to burning number

The contagion number of a graph. How fast can a disease spread?

research. This is followed by a description of our proposed method, a section with an annotated example for the contagion number, and our simulation results for the ACN. We conclude with recommendations to shape public policy and guide managerial decisions. Some open avenues for future research are also presented.

The authors entered COVID-19 and this research endeavor with both motivation and trepidation. We start with this quote in the foreword of the book Moonshot written by the Pfizer Chairman and CEO, Dr. Albert Bourla:

“The largest lessons learned from tackling these diseases is the importance of partnership and collaboration across sectors to drive global public health for all, and most important to never give up.”

President Jimmy Carter (Carter, 2022)

2. Literature Review

In this section, we stress the importance of good, vibrant, and robust disease modeling techniques such as the well-known SIR model. We then focus on the graph theoretic research on the BN of a graph, as an alternate means of studying disease spread. We conclude the literature review by providing a backdrop to the CN, via the literature on the BN.

2.1. Literature Review: Importance of good techniques and contemporary approaches

Pandemics are borderless and non-discriminatory, hence, the need for robust models that can provide decision maker insight for needed solutions and strategies, when managing societies stricken with disease spread. In this light, the Centers for Disease Control and Prevention issues what is known as the ensemble forecast, with input from a multitude of diverse sources. At the time of the writing of this paper, there are over 30 qualified entities who all contribute to our understanding of forecasted COVID-19 cases. (GitHub, 2021). There are also models for cases (Centers for Disease Control and Prevention, 2021c) as well as models for forecasting deaths (Centers for Disease Control and Prevention, 2021d).

The most well-known disease spread model, the SIR model, is the forecasting model that is most used to study the spread of COVID-19 (Liu et al., 2019). Vastly different than the graph-theoretic approach herein, the SIR model is a compartmental model in which individuals are either susceptible (S), infected (I), or recovered (R). Extensions to the basic SIR model include Markovian transition models (Chen et al., 2020), models that incorporate those that fail to recover (D), and models that incorporate those that are simply exposed (E). For a thorough discussion of the difficulty of utilizing the SIR model, as well as exposure to other viable models, we refer the reader to “Why Modeling the Spread of COVID-19 is so Damn Hard” (Hutson, 2020).

To wit, no known literature is similar to our model in approach or outcome: A new polynomial time metric based on the NP-hard BN of a graph.

2.2. Literature Review: Burning number

A relatively new parameter called the burning number of a graph, $b(G)$, has been used to gauge the speed at which information, alarms, or contagions can cover, alert,

The contagion number of a graph. How fast can a disease spread?

or contaminate an entire network, respectively, with the lower the burning number, the faster the spread (Bonato et al., 2016). Noting the problem is NP-hard, Šimon et al. (2019) developed an algorithm that identifies good but admittedly not optimal nodes to burn, and this serves as a call for more research, as recent research has addressed different approaches across networks having varied characteristics but is far from comprehensive.

The BN is also a good means of understanding spread across a social network (Bessy et al., 2017; Bonato et al., 2016). Closely related to the burning number, the contamination number, gives additional insight into networks to which graph theory has applicability. To wit, Bonato et al. (2021) published the best known upper bound for the burning number of a graph:

$$b(G) \leq \left\lceil \frac{8 + \sqrt{12n + 64}}{3} \right\rceil \quad (1)$$

where $\lceil x \rceil$ is the greatest integer function (ceiling) of x and n is the cardinality of the node set in the graph, G .

Given these efforts, it becomes clear that the burning number is a new and emerging area of study. We also know that it is a sound metric for how fast diseases can spread, an able proxy for the efficacy of social distancing, and that the burning number problem is NP-hard, even for trees.

Hence, our effort delineates a new but related, polynomial time metric for measuring how fast a disease can spread, that we coin the contagion number. The difference between the burning number and the contagion number is that the contagion number is only interested in the number of steps, days, or transitions it takes to infect (burn) the entire network, as opposed to the burning number, which counts steps, days, or transitions, irrespective of whether the nodes involved have already been infected (burned). Therefore, we note here that the contagion number (CN), as defined, is also a lower bound to the burning number.

Moreover, since viruses are not “smart” (i.e., they do not choose the optimal locations needed to infect a particular sociogram of people or networked communities as fast as possible), we also present the ACN as a metric for measuring how fast a disease will spread. Essentially, the CN is about what a disease could do (i.e., a worst-case scenario) whereas the ACN is about what a disease would do (i.e., a most-likely scenario). It is thought that both metrics could be used by public policy and managerial decision makers.

3. Methodology

In this section we formally define the BN, as well as our proposed CN. We provide a step-by-step delineation of our new method for solving for the CN. Our proposed new methodology solves for the CN in polynomial time, for trees. We then employ a Python® code (version 3.6 and higher) Monte Carlo simulation for the ACN of a graph, to provide realistic estimates of how fast a disease would propagate across a social network. To aid in this section we first provide some nomenclature in Table 1.

The contagion number of a graph. How fast can a disease spread?

Table 1. Nomenclature

Term	Definition
n	The number of nodes in a graph; the order
$\text{DEG}(n)$	The degree of node n ; how many incident
CEIL denoted by $\lceil x \rceil$	Greatest integer function
FLOOR denoted by $\lfloor x \rfloor$	Smallest integer function
Eccentricity; $e(\text{node})$	Distance between two nodes with unity edge
Radius of a tree/graph	The minimum eccentricity of the tree/graph
Diameter of a tree/graph	The maximum eccentricity of the tree/graph

For a more thorough background on graph theory and its basic nomenclature, the reader is referred to the classic text by West (2001). For a brief synopsis specifically on the diameter, radius, and eccentricity of a graph the reader is referred to Weisstein (2022).

3.1. Methodology: Burning number definition

The generally accepted method to burn a graph starts with burning a single node at time $t=0$. Then one indexes the clock by one unit (e.g., a day, an hour, or a plane flight by the infected person) and burns another node at time $t=1$, while, simultaneously, the original fire set at time $t=0$ spreads to and burns all its the nearest-neighbor nodes. Again, one indexes the clock by one unit and burns another node at time $t=2$, with all the burning nodes spreading to and burning all their nearest-neighbor nodes (i.e., nodes burned at time $t=1$ have now burned all their nearest neighbors, as well as the second echelon nearest neighbors of the node burned at time $t=0$).

This process of lighting a fire continues, ad nauseam, until all of nodes in the graph are burned. The last step in this process is to locate a fire of radius zero and declare the cardinality of the burning node sequence as a feasible solution to the burning number. The problem of finding the smallest such cardinality in this manner is obviously sequence dependent and is easily shown to be NP-hard. Except in very rare instantiations, which are known as perfect burns, and defined as a situation in which no node is burned more than once during the process of burning a graph, the burning number process will burn several nodes more than once (aka overburning or charring).

