A Genetic Algorithm for Finding an Optimal Curing Strategy for Epidemic Spreading in Weighted Networks

Clara Pizzuti
Inst. for High Perform. Comp. and Networking (ICAR),
National Research Council of Italy (CNR)
Rende, Italy
clara.pizzuti@icar.cnr.it

Annalisa Socievole
Inst. for High Perform. Comp. and Networking (ICAR),
National Research Council of Italy (CNR)
Rende, Italy
annalisa.socievole@icar.cnr.it

ABSTRACT

Contact networks have been recognized to have a central role in the dynamic behavior of spreading processes. The availability of cost-optimal curing strategies, able to control the epidemic propagation, are of primary importance for the design of efficient treatments reducing the number of infected individuals and the extinction time of the infection. In this paper, we investigate the use of Genetic Algorithms for solving the problem of finding an optimal curing strategy in a network where a virus spreads following the Susceptible-Infected-Susceptible (SIS) epidemic model. Exploiting the N-Intertwined Mean-Field Approximation (NIMFA) of the SIS spreading process, we propose a constrained genetic algorithm which determines specific curing rates to each node composing the network, in order to minimize the total curing cost, while suppressing the epidemic. Experiments on both synthetic and real-world networks show that the approach finds solutions whose curing cost is lower than that obtained by a classical baseline method.

CCS CONCEPTS

• **Computing methodologies** → **Search methodologies**; • **Applied Computing**; • **Networks** → Network reliability;

KEYWORDS

Epidemic Spreading, Genetic Algorithms, Complex Networks

ACM Reference Format:
Clara Pizzuti and Annalisa Socievole. 2018. A Genetic Algorithm for Finding an Optimal Curing Strategy for Epidemic Spreading in Weighted Networks. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18).* ACM, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3205455.3205508

1 INTRODUCTION

Epidemic spreading in networks is an intensively studied problem in many different fields, including medicine, social science, computer science, biology, physics [13]. The diffusion of viruses among people, as well computers, in fact, represents a threat for society and organizations. The development of policies to control the spreading process is thus a crucial problem, with practical applications in contexts such as public health and network security. Epidemiological models to predict the virus propagation in a population of individuals have been defined since the twenties [11]. In the last years, because of the increasing use of networks to model real-world systems and relationships among people, epidemic models have received an increasing attention from researchers. In fact, a population can be represented with a graph $G = (V, E)$, where V is the set of individuals, called nodes or vertices, and $E \subseteq V \times V$ is the set of links, called edges, between them. Infections can thus be transmitted from an infected individual to his susceptible connected neighbors.

Epidemic models are applied in several spreading processes on complex networks, as viral marketing, the adoption of an idea or a product promoted by an influencer in a social network, like Facebook and Twitter, propagation of computer viruses through the web, information diffusion, social behavior. In general, the problem of controlling of the evolution of epidemics, when the network resources are limited, is solved with optimization techniques. Epidemic models, thus, in order to suppress the epidemic spreading, define a search problem where an objective function must be optimized. For instance, find the minimum number of nodes to immunize, or the number of connections to remove between individuals [3, 13], distribute vaccine to control epidemic outbreaks [2, 19, 20], allocate recovery resources at the lowest cost to prevent the indefinitely persistency of an epidemic [14].

In this paper, we investigate the use of Genetic Algorithms (GAs) for solving the problem of finding an optimal curing strategy in a network where a virus spreads following the *Susceptible-Infected-Susceptible (SIS)* epidemic model of spreading on a weighted graph, by relying on the heterogeneous *N-Intertwined Mean-Field Approximation (NIMFA)* of the SIS spreading process, introduced by Van Mieghen et al. [23, 24]. In this model, it is assumed that each node has its own curing rate and can infect each of its neighbors with different infection rates.

We propose to solve the problem by exploiting a constrained genetic algorithm which determines specific curing rates to each node composing the network, in order to minimize the total curing cost, while suppressing the epidemic. The method is compared with the exact semidefinite programming solver *SDPT*3 [22]. Experiments on both synthetic and real-world networks show that our approach finds solutions whose curing cost is lower than that obtained by the classical baseline method *SDPT*3.

The paper is organized as follows. In the next section we recall the concepts of epidemic spreading in the heterogeneous SIS model and the formalization of the problem as a constrained optimization problem. Section 3 describes the *GA* algorithm designed to solve the

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205508

problem, the representation and the variation operators. In Section 4 we test the method on several real-world and synthetic networks and compare the results with those obtained by the exact semidefinite solver *SDPT3*. Section 5 describes the most recent approaches to deal with epidemic spreading in networks. Finally, Section 6 concludes the paper and points out the future developments in this context.

