IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 7, 2020, accepted January 21, 2020, date of publication January 27, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969430

A Cost-Effective Algorithm for Selecting Optimal Bandwidth to Clear Malicious Codes

JICHAO BI[1], (Student Member, IEEE), XIAOFAN YANG[1], (Member, IEEE), WANPING LIU[2], (Member, IEEE), AND DA-WEN HUANG[1]

[1]School of Big Data and Software Engineering, Chongqing University, Chongqing, 400044, China
[2]College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

Corresponding author: Wanping Liu (lwphe@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61603065, in part by the Chongqing Basic Science and Advanced Technology Research Program under Grant cstc2017jcyjAX0277, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJQN201901101.

ABSTRACT Malicious code has posed a severe threat to modern society. Delivering antivirus program to networks is an important task of a cybersecurity company. As the bandwidth resource in a company is limited and precious, cybersecurity companies have to make a tradeoff between the impact(i.e. the economic loss) of malicious codes and the bandwidth assigned to transmit the antivirus programs. This paper addresses the malicious code and bandwidth tradeoff(MCBT) problem. By developing a novel malicious code and antivirus program interacting model, the total loss, which is the sum of the bandwidth usage fee and the economic loss, is quantified. On this basis, the MCBT problem is modelled as a constrained optimization problem that we refer to as the MCBT model, where the independent variable stands for bandwidth, and the objective function stands for the total loss. Some optimal bandwidth is determined by solving the MCBT model. Based on this, we propose a heuristic algorithm named DOWNHILL, which outperforms random strategies. Finally, the influence of some factors on the optimal bandwidth and the corresponding optimal total loss is uncovered through numerical simulations. To our knowledge, this is the first time the MCBT problem is treated in this way.

INDEX TERMS Cyber security, malicious code, bandwidth, node-level epidemic model, constrained optimization, tradeoff problem.

I. INTRODUCTION

The normal operation of modern society relies largely on computer networks. On a daily basis, people acquire information and knowledge through the Web, communicate with each other through online social networks, and buy goods through electronic payment [1]. Meanwhile, computer networks provide a shortcut for the spread of malicious codes, causing huge economic loss. For instance, a notorious ransomware named Wanna Decryptor has recently swept across the globe, leading to massive computer paralysis [2], [3]. Therefore, controlling the negative impact and potential consequence of malicious codes has long been a hotspot of research in the field of cyber security [4].

Cybersecurity companies, whose main responsibility is to secure the cyberspace, can detect new malicious codes by analyzing data and reports collected from network users. After malicious codes are detected, cybersecurity companies have to develop the corresponding antivirus programs, which are a special kind of computer programs used to prevent and remove malicious codes [4], [5]. Bandwidth is the amount of data that can be transmitted in a fixed amount of time, usually measured by Mbps. No double, allocating much bandwidth to transmit antivirus programs could effectively increase the repair rate against malicious codes, minimizing the impact(i.e. the economic loss) of malicious codes. However, as the bandwidth resource in a cybersecurity company is precious and limited, occupying too much bandwidth for antivirus programs leads to higher bandwidth usage fee and increases the risk of blocking or reducing normal communications, which could inflict inestimable loss. Thus,

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan Bu.

cybersecurity companies have to make a tradeoff between the economic loss and the bandwidth assigned for antivirus programs. We refer to this problem as the malicious code and bandwidth tradeoff (MCBT) problem.

To solve the MCBT problem, we have to assess the economic loss first. As the economic loss is closely related to the malicious code and antivirus program interacting process, we have to formulate the malicious code and antivirus program interacting process. Unlike medical measures that vaccination could only be injected one by one [6], antivirus programs can be disseminated very rapidly through computer networks [7], [8]. Therefore, a malicious code and antivirus program interacting model has to take into account the propagation of both malicious codes and antivirus programs simultaneously. We refer to a malicious code and antivirus program interacting model with this feature as a Susceptible-Infected-Patched-Susceptible (SIPS) model. The earliest SIPS models are compartmental, in which each compartment consists of all the nodes(i.e. devices) with a same state, and the evolution of the expected fraction of each compartment is characterized by a separate differental equation [9]–[11]. The compartmental SIPS models are especially suited to homogeneous networks. However, empirical studies show that many realistic networks, including the router-level Internet, the domain-level Internet and the World-Wide-Web, are scale-free rather than homogeneous [12]. As thus, some malicious code propagation models based on scale-free networks are suggested [13]–[15].

With the progress of wireless and mobile communication technologies, many existing computer networks are neither homogeneous nor scale-free. Rather, they may admit an arbitrary topology [16]–[18]. Individual-level propagation models, in which the probability of each node being in each state is characterized by a separate differential equation, are well suited to the study of various propagation phenomena on arbitrary networks [19]–[24].

To disseminate an antivirus program on a large-sized computer network, we have to specify a set of nodes into which the antivirus program is injected. Due to the limitation in network bandwidth, this set is small as compared with the size of the network [25]. However, all the previous SIPS models ignore the antivirus program injection subset, limiting their applicability.