3.2. Methodology: Contagion number definition

The contagion number, on the other hand, is the smallest number of infections that need to be located (as opposed to fires to be set) on a graph, so that the whole graph becomes infected, without infecting the same node more than once. Whereas the burning number is about the complete set of burns of different sizes, the contagion number ignores iterations of the process that simply infect the same node more than once.

3.2.1. Methodology: Contagion number definition: Formal delineation

More formally, let G be a connected graph and (v_1, \dots, v_j) be a sequence of nodes in $V(G)$, which will be called hubs and colored red, all other nodes are non-hubs and colored white. For each integer $i = (1, 2, \dots, j)$, apply the following rules in order:

1. Any non-hub that is currently colored red colors each of its white neighbors red.

The contagion number of a graph. How fast can a disease spread?

2. After the non-hubs have colored their respective white neighbors red, we next allow the hub v_i to color its white neighbors red.

If all of $V(G)$ is colored red after j time steps have been performed, then the sequence (v_1, \dots, v_j) is called a contagion sequence of G . The length of the shortest contagion sequence in G is the contagion number of G .

3.3. Methodology: New method for the contagion number

Let $G(n, e)$ be an undirected graph with n nodes and e edges. Let $b(G)$ be the burning number of the graph G . The burning numbers of all the possible spanning trees of a graph, is equal to or greater than the burning number of the original graph (Bonato et al., 2016). At one extreme, the burning number of any completely connected graph (i.e., everyone on the planet personally interacts with everyone else) is 2. However, at the other end, no one interacts with anyone else (i.e., we are all perfectly socially distanced) such that no burning or infection spread takes place. Obviously, human interactions fall somewhere in between. Nonetheless, the $b(G)$ on a generalized graph can be trivial, whereas, except for paths, the $b(G)$ for a tree is always NP-hard. In other words, solving for the $b(G)$ on a tree is as hard as solving for $b(G)$ on a graph. Taken in concert, this means we can focus our efforts on spanning trees for a particular area or region, as the burning number for spanning trees is always an upper bound for the/any underlying graph.

Let $T(n, e)$ be an arbitrary undirected tree. (It is important to bear in mind that previous attempts in top mathematics journals to solve for the $b(T)$ have all noted that the problem is NP-hard (Bessy et al., 2017; Bessy et al., 2018; Liu et al., 2019). The methodology in all these previous attempts starts with setting a fire at time $t=0$ at an arbitrary node. This fire would then grow, at every iteration, into a sub-tree of radius k at time k (where k is just a possible burning number of the tree, but not necessarily the smallest such one). The second fire set at time $t=1$, would also grow in a catawampus manner, to a sub-tree of radius $k-1$ on day k , and the last fire set at time $t=k$ at yet another node, will simply be a sub-tree of radius zero (i.e., a single node) at time k . This is the accepted approach in the literature to solving for the burning number. The problem is in finding the smallest such number of iterations.

We posit that there is an issue with using this accepted approach to find the smallest number of iterations. The problem is not that fires grow; that is what fires do. The problem is that, at every iteration, they grow with respect to their original location. Thus, all of the previous attempts in the literature were essentially trying to solve a complex, sequence-dependent dynamic location problem: Locate each of the fires that will all end up being different sizes and different nodal cardinalities, sequentially, to determine the minimum number of such fires needed to burn the entire network.

So, we began to study our research question as a location problem: A generalized set covering location problem¹, to be exact, except that the radii of all the individual covers are of different sizes. It should be noted here that the generalized set cover location problem on a graph is NP-complete (Garey & Johnson, 1979, p. 222). However, the generalized set cover location problem on a tree is known to be exceedingly simple: It is solved to optimality with the greedy algorithm. Our location problem then became how to handle the varying radii of all the covers.

¹ The set cover location problem was first described by (ReVelle et al., 1976). Literature surveys in location analysis in general include Brandaue & Chiu, 1989, Hale & Moberg, 2003, Drezner & Hamacher, 2002, Nickel & Albandoz, 2005, and the award winning Laporte et al., 2019. Surveys specific to covering location problems can be found in Schilling et al. (1993) and Snyder (2015).

The contagion number of a graph. How fast can a disease spread?

3.3.1. Methodology: New method for the contagion number: Key finding

We then noticed that in all the previous instantiations of the BN problem in the literature, once a tree (or a graph, for that matter) was burned completely, it was always covered with a sequential set of burns, all of which were 1-center trees of radii $\{k, k-1, k-2, \dots, 2, 1, 0\}$. Put another way, every feasible burning number solution was always covered with a set of burns of radii $\{0, 1, 2, \dots, k-2, k-1, k\}$. This is key.

This led to our idea of approaching the entire problem in a backwards manner: Locate and burn what was in the prescribed and accepted method in the literature as the last burn (a 1-center tree of radius 0; a single node) first. It is then fixed and does not grow. Simply put, locate a single node, and burn it. Then locate what was the second to last burn (a 1-center tree of radius 1) second. It is also fixed and does not grow. Locate a very small tree and burn it. Then locate what was the third to last burn (a 1-center tree of radius 2) third. It is also fixed and does not grow. Locate a slightly larger tree and burn it. Continue this process in an orderly manner, until all the nodes are burned. This backwards approach is obviously simpler.

3.4. Methodology: A small annotated example

In this section, we delineate through the use of an example of our new method, to solve for the CN of a tree. This example, albeit small, clearly shows the efficacy of our proposed new method. We first note that the burning number on a path with n nodes is trivial and is strictly equal to $\lceil \sqrt{n} \rceil$. We further note that the contagion number on a path with n nodes is less than or equal to $\lfloor \sqrt{n} \rfloor$. We also note that the burning/contagion numbers of any tree with n nodes is greater than or equal to the burning/contagion numbers for a generalized graph with n nodes. Hence, it is the tree that offers the most elusive, largest, and non-trivial problem, with n nodes.

The method prescribed here is to iteratively choose the nodal center location of a fixed radius infection (a set cover of radius k) such that the sum of the eccentricities of all the remaining non-leaves (all nodes of $\text{DEG} = 2$ or higher) is minimized. In other words, after the current fixed radius set cover is located, it leaves behind a tree that has the smallest total distance sum, to the remaining tree's center, for every node of degree 2 or greater. Next, the infected nodes are removed from further consideration, the fixed radius infection k is increased by one, and the process is repeated until all nodes are infected.