2 BACKGROUND

The modeling of epidemic spreading on a network has long been studied and several models have been defined. Usually, it is assumed that each individual node can be in a disease stage such as *susceptible*, denoted by S, in which it can contract the infection, *infectious*, denoted by I, i.e. it contracted the infection, and *recovered*, denoted by R, meaning that the nodes recovered from the disease. The transition with an infection rate from the susceptible state to the infected state is due to the interactions with infected individuals.

The *Susceptible-Infected-Susceptible (SIS)* is one of the simplest compartmental models. It assumes that an individual can be infected more times, thus undergoing the cycle $S \rightarrow I \rightarrow S$, eventually forever. The *SIS* epidemic process on an undirected graph $G(V, E)$ with N nodes, is formalized as follows. The viral state of a node i is described by a Bernoulli random variable $X_i \in \{0, 1\} : X_i = 0$ for a healthy node and $X_i = 1$ for an infected node. At time t a node can thus be in the *infected state* with probability $v_i(t)$ =Pr$[X_i(t) = 1]$ or in the *healthy state* with probability $1 - v_i(t)$. In the homogeneous setting (Fig. 1), the *curing process* for a node i and the *infection rate* per link are independent Poisson processes with rates δ and β, respectively. In such setting, the *effective infection rate* is defined as $\tau = \beta/\delta$ and the problem is to compute $v_i(t)$ for each node in the graph. In the heterogeneous case (Fig. 2), instead, each node i recovers at rate δ_i, so that the curing rate is node-specific, and it can infect the other nodes j with different infection rates, denoted by β_{ij}.

For a network with N nodes, the SIS model can be expressed exactly in terms of a continuous Markov chain with 2^N states, which corresponds to all the combinations the nodes can be infected [24]. A remarkable property of the exact *SIS* Markov process is that it converges to the so called *absorbing state*, where all the nodes are healthy, that is $v_i(t) = 0$. However, the process shows the existence of a phase transition, that is there is a critical value τ_c, named *epidemic threshold*, such that if the effective infection rate τ is higher than τ_c, $\tau > \tau_c$, the infection becomes persistent, while if $\tau < \tau_c$, the virus dies out and the network is virus-free.

In real-world networks, where the number of nodes is high, the determination of the exact solution requires the resolution of a system of linear differential equations, whose number increases exponentially with N, thus, approximate models have been proposed [21, 24]. Among these approximate models, the *mean-field approximation model NIMFA*, replaces the original 2^N linear differential equations with N non-linear differential equations, representing the evolution of the infection probability of each node over time. In

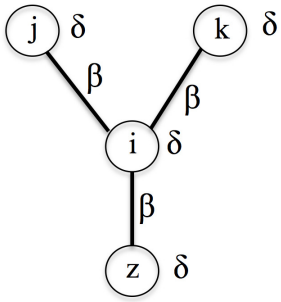

Figure 1: 4-nodes example of homogeneous SIS setting.

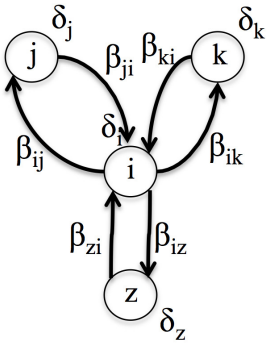

Figure 2: 4-nodes example of heterogeneous SIS setting.

the homogeneous setting, *NIMFA* determines the epidemic threshold of the spreading rate as the inverse of the spectral radius[1] $\lambda_{max}(A)$ of the adjacency matrix A associated with the graph G, i.e. $\tau_c^{(1)} = \frac{1}{\lambda_{max}(A)}$.

In the heterogeneous case [23], where each node can infect each of its neighbors with different infection rates (β_{ij}) and has its own curing rate δ_i, the *NIMFA* model describes the probability $v_i(t)$ for the node i of being infected as:

$$\frac{dv_i(t)}{dt} = \sum_{j=1}^{N} \beta_{ij} v_j(t) - \sum_{j=1}^{N} \beta_{ij} v_i(t) v_j(t) - \delta_i v_i(t). \quad (1)$$

These equations can be rewritten as

$$\frac{dV(t)}{dt} = \bar{A}V(t) + F(V) \quad (2)$$

where $V(t)$ is the vector $V(t) = (v_1(t), v_2(t), ..., v_N(t))$, $\bar{A}$ is defined as

$$\bar{A} = \begin{bmatrix} \delta_1 & \beta_{12} & . & . & . & \beta_{1N} \\ \beta_{21} & \delta_2 & & & & \\ . & & . & & & \\ . & & & . & & \\ . & & & & . & \\ \beta_{N1} & . & . & . & \beta_{NN-1} & \delta_N \end{bmatrix} \quad (3)$$

and $F(V)$ is a column vector having as i-th element

[1]The spectral radius of a square matrix $A \in C^{N \times N}$ is the largest absolute value of its (real or complex) eigenvalues $\lambda_1, \ldots, \lambda_N$, i.e. $\lambda_{max}(A) = max\{| \lambda_1 |, \ldots, | \lambda_N |\}$.