The main contributions of this paper are as follows:

- First, we propose a novel malicious code and antivirus program interacting model, in which the antivirus program injection subset is specified. On this basis, we quantify the economic loss caused by malicious codes as the expected loss of all network users.
- Second, we model the MCBT problem as a minimization problem (which we refer to as the MCBT model), where the independent variable stands for the bandwidth, and the objective function stands for the total loss that is the sum of the bandwidth usage fee and the economic loss.
- Third, we propose DOWNHILL algorithm to address the MCBT problem. Simulation results prove that DOWNHILL algorithm performs much better than random strategies.
- Finally, we examine the influence of some factors on the optimal bandwidth and the corresponding optimal total loss through numerical simulations. This potentially provides cybersecurity companies with knowledge to quickly take measures against malicious codes.

The subsequent materials are organized in this fashion: Section 2 models the MCBT problem. Section 3 solves the MCBT model and proposes a heuristic algorithm(i.e. DOWNHILL). The performance of DOWNHILL algorithm is evaluated in Section 4, and Section 5 examines the influence of different factors on the optimal bandwidth and the corresponding total loss. Finally, conclusion is given in Section 6 and Section 7 closes this work.

II. MALICIOUS CODE AND ANTIVIRUS PROGRAM TRADEOFF PROBLEM

This section is dedicated to the modeling of the following problem:

Malicious Code and Antivirus Program Tradeoff (MCBT) problem: Given a computer network and an antivirus program injection subset, determine a bandwidth assigned to transmit antivirus programs to minimize the total loss.

First, we introduce a set of terminologies and notations. Second, we describe a malicious code and antivirus program interacting model. Finally, we present a model for the MCBT problem.

A. TERMINOLOGIES AND NOTATIONS

Consider a computer network $G = (V, E)$, where G represents the topology of a computer network, V denotes the node set, and E is the set of edges connecting the nodes. Let $\mathbf{A} = \left[a_{ij}\right]_{N\times N}$ denote the adjacency matrix for G. Thus, $a_{ij} = 1$ or 0 if and only if $\{i, j\} \in E$ or not.

Suppose a malicious code targeting a computer network can be injected into any node of the network and can propagate through the network. Suppose its corresponding antivirus program can only be injected into a small subset of nodes of the network, denoted as U, and can propagate through the network. Assume each and every node in the network is in one of three possible states: *susceptible*, *infected*, and *patched*. Susceptible nodes are those that are not infected with the malicious code but have not received the corresponding antivirus program. So, susceptible nodes are vulnerable to this malicious code. Infected nodes are those that are infected with the malicious code. Patched nodes are those that are not infected with the malicious code and have received the corresponding antivirus program. So, patched nodes possess immunity to this malicious code.

For $t \geq 0$, let $X_i(t) = 0$, 1, and 2 denote the event that node i is susceptible, infected, and patched at time t, respectively. Then the vector

$$\mathbf{X}(t) = [X_1(t), \ldots, X_N(t)] \tag{1}$$

stands for the state of the network at time t. Let $S_i(t)$, $I_i(t)$, and $P_i(t)$ denote the probability of the event that node i is susceptible, infected, and patched at time t, respectively.

$$\begin{aligned} S_i(t) &= \Pr\{X_i(t) = 0\}, \\ I_i(t) &= \Pr\{X_i(t) = 1\}, \\ P_i(t) &= \Pr\{X_i(t) = 2\}. \end{aligned} \tag{2}$$

As $S_i(t) + I_i(t) + P_i(t) \equiv 1$, $t \geq 0$, $1 \leq i \leq N$, the vector

$$\mathbf{E}(t) = [I_1(t), \ldots, I_N(t), P_1(t), \ldots, P_N(t)] \tag{3}$$

stands for the expected state of the network at time t.

Remark 1: In practice, $\mathbf{E}_0$ *may be estimated through the relevant user reports.*

B. MALICIOUS CODE AND ANTIVIRUS-PROGRAM INTERACTING MODEL

To model the malicious code and antivirus program interactive process, let us introduce a set of rational hypotheses as follows.

(H$_1$) At any time t, every susceptible node $i \in V$ is injected with a malicious code at the average rate $\beta_I > 0$. We refer to the rate as *malicious code injection rate*.

(H$_2$) At any time t, every susceptible node $i \in V$ is infected by a neighboring infected node at the rate $\beta_P \sum_{i=1}^{N} a_{ij} I_j(t)$, where $\beta_P > 0$ is a constant. We refer to β_P as *malicious code propagation rate*.

(H$_3$) At any time t, every node $i \in U$ is injected with the antivirus program at the average rate $\gamma_I > 0$. We refer to the rate as *antivirus program injection rate*.

(H$_4$) At any time t, every node $i \in V$ gets the antivirus program by a neighboring patched node at the rate $\gamma_P \sum_{i=1}^{N} a_{ij} P_j(t)$, where $\gamma_P > 0$ is a constant. We refer to γ_P as *antivirus program propagation rate*.