For expository purposes, and without loss of generality, we draw the tree of interest as shown in Figure 3, with the 1-node or 2-node center at the bottom and progressive "levels" growing up and numbered accordingly. In Figure 3, the tree has a diameter of 15 and, hence, is 2-node center (all odd diameter trees have a 2-node center, while all even diameter trees have a 1-node center). Also, we label all the "major branches" of the tree. This choice of these labels are completely arbitrary and mostly for ease of exposition. In Figure 3, we chose A through G to be our major branches. We further note that branches with peripheral points, A, B, and C, are on the other side of the 2-node center than branches D and E.

The "backwards" process (and, if necessary, at the beginning of every iteration) is started by identifying the best location for the zero-radius infection: a single node. The CN, as it is defined, rarely needs the single node infection to define it. Thus, before proceeding with the annotated example, this point is demonstrated. For example, on paths where the burning number is at its greatest, per the cardinality of the node set, from a path of one node (a single node) up to and including a path of one thousand nodes, only 3.1% of them ($n = 1, 4, 9, 16, 25, \dots, 900, 961$) need to use the single node burn to ascertain the burning number. As noted above, these 3.1% are known as

The contagion number of a graph. How fast can a disease spread?

perfect burns, as no node is burned more than once. A perfect burn for the P9 path is depicted in Figure 1 below. It has a burning number of 3 and a contagion number of 3.

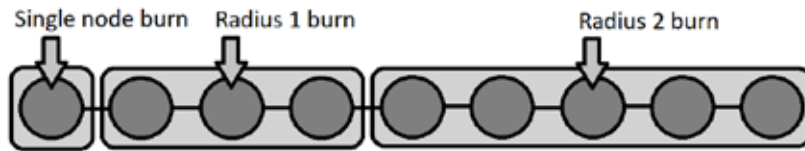


Figure 1. A path of length nine showing the three burns

A path of length 8, depicted in Figure 2 below, also has a burning number of 3. It has one overburned node. However, a path of length 8 has a contagion number of 2, as the contagion number does not include the unnecessary single node burn, given its focus on only first-time infections. Moreover, a path of length 5 (not pictured) has a burning number of 3, but a contagion number of 1. All that is needed is a radius 2 burn and does not include the radius 1 burn or the single node burn used in the burning number calculation.

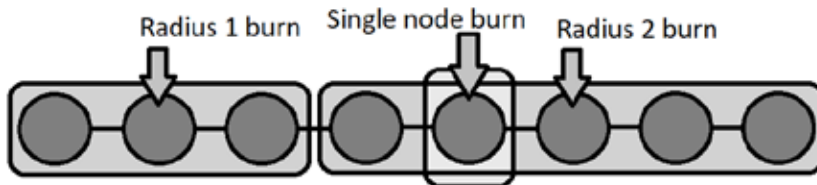


Figure 2. A path of length eight showing the three burns and an overburn

Note that if the locations of the radius 1 burn and the radius 2 burn are as shown in Figure 3 (i.e., they are efficient, per se) then the location of the single node burn is moot.

In terms of the burning number/contagion number of a tree, without at least one of these conditions, this means that the single node burn location is of no consequence as it will be overburned, regardless. In such cases, the CN will be at least one less than the burning number.

Hence, there is a special proviso for the single node infection in our methodology. In general, the location rule herein is to choose the current iteration's infection center node, as the node that minimizes the sum total of the eccentricities of all the remaining non-leaves once the current iteration is burned away. For the single node infection to be of any consequence, that means at every iteration, there would need to exist a peripheral point that is by itself, per se. This is to say, a leaf with its neighbor node/support node having a degree of 2. If such a leaf does not exist, we skip the single node infection for the current iteration and move on to the next infection radius. In Figure 3, at the first iteration, node A8 is a peripheral leaf that is by itself (its neighbor in level 7 has degree 2), so we infect it and continue. After infecting A8, we note that the tree's 2-node center location has not changed, and we now look at all remaining uninfected nodes. We choose B7 as it minimizes the sum total of the remaining non-leaves' eccentricities. After infecting B7 and its neighbors, we again note that the tree's 2-node center location has not changed, and we now look at all remaining uninfected nodes. We subsequently choose D6 as it minimizes the sum total of the eccentricities of the remaining non-leaves.

The contagion number of a graph. How fast can a disease spread?

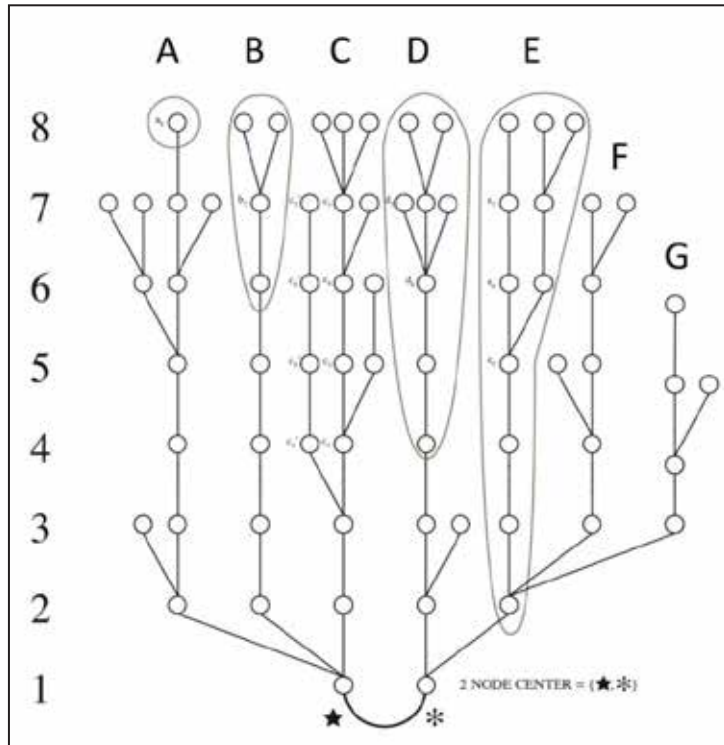


Figure 3. Original tree showing first four burns

We note here that there can be ties in this process and it is in this choice that the problem remains NP-hard (otherwise $P = NP$). Thus, our methodology does not solve an NP-hard problem: the burning number on a tree. It solves for a proxy of the burning number that we coin the contagion number, in a deterministic fashion, shedding light on disease spread for the benefit of managerial and public policy decision makers.

After infecting D6 and all nodes within a radius of 2, we again note that the tree's 2-node center location has not changed, and we now look at each remaining uninfected node. We choose E5 as it, again, minimizes the sum total of the eccentricities of the remaining non-leaves.