$$-\sum_{j=1}^{N}\beta_{ij}v_i(t)v_j(t) \tag{4}$$

In [14] the epidemic threshold for the heterogeneous setting is obtained by exploiting the results of Lajmanovich and Yorke [9] regarding the real component of the maximum eigenvalue $r(\bar{A})$ of $\bar{A}$:

$$r(\bar{A}) = max_{1\leq j\leq N} Re(\lambda_j(\bar{A})) \tag{5}$$

where $Re(\lambda_j(\bar{A}))$ is the real part of the eigenvalues of $\bar{A}$. The authors, by using the Theorem 3.1 of [9] stating that if $r(\bar{A}) \leq 0$, then the epidemic will go extinct (Theorem 2.1 in [14]), are able to identify the epidemic threshold.

Taking into account the *NIMFA* model, the problem of suppressing the spreading of a virus within a generic weighted network by a cost-optimal assignment of curing resources to nodes has been formalized in [14] in the following way. Each node i is cured with rate δ_i and the cost for allocating recovery resources (e.g. medicines, medical staff, etc.) to this node is c_i. The aim is to *minimize the total cost needed for curing the network* thus suppressing the viral infection. The total cost can be expressed as

$$U(\Delta) = \sum_{i=1}^{N} c_i\delta_i \tag{6}$$

where $\Delta = (\delta_1, \delta_2, ..., \delta_N)$ is the curing rate vector we want to find, knowing the cost c_i for each node.

For undirected weighted networks, $\beta_{ij} = \beta_{ji}$. As such, the adjacency matrix $A = (\beta_{ij})$ is symmetric and consequently the eigenvalues are real. For Theorem 2.1 in [14], if $\lambda_{max}(A - diag(\Delta)) \leq 0$ the infectious process dies out and all nodes are healthy. As such, the largest eigenvalue of $(A - diag(\Delta))$ identifies the epidemic threshold for the network considered. In order to find a cost-optimal allocation of curing resources, it is necessary to solve the following optimization problem.

Problem *Optimal Curing Policy OCP*. Given a graph $G = (V, E)$, the weighted adjacency matrix A with elements $a_{ij} = \beta_{ji}$ meaning that node i can infect node j with rate β_{ji}, and the cost coefficients $c_i > 0, i = 1, \ldots N$, find the curing rate vector $\Delta \geq 0$ which solves the nonlinear constrained problem:

$$\begin{array}{ll} \text{minimize} & U(\Delta) \\ \text{subject to} & \lambda_{max}(A - diag(\Delta)) \leq 0 \\ & \Delta \geq 0 \end{array}$$

This problem can be reformulated as the semidefinite programming problem (*SDP*) [25]:

$$\begin{array}{ll} \text{minimize} & U(\Delta) \\ \text{subject to} & diag(\Delta) - A \geq 0 \\ & \Delta \geq 0 \end{array}$$

since $diag(\Delta) \geq 0$ and the inequality sign in $diag(\Delta) - A \geq 0$, being $diag(\Delta) - A$ a matrix, means that it is semidefinite positive[2]. This implies that the *OCP* problem can be solved by using an *SDP* solver, like *SDPT3* [22].

[2]A semidefinite positive matrix $A \in R^{N\times N}$ is a symmetric matrix such that $x^T Ax \geq 0$ for all the $x \in R^N$. Equivalently, all the eigenvalues of A are nonnegative.

3 A CONSTRAINED GA SOLVER FOR THE OPTIMAL-COST CURING PROBLEM

In this section, we propose to find a solution of the *OCP* problem by using a constrained genetic algorithm, named *OCPGA*, that evolves a population of individuals by minimizing the total curing cost $U(\Delta)$ as fitness function. We adopt a real-coded representation in which an individual I of the population represents a vector $\Delta = (\delta_1, \delta_2, ..., \delta_N)$. Each variable δ_i is thus a curing rate which can assume a value in the interval $[x_i^l, x_i^u] = [0, 1]$. A value δ_i assigned to the i-th gene means that node i is cured with rate δ_i. The constraints to satisfy when searching for a solution, as can be seen from the *OCP* formulation, are the positiveness of the real part of the largest eigenvalue of the matrix $A - diag(\Delta)$, besides the lower and upper bound of each variable, i.e. $x_i^l \leq \delta_i \leq x_i^u$.