After these four iterations, the diameter and, hence, the center of the tree have both changed. The remaining tree is shown in Figure 4 below. It now has a diameter of 14 and, hence, is now a 1-center tree with peripheral points still at level 8, on major branches C and F. It also has what we term here as a "ghost node" at E3, which is needed for our new method, going forward. To keep the tree connected, previously infected "ghost" node E3 and its incident edges are designated by dotted lines as shown in Figure 4 below.

The contagion number of a graph. How fast can a disease spread?

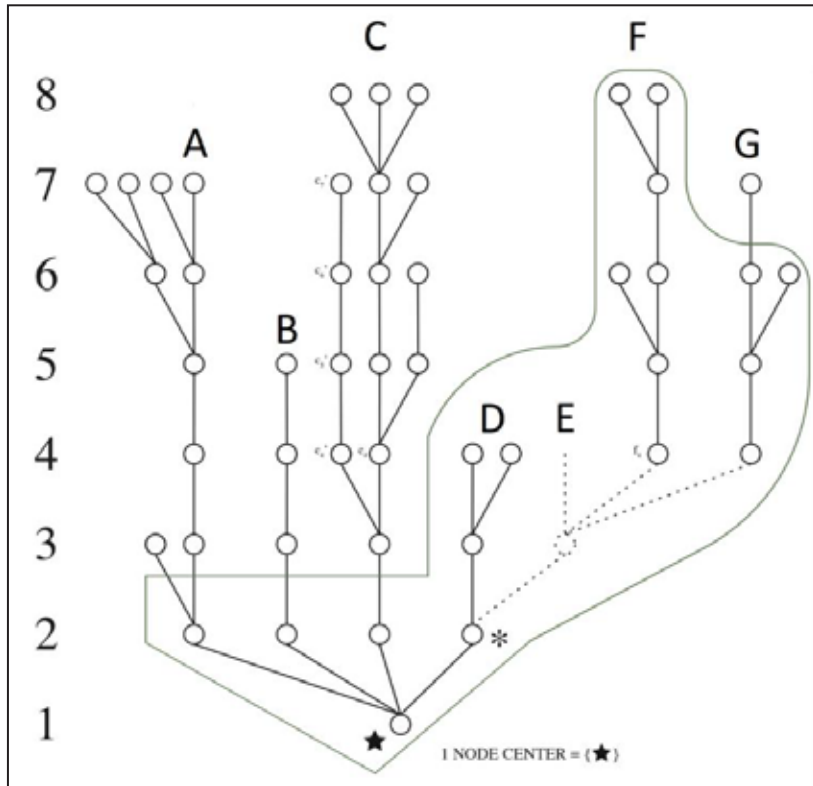


Figure 4. Remaining tree after four iterations

We again investigate all the remaining uninfected nodes. We choose F4, as it minimizes the sum of all the eccentricities of the remaining non-leaves in the tree that remains, which, since the diameter and center have changed again, is depicted in Figure 5 below.

We notice that the remaining tree in Figure 5 now has a diameter of 13, so it is once again a 2-node center tree. We take special note that the branches on either side of the 2-node center have realigned. We are now looking at all remaining unburnt nodes on this tree.

The contagion number of a graph. How fast can a disease spread?

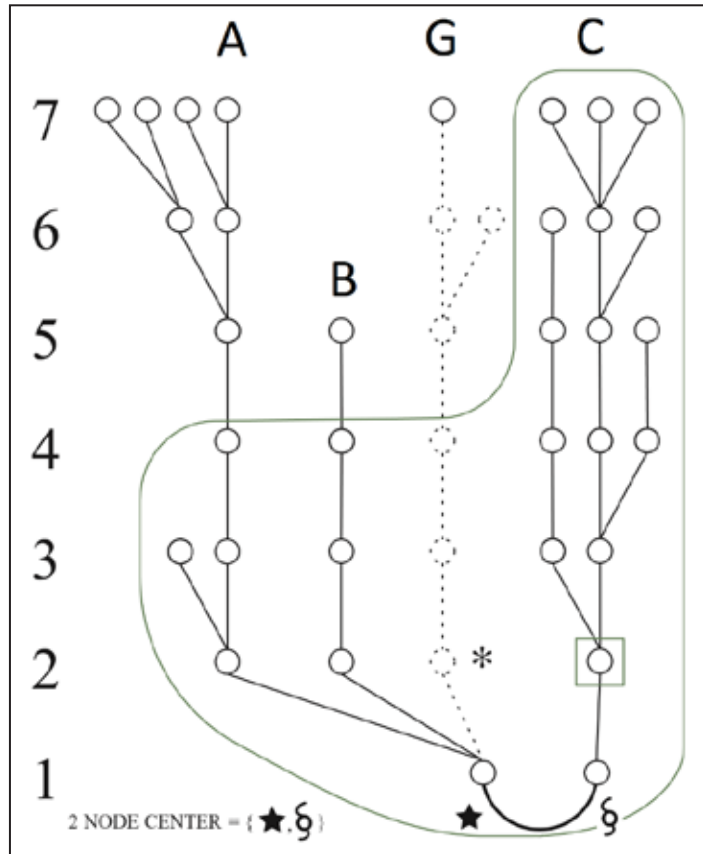


Figure 5. Remaining tree after five iterations

We choose C2 (inside the square box in Figure 5) as it minimizes the sum of eccentricities of all the non-leaves that remain after we infect all the nodes. We are now left with a shell of our original tree, and it is depicted in Figure 6 below.

The remaining tree is now of diameter 12, so it is a 1-node center tree. Since we now need to infect a 1-node center tree of radius $k = 6$, as per the pseudo code delineated in the next section, satisfies one of the stopping conditions, we simply choose the 1-node center and infect all the remaining uninfected nodes. So, after seven iterations ($k = 0, 1, 2, 3, 4, 5$, and 6), we have infected the entire tree and, in this case, the contagion number is 7.

It should be pointed out here that in the hundreds of trial efforts by the authors, utilizing this method on arbitrary trees, the BN was also found correctly via our new method, in most of them. This is admittedly anecdotal, but it is hoped that a future refinement of our methodology, could result in BN solutions for at least some classes of trees. For now, we simply postulate that determining the CN via our “backwards” method results in polynomial solution times. Utilizing the fixed radii of the set covers, approaching the problem backwards, and the “leveled configuration” of the tree in our example, all provides much needed insight and direction, for decision makers navigating complex situation such as those brought on by disease spread.

The contagion number of a graph. How fast can a disease spread?

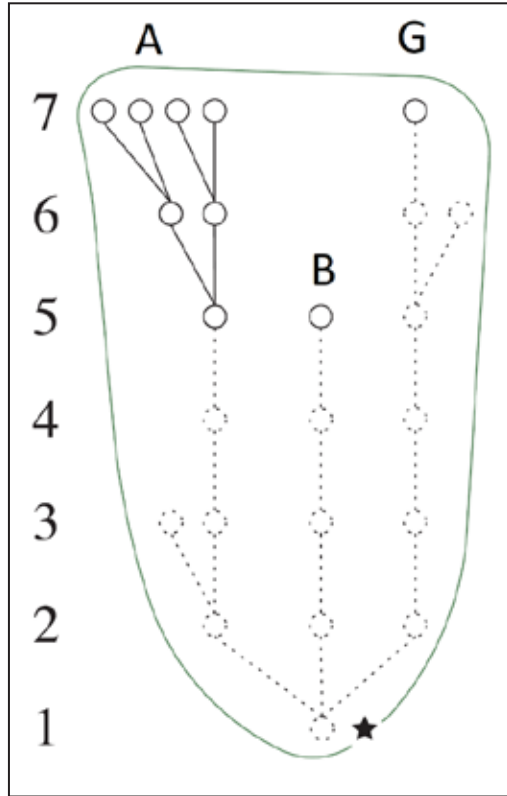


Figure 6. Remaining tree after six iterations

3.5. Methodology: Pseudo code algorithm

Pseudo code is provided to additionally demonstrate that the method prescribed here is to iteratively choose the nodal center location of a fixed radius infection (a set cover of radius k), such that the sum of the eccentricities of all the remaining non-leaves (all nodes of $DEG = 2$ or higher) is minimized. A tree that has the smallest total distance sum to the remaining tree's center, for every node of degree 2 or greater, remains, and the methodology continues by removing the infected nodes from further consideration, increasing the fixed radius infection k by one, and repeating this process until all nodes are infected.

This process, admittedly, requires some substantial calculations along the way. However, at worst via brute force, this would require calculating the eccentricity for all of the non-leaves for all of the possible burn center locations for each of the k iterations. This first part of the brute force process is no larger than $n - 2$ non-leaves by no larger than $n - 2$ possible burn centers. Since k is approximately equal to \sqrt{n} , the computational complexity of our proposed method is then approximately equal to $(n)(n)(\sqrt{n}) = n^{2.5}$.

Our proposed method further decreases this computational complexity by reducing the set of possible current iteration burn locations, to only those nodes on a diameter that are also a distance of k (the current iteration index) from at least one peripheral point. This reduces the feasible solution space significantly.

A trivial upper bound to the BN of a tree is the radius plus the cardinality of the centers (1 or 2). The pseudo code delineated below will terminate in no more than this

The contagion number of a graph. How fast can a disease spread?

upper bound number of iterations. Each of the steps involved is deterministic and computationally polynomial with respect to n . Hence, our entire process is polynomial with respect to n .

Our proposed methodology is shown below in pseudo-code form. There is a flag, z , and an iteration counter, k , embedded in the code. The flag is for placing the zero radius single node if and only if a particular iteration has a support vertex (a node next to a peripheral leaf) has degree 2. The counter is for tracking the iterations and has three distinct and complete stopping conditions.

Pseudo-code to find the contagion number of a tree

1. Set Zero Radius $z = 1$
2. Set Infection Radius $k = 0$
3. If any peripheral leaf has a support vertex that has $\text{DEG} = 2$ then infect any one such peripheral leaf with a zero radius infection and set $z = 0$ and set $k = k + 1$
4. Set $k = k + 1$
 - a. If the current iteration diameter is even and k is greater than or equal to the current iteration radius, then STOP: k is the contagion number
 - b. Elseif the current iteration diameter is odd and k is strictly greater than the current iteration radius, then STOP: k is the contagion number
 - c. Endif
5. For every unburnt node, determine if it were to be the burn center of the current iteration the sum of the eccentricities of all would be remaining nodes of $\text{DEG} = 2$ or higher.
6. Infect the node with the minimum sum per above and every node within the k -neighborhood of this node. Break ties arbitrarily.
7. If all nodes are infected, then STOP: k is the contagion number.
8. If remaining tree is cut and has become unconnected by the current infection iteration, then keep uninfected nodes attached with infected edges and infected nodes.
9. Determine the remaining tree's center(s), diameters, radius, and peripheral pendants.
10. If $z = 1$ then goto 3 else goto 4

3.6. Methodology: A large computer-generated example

To demonstrate our methodology's efficacy on larger trees, we programmed a script for our pseudo code in Python©. For 10 random trees of 1000 nodes, the algorithm found a CN as small as 19 and as large as 28. This comports with the theoretically largest burning number on a tree with 1000 nodes which is 32 for a path on 1000 nodes. Our generalized Python© code for both the CN (CN) of a tree and the ACN of a graph is located in Appendix A.

3.7. Methodology: Simulations of the average contagion number

Most of the focus of this paper thus far has been on the worst-case scenario (i.e., how fast a disease could spread). It resulted in a metric that is a variant to the well-known NP-hard burning number, that via the utilized "backwards" methodology, is always solvable in polynomial time. In this section, the focus now shifts to delineating a most-likely scenario (i.e., how fast a disease would spread). This is important because diseases aren't smart and they don't pick the optimum locations to infect an entire network as fast as possible. Diseases are, however, opportunistic and spread through human social networks. This section thus delineates, with just a slight modification to our CN methodology, an estimate of the ACN.

The contagion number of a graph. How fast can a disease spread?

The slight modification is that, instead of choosing an optimum or near optimum host to infect at every step of the way, such that the entire network is infected as fast as possible, an uninfected node is simply chosen at random, and the number of steps taken to infect the entire network is counted. Via our Python© code (see Appendix A), simulations were run on various graphs, and we now summarize these results in the tables and figures below. In Appendix B, an example of contagion spread in a 350-node network is provided, using the Lancichinetti et al. (2008) benchmark graph.

To begin we note that human connections (i.e., social networks) are not modeled well with random trees. Sociologists model human connections through graphs called sociograms, where the nodes represent people, and the edges represent a social interaction between two people. Therefore, we modified our ACN code to simulate sociograms on what are known as scale-free networks (see Barabási & Albert, 1999).