The method receives in input the matrix $A = (\beta_{ij})$ of the infection rates, the vector of curing costs $C = (c_1, c_2, ..., c_N)$, and performs the following steps:

- step 1: computes the diagonal matrix Δ;
- step 2: computes the real part of the largest eigenvalue λ_{max} of $A - diag(\Delta)$;
- step 3: runs for a fixed number of iterations by using $U(\Delta)$ as fitness function to minimize subject to $\lambda_{max} \leq 0$ and $\Delta \geq 0$, applying crossover and mutation operators;
- step 4: obtains the solution $\Delta^* = (\delta_1^*, \delta_2^*, ..., \delta_N^*)$ corresponding to the solution with the lowest fitness function.

Since the problem is constrained, once feasible parents are available, the variation operators should create feasible children. To this end we employed as mutation operator the *mutation feasible operator* (implemented in the Global Optimization toolbox of Matlab) which randomly generates directions and chooses a direction that satisfies bounds and linear constraints.

Crossover is an important operator for obtaining effective solutions to the problem. Classical one-point and two-point crossover operators, as pointed out by Deb [4], could not be able to create feasible children. Instead, when feasible solutions are available, a crossover operator that controls the spread of children around the parents is more suitable. To this end, Deb [4] proposed the *simulated binary crossover (SBX)*, which controls the spread using a distribution index η_c and it is able to explore contiguous regions if the diversity among parents is sufficient. The *SBX* operator computes the children solutions $y^{(1)}$ and $y^{(2)}$ from the two feasible parents $x^{(1)}$ and $x^{(2)}$ as follows:

$$y^{(1)} = 0.5\left[(x^{(1)} + x^{(2)} - \bar{\beta}[x^{(2)} + x^{(1)}]\right] \tag{7}$$

$$y^{(2)} = 0.5\left[(x^{(1)} + x^{(2)} + \bar{\beta}[x^{(2)} + x^{(1)}]\right] \tag{8}$$

where

$$\bar{\beta} = \begin{cases} (\alpha u)^{1/(\eta_c+1)} & \text{if } u \leq 1/\alpha \\ (\frac{1}{2-\alpha u})^{1/(\eta_c+1)} & \text{otherwise} \end{cases} \tag{9}$$

with $\alpha = 2 - \beta^{-(\eta_c+1)}$, u is a random number between 0 and 1, η_c is the so-called distribution index of *SBX*, taking only nonnegative values, such that a small value allows children solutions far from parents, while a large value generates solutions near to the parents,

and

$$\beta = 1 + \frac{2}{y^{(2)} - y^{(1)}} min[(x^{(1)} - x^l), (x^u - x^{(2)})]$$

Deb showed that this computation generates children solutions outside the fixed range $[x^l, x^u]$ with zero probability.

4 EXPERIMENTAL EVALUATION

In this section we test the *OCPGA* algorithm on both real-world and synthetic networks by comparing the fitness values it obtains with those found by the *SDPT*3 method. *OCPGA* has been written in Matlab, version 2015b, by using the Genetic Algorithm solver implemented in the Global Optimization Toolbox. To run *SDPT*3 over Matlab, we used the *CVX* package for specifying and solving convex programs [7]. For all the experiments we set crossover fraction 0.9, mutation rate 0.2, population size 1000, number of generations 1000, and, as suggested in [4], $\eta_c = 1$. These parameter values have been obtained by employing a trial-and-error procedure on the benchmark data sets. It is worth pointing out that for the majority of the networks, the method stopped with a much lower number of generations. Each experiment has been executed 10 times and the average values of the fitness function, along with the standard deviation have been reported.

In the following, we describe the real and the synthetic networks used, and the results obtained by the experiments.

4.1 Datasets

We analyze both real-world and synthetic networks. Table 1 summarizes the topological features of these networks. As real-world networks, we selected 5 Internet backbones from the Internet Topology Zoo[3], a repository containing backbone topologies coming from all over the world and made public from several network operators. Internet backbones are networks interconnecting different kinds of networks designed to transfer network traffic at high speeds. The selected topologies, available in .gml format, have been converted in adjacency matrices using a Python script. Internet backbones are often subject to attacks which can severely affect a subset of hosts or even entire networks composing the Internet. A compromised Border Gateway Protocol (BGP) router could cause, for example, blackholing, that is incoming or outgoing traffic is discarded without informing the source that the data did not reach its intended recipient, or redirect the Internet traffic to destinations that are not the intended destinations, thus provoking instability.

Figs. 3-7 show the topologies and the physical location of the networks chosen from the repository. Together with the Internet topologies, we selected another class of networks often subject to attacks: Facebook friend lists networks[4]. These social networks are composed by several ego networks, where the ego is a social profile, linked to other egos, also known as Facebook friends. This kind of networks are subject to the spreading of fake news from an hacked social profile to its friends.