3.7.1. Methodology: Simulations of the ACN: Scale free graphs

A note on scale free graphs: As observed by Barabási & Albert (1999), many real-world networks have the property that their degree distribution approximates a power law. This was somewhat unexpected as the degree distribution of a random network should instead follow a Poisson distribution, thus indicating that real world networks are not generated in the same way as random graphs. That is, the famous Erdős & Rényi (1960; reprinted 2011) model for random graph generation is inadequate for modelling real world networks. In fact, real world networks seem to be generated by a method of *preferential attachment*, whereby the probability of newly added nodes being connected to old nodes is dependent on the degree of the old nodes. This leads to situations, as is seen in the world-wide-web network, where there are more than would be expected nodes of very high degree (called hubs), while also more than would be expected nodes of very small degree (which essentially does not happen in random networks).

The networks whose degree distribution follows a power law were called *scale-free* by Barabási and Albert, and many generative models have been proposed to create them. Ideas related to scale-free networks and preferential attachment, in a wide range of fields, were also studied prior to Barabási and Albert (see: Barabási, 2013). In line with these results and this history, this is why we investigate the CN herein on scale-free networks generated with the Barabási and Albert model, on the assumption that social networks, where a virus might spread, are approximately of this form.

3.7.2. Methodology: Simulations of the ACN: Percent connectedness

Social networks also have different edge densities with respect to the nodes in a particular sociogram. The Barabási and Albert method utilizes a parameter, m , to designate a particular social density, per se. It is an integer that is loosely based on the edges in terms of n emanating from a particular node. It is maximal at $n/2$. We summarize some results for ACNs from the Barabási and Albert method generated scale-free graphs at minimal, 25th percentile connectedness, 50th percentile connectedness, 75th percentile connectedness, and maximally connected scale-free graphs in Table 2 below.

The contagion number of a graph. How fast can a disease spread?

Table 2. Estimates for the average contagion numbers of scale-free graphs of order n

<i>Order</i>	$m=1$	$m=n/8$	$m=n/4$	$m=3n/8$	$m=n/2$
16	4.1	3.4	3.0	2.1	2.2
24	5.3	3.3	3.0	2.0	2.7
32	5.9	3.3	3.0	2.0	2.6
40	6.5	3.0	3.0	2.0	2.8
48	6.7	3.1	3.0	2.0	2.7
56	7.0	3.0	3.0	2.0	2.6
64	7.5	3.0	3.0	2.0	2.7
72	7.5	3.0	3.0	2.1	2.7
80	7.2	3.0	3.0	2.0	2.9
88	7.5	3.0	3.0	2.0	3.0
96	8.1	3.0	3.0	2.1	3.0
104	7.9	3.0	3.0	2.0	3.0
112	7.9	3.0	3.0	2.0	3.0
120	8.2	3.0	3.0	2.0	3.0

The table shows that, in general, as order increases, so does the ACN. What is interesting here is for even loosely connected scale-free graphs ($m=n/8$), as long as they are not trees, the ACN is the same as for scale-free networks with much higher edge density ($m=n/2$). Moreover, for every column in the table except the first, the ACN is either 2 or 3 after rounding to nearest integer, while for scale-free trees, it increases from 4.1 to 8.2. From this we can infer that any virus spreading through a community which forms a network that is at least approximately scale-free, if it spreads by a mechanism similar to the CN, will infect the entire community in roughly the same amount of time regardless of how dense the corresponding network is. The exceptional case of a scale-free tree is an outlier in this context.

A final note about the table is that the authors note that the 75th percentile connectedness column ($m=3n/8$) behaves differently than the other columns, in that it has ACN close to 2 for every order shown, while the others ($m=n/8$, $m=n/4$, $m=n/2$) have ACNs close to 3 for every order shown. Again, our generic Python© code for both the CN of a tree and the ACN of a graph is available in Appendix A.

Because the contagion number of a network in relation to the number of nodes is of interest, we next implement a standard linear regression model on the data generated in prior sections. However, in order to apply linear regression, we first implemented a common data transformation often used in data science. Namely, we used the log (base e) of the number of nodes for our input feature and the average computed contagion number for our respective labels. In the case of the Barabási and Albert graphs, the mean squared error was roughly 0.18 on average across many different instances of test-train splits; for example, see Figure 7. Our trained regressor did exceptionally well when plotted against the seen minimum and maximum contagion numbers over the Barabási and Albert graphs as can be seen in Figure 8. Our results suggest that labeling networks by numerical values may be a good predictor of how many hubs on average will be needed to infect the whole network, via machine learning techniques.

The contagion number of a graph. How fast can a disease spread?

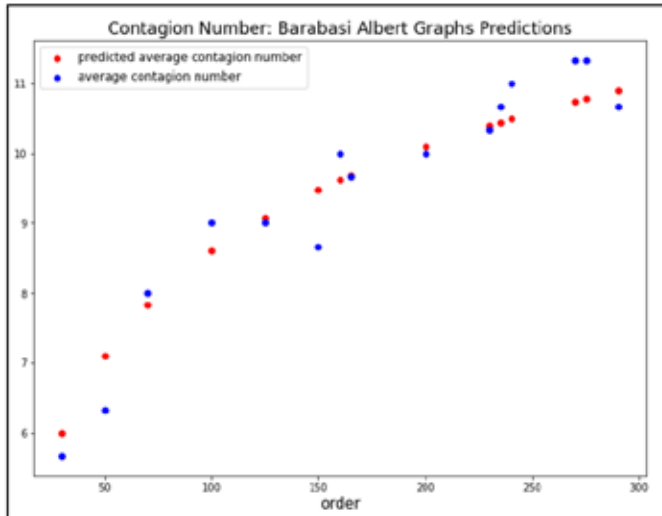


Figure 7. Predictions made on test data

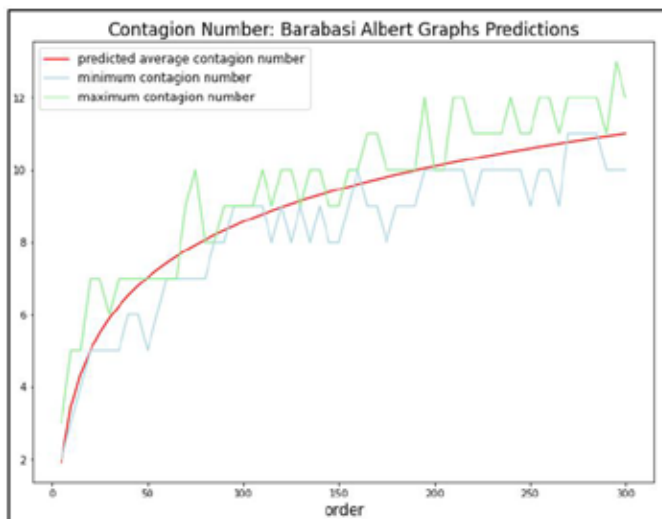


Figure 8. Regressor plotted against CN over Barabási and Albert scale-free graphs

4. Discussion

We began our research with the notion that there is something to be gained by having an understanding of disease spread. When considering that “Manufacturing the vaccine had left us with a product as fragile as a snowflake,” (Bourla, 2022 p. 94), the need to protect limited supplies becomes apparent. With social welfare top of mind when vaccines are limited in supply, we propose that a network structure dependent allocation policy can be used to control the spread, as well as meter the impact, of a disease like COVID-19.

At the heart of this study is the finding that, whether referring to the CN or ACN, structures that look like trees will have higher values (i.e., slower contagion spread), than their more connected counterparts, and that more connected counterparts will

The contagion number of a graph. How fast can a disease spread?

have almost identical contagion spread, regardless of the number of connections, for networks with the same number of nodes (i.e., people). Social distancing, along with obeying stay at home orders for all non-essential personnel and properly/diligently wearing masks can, in effect, change the shape of the network to be more tree-like. Social distancing is believed to be an effective strategy for decreasing pandemic spread such as COVID-19 (Liu et al., 2019).

Table 3 and Figure 9 are provided to demonstrate how disease spread can be slowed when connections are broken in a social network. Note that the ACN becomes further apart as order increases, speaking to large-scale social distancing. However, we do acknowledge that sometimes social distancing can be a hard-fought achievement, as it is not always possible to simply maintain 100% social distancing (e.g., such as essential workers when the need exists to not turn off essential services).

Table 3. Comparison of min, average, and max CN for trees and for scale-free graphs

<i>Order</i>	<i>Min Tree</i>	<i>Avg Tree</i>	<i>Max Tree</i>	<i>Min Scale-Free</i>	<i>Avg Scale-Free</i>	<i>Max Scale-Free</i>
5	3	3.33	4	2	2.33	3
10	4	4.33	5	3	4.00	5
15	5	5.67	6	4	4.33	5
20	6	6.33	7	5	5.67	7
25	6	6.67	7	5	6.00	7
30	8	8.00	8	5	5.67	6
35	8	8.33	9	5	6.00	7
40	7	8.00	9	6	6.67	7
45	8	9.00	10	6	6.33	7
50	8	8.67	9	5	6.33	7
55	9	9.33	10	6	6.67	7
60	9	9.67	10	7	7.00	7
65	9	9.33	10	7	7.00	7
70	9	9.67	10	7	8.00	9
75	10	10.33	11	7	8.33	10
80	10	11.33	12	7	7.67	8
85	11	11.67	12	8	8.00	8
90	12	12.00	12	8	8.67	9
95	11	11.33	12	9	9.00	9
100	11	11.33	12	9	9.00	9

The contagion number of a graph. How fast can a disease spread?

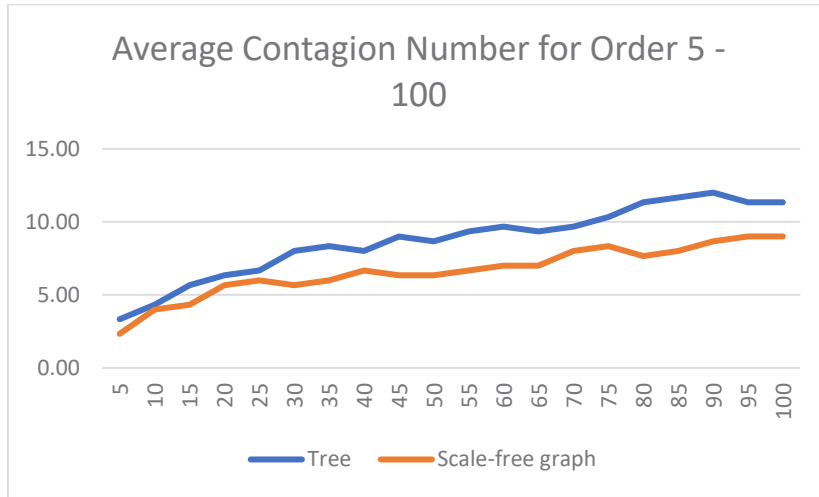


Figure 9. Average contagion number for order 5 - 100 for trees and scale-free graphs

5. Conclusions, limitations, and future research

This study delineated a new graph theoretic method, known as the contagion number, to model worst-case disease spread throughout a sociogram of people or collection of networked communities. The new method delineated here is simple and robust; it is only slightly more complicated than the greedy heuristic. It takes advantage of the fixed radii afforded, to locate each of the burns (set covers), in an orderly and efficient manner. Our new method discounts any set covers that cause overburns. Additionally, our methodology’s outcome is deterministic and leads to the CN for any tree in polynomial time. Indeed, taken in concert, the irreverence of our new method with respect to multiple infections of the same node combined with its deterministic, polynomial time algorithm, speaks directly towards the practicality of its use, when compared to the overburns and NP-hardness of the associated BN problem.

This study then generated simulations of the same metric, to delineate a most likely metric of disease spread throughout a sociogram of people or collection of networked communities. The results can best be seen when comparing networks of trees to networks more representative of human social networks (e.g., scale-free graphs). Taken in combination, the implication is that decision makers can utilize both worst-case scenarios and most likely scenarios, when setting public policy and in managerial decision making.

The methodology presented herein is not without limitations. These include the which of the spanning trees of a social network graph should be chosen, the ability to model “dense” social networks is suspect (i.e., the BN and CN of a completely connected graph is 2), and some connections will be “stronger” than others (which is addressed in future resear4ch below).

Future research is aimed at developing models that incorporate additional components, such as how the SIR model has evolved over time (e.g., nodes representing vaccinated individuals), a SIR / Markovian transmission probability contagion number hybrid model of sorts, the stochastic contagion number (i.e., virus transmission along an edge follows some prescribed density function), and the dynamic contagion number (i.e., the cardinality of the node set and/or the edge

The contagion number of a graph. How fast can a disease spread? density of the graph changes over time). It is our hope that the new metric defined herein provides a benefit for public policy decision makers.

Author Contributions: Conceptualization, T.H., R.D. and R.P.; methodology, T.H., R.D. and R.P.; software, R.D.; validation, ALL; formal analysis, ALL; investigation, ALL; resources, N.A.; data curation, N.A.; writing—original draft preparation, M.B. and T.H.; writing—review and editing, M.B. and T.H.; visualization, N.A.; supervision, N.A.; project administration, N.A.; funding acquisition, N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix: The appendices are attached separately.