As synthetic networks, we considered the three popular classes of random networks with $N = 128$ nodes, the Erdős-Rényi (*ER*) graphs [5], the Watts-Strogatz (*WS*) [27] graphs, and the Bárabasi-Albert (*BA*) [1] networks. The Erdős-Rényi networks can reasonably

[3]http://www.topology-zoo.org/
[4]https://snap.stanford.edu/data/egonets-Facebook.html

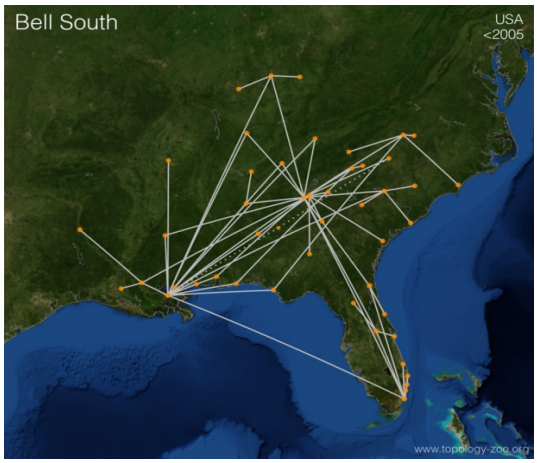

Figure 3: Topology of the Internet Backbone Bell South.

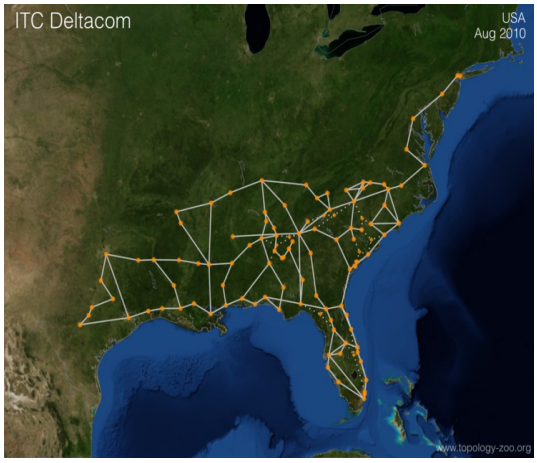

Figure 4: Topology of the Internet Backbone ITC Deltacom.

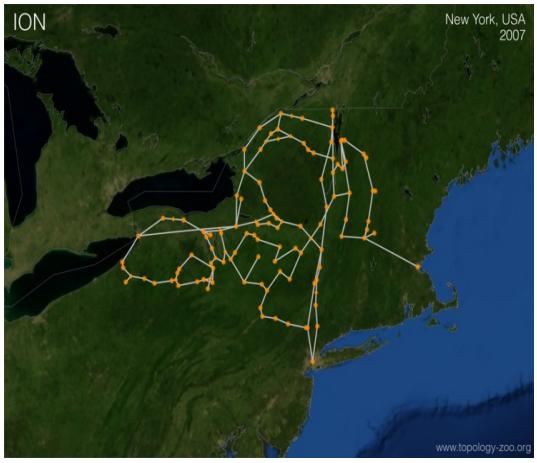

Figure 5: Topology of the Internet Backbone ION.

Figure 6: Topology of the Internet Backbone OTEGlobe.

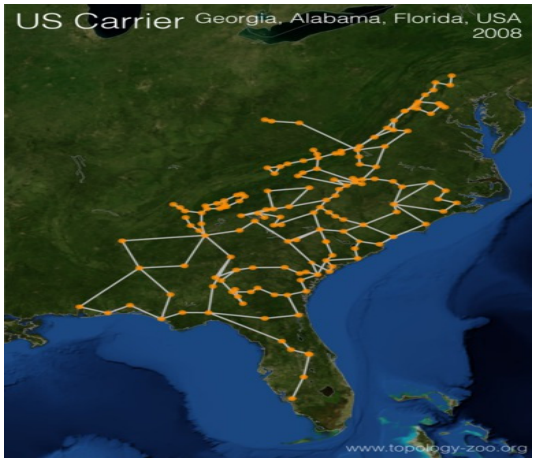

Figure 7: Topology of the Internet Backbone US Carrier.

model peer-to-peer networks and ad-hoc networks. An Erdős-Rényi graph can be generated from a set of N nodes connecting two nodes with probability p_c. In our simulations, we set $p_c = \ln(N)/N$ and checked if each resulting graph was connected. As second class, we generated Watts-Strogatz small-world networks [27], whose main features are a high clustering coefficient and have nodes which can be easily reached in few hops by the other nodes. Examples of networks which can be modeled through the Watts-Strogatz class are mobile contact networks, as Bluetooth or Wi-Fi device-to-device networks. A Watts-Strogatz graph is generated from a ring lattice of N nodes, where each node is connected to k nodes, by rewiring each edge with probability p. Here, we set $k = 6$ and $p = 0.5$. We finally generated Bárabasi-Albert [1] scale-free networks, where the *preferential attachment* feature, for which low degree nodes tend to attach to high degree nodes, well model real-world complex networks like the Internet, the World Wide Web and social networks. A Bárabasi-Albert graph can be generated from an initial connected graph of m_0 nodes. At each time step, a new node is connected to $m \leq m_0$ existing nodes with a probability that is proportional to the degree of the existing node. In our simulations, we fixed $m_0 = 5$ and $m = 2$.