References

- Adbi, A., Chatterjee, C., Drev, M., & Mishra, A. (2019). When the Big One Came: A Natural Experiment on Demand Shock and Market Structure in India's Influenza Vaccine Markets. *Production and Operations Management*, 28(4). <https://doi.org/10.1111/poms.12948>
- Barabási, A. L. (2013). Network science. In *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* (Vol. 371, Issue 1987). <https://doi.org/10.1098/rsta.2012.0375>
- Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439). <https://doi.org/10.1126/science.286.5439.509>
- Bessy, S., Bonato, A., Janssen, J., Rautenbach, D., & Roshanbin, E. (2017). Burning a graph is hard. *Discrete Applied Mathematics*, 232. <https://doi.org/10.1016/j.dam.2017.07.016>
- Bessy, S., Bonato, A., Janssen, J., Rautenbach, D., & Roshanbin, E. (2018). Bounds on the burning number. *Discrete Applied Mathematics*, 235. <https://doi.org/10.1016/j.dam.2017.09.012>
- Bonato, A., English, S., Kay, B., & Moghbel, D. (2021). Improved Bounds for Burning Fence Graphs. *Graphs and Combinatorics*, 37(6). <https://doi.org/10.1007/s00373-021-02390-x>
- Bonato, A., Janssen, J., & Roshanbin, E. (2016). How to burn a graph. *Internet Mathematics*, 12(1–2). <https://doi.org/10.1080/15427951.2015.1103339>
- Bourla, A. (2022). *Moonshot: Inside Pfizer's Nine-Month Race to Make the Impossible Possible*. New York, NY: HarperCollins Publishers.
- Brandeau, M. L., & Chiu, S. S. (1989). An Overview of Representative Problems in Location Research. *Management Science*, 35(6). <https://doi.org/10.1287/mnsc.35.6.645>

The contagion number of a graph. How fast can a disease spread?

Carter, J. (2022) Foreword to Moonshot. In Bourla, A. (2022). Moonshot: Inside Pfizer's Nine-Month Race to Make the Impossible Possible. New York, NY: HarperCollins Publishers.

Center for Disease Control and Prevention (2021a). CDC Influenza SARS-CoV-2 (Flu SC2) Multiplex Assay (Report No.: LB-135).

Center for Disease Control and Prevention (2021b). Reinfection with COVID-19, accessed November 11, 2021, available at <https://www.cdc.gov/coronavirus/2019-ncov/your-health/reinfection.html>

Centers for Disease Control and Prevention (2021c). COVID-19-Forecasts: Cases, accessed February 9, 2021, available at <https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/forecasts-cases.html>

Centers for Disease Control and Prevention (2021d). COVID-19-Forecasts: Deaths, accessed February 9, 2021, available at <https://www.cdc.gov/coronavirus/2019-ncov/covid-data/forecasting-us.html>

Chen, J., Fu, M. C., Zhang, W., & Zheng, J. (2020). Predictive Modeling for Epidemic Outbreaks: A New Approach and COVID-19 Case Study. *Asia-Pacific Journal of Operational Research*, 37(3). <https://doi.org/10.1142/S0217595920500281>

Drezner, Z., & Hamacher, H. W. (2002). Facility location: applications and theory. In *Facility Location Application and Theory*.

Duijzer, L. E., van Jaarsveld, W. L., Wallinga, J., & Dekker, R. (2018). Dose-Optimal Vaccine Allocation over Multiple Populations. *Production and Operations Management*, 27(1). <https://doi.org/10.1111/poms.12788>

Eiselt, H. A., & Marianov, V. (2015). Applications of Location Analysis. In *Applications of Location Analysis* (Vol. 232). <https://doi.org/10.1007/978-3-319-20282-2>

Erdos, P., & Rényi, A. (1960; reprinted 2011). On the evolution of random graphs. In *The Structure and Dynamics of Networks* (Vol. 9781400841356). <https://doi.org/10.1515/9781400841356.38>

Garey, M. R. and Johnson, D. S. (1979), *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco, CA: W. H. Freeman and Co.

Git Hub. (2021). COVID-19 Forecasting Model Descriptions, accessed February 9, 2021, available at https://github.com/cdcepi/COVID-19-Forecasts/blob/master/COVID-19_Forecast_Model_Descriptions.md

Hale, T. S., & Moberg, C. R. (2003). Location Science Research: A Review. In *Annals of Operations Research* (Vol. 123, Issues 1-4). <https://doi.org/10.1023/A:1026110926707>

Hutson, M. (2020). Why modeling the spread of COVID-19 is so damn hard. *IEEE Spectrum*, accessed June 4, 2022, available at <https://spectrum.ieee.org/artificial-intelligence/medical-ai/why-modeling-the-spread-of-covid19-is-so-damn-hard>

Keskinocak, P. (2021). Infectious Disease Modeling and Informing Decisions. *Production and Operations Management Society Annual Conference Plenary Session* on May 2, 2021.

The contagion number of a graph. How fast can a disease spread?

Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 78(4). <https://doi.org/10.1103/PhysRevE.78.046110>

Laporte, G., Nickel, S., & Saldanha-da-Gama, F. (2019). Introduction to Location Science. In *Location Science*. https://doi.org/10.1007/978-3-030-32177-2_1

Liu, H., Zhang, R., & Hu, X. (2019). Burning number of theta graphs. *Applied Mathematics and Computation*, 361. <https://doi.org/10.1016/j.amc.2019.05.031>

Nickel, S., & Albandoz, J. P. (2005). Location theory: A unified approach. In *Location Theory: A Unified Approach*. <https://doi.org/10.1007/3-540-27640-8>

ReVelle, C., Toregas, C., & Falkson, L. (1976). Applications of the Location Set-covering Problem. *Geographical Analysis*, 8(1). <https://doi.org/10.1111/j.1538-4632.1976.tb00529.x>

Schilling, D. A., Jayaraman, V., & Barkhi, R. (1993). A review of covering problems in facility location. *Location Science*, 1(1).

Šimon, M., Huraj, L., Luptáková, I. D., & Pospíchal, J. (2019). Heuristics for spreading alarm throughout a network. *Applied Sciences (Switzerland)*, 9(16). <https://doi.org/10.3390/app9163269>

Snyder, L. V. (2011). Covering problems. In H. A. Eiselt and V. Marianov (Eds.), *Foundations of Location Analysis*. Berlin, Germany: Springer-Verlag, 109-135.

Weisstein, E. W. (2022). Graph Eccentricity, accessed June 4, 2022, available at <https://mathworld.wolfram.com/GraphEccentricity.html>

West, D. B. (2001). *Introduction to Graph Theory* (2nd ed.) Englewood Cliffs, New Jersey: Prentice-Hall.



© 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).