4.2 Results

Real-world networks. Table 2 shows the values of the objective function obtained by the execution of *SDPT*3 and *OCPGA* over the real-world networks. We executed the methods 10 times and reported the average values of the objective function. We do not show the standard deviation because very low.

We considered the case in which the costs are equal for all the nodes (i.e., $C = (c_1, c_2, ..., c_N) = (1, 1, ..., 1)$) and the case in which the vector of curing costs contains random values. For setting the infection spreading rates of each node, we randomly generated rates in the range [0, 1] and then multiplied them for 10^{-3}. Regarding the Internet Backbones, *OCPGA* is always able to find solutions better than those obtained by *SDPT*3, both with equal curing costs and random costs. Observe that random costs, since curing rates are lower than or equal to 1, produce lower values of the objective function $\Delta(U)$. The better performance of *OCPGA* with respect to *SDPT*3 is more noticeable when equal curing costs are fixed. For instance, on the *OTEGlobe* network, the total curing cost diminishes from 0.12 to 0.026, while for random costs, *OCPGA* obtains 0.014 and *SDPT*3 0.037. For the Facebook networks, we found similar results. *OCPGA* outperforms *SDPT*3 obtaining a fitness value of 0.056 instead of 0.162, when unitary costs are considered, and 0.045 instead of 0.062 with random costs, on the Ego 3980 network. On the Ego 686, though the values obtained by *OCPGA* are lower than those obtained by *SDPT*3, the difference is less marked.

Synthetic networks. In Table 3, we show the results obtained for the Erdős-Rényi, Watts-Strogatz, and Bárabasi-Albert random networks. For these experiments, we used the same strategy adopted in real-world networks for setting the spreading infection rates and the curing costs. *OCPGA* outperforms SDPT3 in all the scenarios, except for the *ER* networks when the random costs are considered. The reduction of $\Delta(U)_{OCPGA}$ for these synthetic networks is less strong, probably due to the network topology. In fact, these networks have a higher number k of neighbors, as can be seen from Table 1, thus obtaining low curing costs is more difficult because the infection can be propagated more easily. In any case, *OCPGA* finds curing solutions with low good values.

5 RELATED WORK

The study and analysis of epidemic spreading in networks has been investigated by several researchers coming from different fields such as epidemiology, mathematics, physics, computer science. The models developed by these scientists are similar and fundamental for the understanding of infectious diseases. Extensive surveys describing the main results on disease modeling can be found in [13, 16]. In the following we report some of the most recent proposals, by focusing more on the evolutionary computation based approaches.

Newman [12] studied the spread of epidemic disease on random graphs, while Pastor-Satorras and Vespignani [17] on scale-free networks. Wang et al. [26] analyzed the spreading mechanism in contact networks. Borgs et al. [2] studied how to control epidemics on social and technological networks by distributing antidote to nodes. Gourdin et al. [6] investigated the minimization of a curing

Table 1: Topological features of the analyzed networks: number of nodes (N), average degree (<k>), average clustering coefficient (<C>) and density (D). For the synthetic networks, the topological measures have been averaged over 10 network realizations.

Network type	Network name	N	<k>	<C>	D
Backbone	Bell South	51	1.294	0.081	0.052
	OTE Globe	93	1.108	0.011	0.024
	ITC Deltacom	113	1.425	0.053	0.025
	ION	125	1.168	0.006	0.019
	US Carrier	158	1.196	0.002	0.015
Facebook	Ego 3980	52	5.625	0.462	0.11
	Ego 686	168	19.714	0.534	0.118
Synthetic	Erdős-Rényi	128	5.23	0.054	0.041
	Watts-Strogatz	128	6	0.109	0.047
	Bárabasi-Albert	128	3.954	0.132	0.031

Table 2: Comparison between SDPT3 and *OCPGA* values of objective function over real-world networks for equal costs (e) and random costs (r).

Network	$\Delta(U)^e_{SDPT3}$	$\Delta(U)^e_{OCPGA}$	$\Delta(U)^r_{SDPT3}$	$\Delta(U)^r_{OCPGA}$
Bell South	0.075	0.008	0.033	0.007
OTEGlobe	0.12	0.026	0.037	0.014
ITC Deltacom	0.196	0.047	0.067	0.038
ION	0.148	0.034	0.06	0.03
US Carrier	0.182	0.042	0.079	0.032
Ego 3980	0.162	0.056	0.062	0.045
Ego 686	1.69	1.342	0.773	0.645

Table 3: Comparison between SDPT3 and *OCPGA* values of objective function over Erdős-Rényi (ER), Watts-Strogatz (WS) and Bárabasi-Albert (BA) networks with 128 nodes. Each result has been averaged over 10 graph realizations.

Network type	$\Delta(U)^e_{SDPT3}$	$\Delta(U)^e_{OCPGA}$	$\Delta(U)^r_{SDPT3}$	$\Delta(U)^r_{OCPGA}$
ER	0.639 (0.027)	0.358 (0.044)	0.257 (0.015)	0.2910 (0.0455)
WS	0.394 (0.006)	0.1864 (0.014)	0.165 (0.007)	0.144 (0.009)
BA	0.266 (0.006)	0.1301 (0.01)	0.109 (0.003)	0.09 (0.007)

budget, given the level of network infection. Prakash et al. [18] dealt with the problem of distributing a fixed amount of resources to network nodes in order to minimize the infection rate. Preciado et al. [19, 20] proposed optimal resource allocation in spreading processes by assuming to modify the infection rates of individuals. More recently, Ottaviano et al. studied optimal curing policy over a community network [14]. Zhai et al. [28] analyzed several epidemic evolution models and proposed a framework for controlling the epidemic spread in terms of optimization of the rate of the epidemic evolution.

Regarding the use of Genetic Algorithms and evolutionary based approaches, there are a few number of proposals.

Parousis-Orthodoxou and Vlachos [15] for example, exploited the capabilities of a genetic algorithm to obtain an optimal vaccination scheme over a network where the epidemic spread is modeled through a SIR (Susceptible-Infected-Recovered) process. Here, the function to optimize is the number of vaccines that would cause a minimal percentage of infected nodes, considering that each vaccine has a cost and there is also a cost for treating an infected node. Differently from this scheme, our algorithm focuses on a different epidemic model, i.e. the SIS. Moreover, *OCPGA* aims to find the optimal curing strategy that makes the epidemic to disappear. In addition, our algorithm considers an heterogeneous setting where the infection and the curing rates can be different from node to node, and different genetic operators.

In another work, Concatto et al. [3] used a genetic algorithm to minimize the infection spreading over a network by removing a specified number of connections. The problem analyzed is the Min-SEIS-Cluster where a node can be in a further state, the exposed (E) state, and the organization in cluster of nodes is also taken into account. Basically, nodes of the same cluster can infect each other with a higher intensity. Similarly to the work of Parousis-Orthodoxou and Vlachos, this algorithm minimizes the number

of infected nodes, while our algorithm focuses on achieving the healty-state with an optimal curing policy.

Lahiri and Cebrian [8] focused on information diffusion processes on social networks by proposing a *genetic algorithm diffusion model* (GADM). A canonical genetic algorithm with binary string chromosomes and one-point crossover is used to model different diffusion processes, including the scheme optimizing the amount of information a node has. Each node possesses a 'unit' of information generated through *hyperplane-defined functions* (HDF) which is spread through the network.

Liao et al. [10] proposed a ripple-spreading network model in the study of infectious disease transmission between individuals moving and contacting each other randomly. Similarly to the ripple-spreading phenomenon on liquid surfaces, the infection probability is reflected by the energy point of ripples. As the ripple spreads out, the energy point decays, thus indicating that long distance and few contacts have a low probability of infection. Since the proposed model has many parameters, a genetic algorithm is used here to tune them.

6 CONCLUSION

The paper proposed a constrained genetic algorithm to solve the problem of finding an optimal curing policy in a network where a virus spreads following the Susceptible-Infected-Susceptible (SIS) epidemic model. Exploiting the N-Intertwined Mean-Field Approximation (NIMFA) of the SIS spreading process, the method finds curing rates, different for each node, with minimum cost. Moreover, because of the constraint satisfaction, the solution guarantees that the spreading process will always converge towards the absorbing state in which the probability of a node to be infected is zero, thus that the epidemic will extinct. Experiments on both synthetic and real-world networks showed that the approach finds solutions whose curing cost is lower than that obtained by the *SDPT3* method. Several aspects, however, still need to be investigated, including the effect of different policies for population initialization, a more specialized mutation operator, the evaluation of the approach to other real-world networks, the extension of the method to networks with community structure. Future work will concentrate on these aspects.

REFERENCES

[1] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 1 (2002), 47–97.

[2] Christian Borgs, Jennifer Chayes, Ayalvadi Ganesh, and Amin Saberi. 2010. How to Distribute Antidote to Control Epidemics. *Random Struct. Algorithms* 37, 2 (Sept. 2010), 204–222.

[3] Fernando Concatto, Wellington Zunino, Luigi A Giancoli, Rafael Santiago, and Luís C Lamb. 2017. Genetic algorithm for epidemic mitigation by removing relationships. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 761–768.

[4] Kalyanmoy Deb. 2000. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering* 186 (2000), 311–338.

[5] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5 (1960), 17–61.

[6] Eric Gourdin, Jasmina Omic, and Piet Van Mieghem. 2011. Optimization of network protection against virus spread. In *Proceedings of the 8th International Workshop on Design of Reliable Communication Networks (DRCN), 2011*. 659–667.

[7] Michael Grant, Stephen Boyd, and Yinyu Ye. 2008. CVX: Matlab software for disciplined convex programming. (2008).

[8] Mayank Lahiri and Manuel Cebrian. 2010. The Genetic Algorithm as a General Diffusion Model for Social Networks.. In *AAAI*.

[9] Ana Lajmanovich and James A. Yorke. 1976. A deterministic model for gonorrhea in a nonhomogeneous population. *Mathematical Biosciences* 28, 3 (1976), 221 – 236.

[10] Jian-Qin Liao, Xiao-Bing Hu, Ming Wang, and Mark S Leeson. 2013. Epidemic modelling by ripple-spreading network and genetic algorithm. *Mathematical Problems in Engineering* 2013 (2013).

[11] A.G. McKendrick. 1926. Applications of mathematics to medical problems. In *Proceedings of Edin. Math Society*, Vol. 14. 98–130.

[12] M.E.J. Newman. 2002. Spread of epidemic disease on Networks. *Physical Review E* 66, 1 (2002), 016128.

[13] Cameron Nowzari, Victor M. Preciado, and George J. Pappas. 2016. Analysis and Control of Epidemics: A Survey of Spreading Processes on Complex Networks. *IEEE Control Systems* 36, 1 (2016), 26–46.

[14] Stefania Ottaviano, Francesco De Pellegrini, Stefano Bonaccorsi, and Piet Van Mieghem. 2017. Optimal curing policy for epidemic spreading over a community network with heterogeneous population. *Information and Inference: A Journal of the IMA* doi:10.1093/imaiai/drn000 (2017).

[15] KJ Parousis-Orthodoxou and DS Vlachos. 2014. Evolutionary Algorithm for Optimal Vaccination Scheme. In *Journal of Physics: Conference Series*, Vol. 490. IOP Publishing, 012027.

[16] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. 2015. Epidemic Processes in Complex Networks. *Review of Modern Physics* 87, 3 (2015), 925–979.

[17] Romualdo Pastor-Satorras and Alessandro Vespignani. 2014. Epidemic Spreading in Scale-Free Networks. *Physical Review Letters* 86, 14 (2014), 99–108.

[18] B. Aditya Prakash, Lada Adamic, Theodore Iwashyna, Hanghang Tong, and Christos Faloutsos. 2013. Fractional Immunization in Networks. In *Proceedings of the SIAM Data Mining Conference*. 659–667.

[19] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. 2013. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013, December 10-13, 2013, Firenze, Italy*. 7486–7491.

[20] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. 2014. Optimal Resource Allocation for Network Protection Against Spreading Processes. *IEEE Trans. Control of Network Systems* 1, 1 (2014), 99–108.

[21] Faryad Darabi Sahneh, Caterina Scoglio, and Piet Van Mieghem. 2013. Generalized Epidemic Mean-field Model for Spreading Processes over Multilayer Complex Networks. *IEEE/ACM Trans. Netw.* 21, 5 (Oct. 2013), 1609–1620.

[22] R. H. Tütüncü, K. C. Toh, and M. J. Todd. 2003. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming* 95, 2 (01 Feb 2003), 189–217.

[23] Piet Van Mieghem and Jasmina Omic. [n. d.]. In-Homogeneous Virus Spread in Networks. *arxiv:1306.2588* ([n. d.]).

[24] Piet Van Mieghem, Jasmina Omic, and Robert Kooij. 2009. Virus Spread in Networks. *IEEE/ACM Trans. Netw.* 17, 1 (Feb. 2009), 1–14.

[25] Lieven Vandenberghe and Stephen Boyd. 1996. Semidefinite Programming. *SIAM Rev.* 38, 1 (March 1996), 49–95.

[26] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. 2003. Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint. In *In Proc. International Symposium Rel. Distr. Syst. (SRDS)*. 25–34.

[27] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440.

[28] Xiangping Zhai, Liang Zheng, Jianping Wang, and Chee Wei Tan. 2013. Optimization algorithms for epidemic evolution in broadcast networks. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 1540–1545.