(H$_5$) Due to the reinstall of computer operation system, at any time every patched node becomes susceptible at the average rate $\delta > 0$. We refer to the rate as *antivirus program failure rate*.

Remark 2: The rate at which a specific event occurs is a basic notation in stochastic process theory [26]. As a matter of fact, the reciprocal of rate stands for the mean time that elapses before the event occurs. Thus, the reciprocal of malicious code injection rate β_I *at a given time stands for the mean time that elapses from this time to the time this node is infected with malicious code through injection; the reciprocal of malicious code propagation rate* β_P *at a given time stands for the mean time that elapses from this time to the time this node is infected with malicious code through propagation; the reciprocal of antivirus program injection rate* γ_I *at a given time stands for the mean time that elapses from this time to the time this node gets antivirus program through injection; the reciprocal of antivirus program propagation rate* γ_P *at a given time stands for the mean time that elapses from this time to the time this node gets antivirus program through propagation; the reciprocal of antivirus program failure rate* δ *at a given time stands for the mean time that elapses from this time to the time the antivitus of this node fails.*

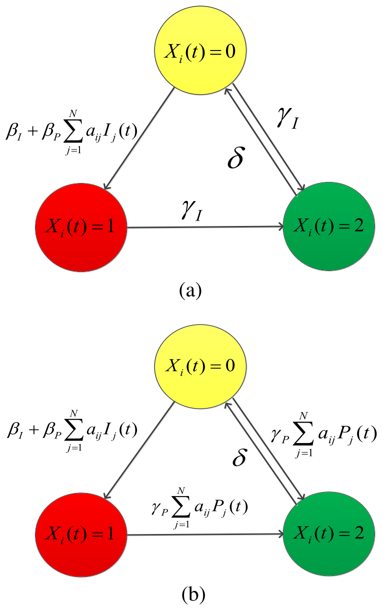
$X_i(t)=0$
$\beta_I + \beta_P \sum_{j=1}^{N} a_{ij} I_j(t)$
γ_I
δ
$X_i(t)=1$
γ_I
$X_i(t)=2$
(a)
$X_i(t)=0$
$\beta_I + \beta_P \sum_{j=1}^{N} a_{ij} I_j(t)$
$\gamma_P \sum_{j=1}^{N} a_{ij} P_j(t)$
δ
$X_i(t)=1$
$\gamma_P \sum_{j=1}^{N} a_{ij} P_j(t)$
$X_i(t)=2$
(b)

FIGURE 1. Diagram of the hypotheses (H$_1$)–(H$_5$), where (a) $i \in U$, (b) $i \in V - U$.

$$\begin{cases} \dfrac{dI_i(t)}{dt} = \left[\beta_I + \beta_P \sum_{j=1}^{N} a_{ij} I_j(t)\right] [1 - I_i(t) - P_i(t)] \\ \qquad - \gamma_I I_i(t), \quad t \geq 0, i \in U, \\ \dfrac{dP_i(t)}{dt} = \gamma_I [1 - P_i(t)] - \delta P_i(t), \quad t \geq 0, i \in U, \\ \dfrac{dI_i(t)}{dt} = \left[\beta_I + \beta_P \sum_{j=1}^{N} a_{ij} I_j(t)\right] [1 - I_i(t) - P_i(t)] \\ \qquad - \gamma_P I_i(t) \sum_{j=1}^{N} a_{ij} P_j(t), \quad t \geq 0, i \in V - U, \\ \dfrac{dP_i(t)}{dt} = \gamma_P [1 - P_i(t)] \sum_{j=1}^{N} a_{ij} P_j(t) - \delta P_i(t), \\ \qquad t \geq 0, i \in V - U. \end{cases} \tag{4}$$

where $E(0) = E_0$, $t \geq 0$, $1 \leq i \leq N$.

This system characterizes the expected interactive process of malicious codes and the corresponding antivirus programs. We refer to the system as *malicious code and antivirus program interacting (MCBI) model*.

C. MODELING OF MCBT PROBLEM

An MCBI model involves seven factors: network topology $G = (V, E)$, antivirus program injection subset U, malicious code injection rate β_I, malicious code propagation rate β_P, antivirus program injection rate γ_I, antivirus program propagation rate γ_P, and antivirus program failure rate δ. Among these factors, antivirus program injection rate γ_I is controllable by the cybersecurity company that provides cyber security service to the network. Specifically, the antivirus program injection rate is determined by the bandwidth allocated to transmit the antivirus program. The larger the bandwidth, the higher the antivirus program injection rate. We denote the bandwidth for the antivirus program per unit time by x. Now, let us introduce the following reasonable hypotheses.

(H_6) The bandwidth x is bounded from above by $x_{\max}$, where $x_{\max} > 0$ is a constant. We refer to $x_{\max}$ as *maximum bandwidth.*

(H_7) $M = c_1\, x$, in which M is the bandwidth usage fee per unit time for every node in U and $c_1 > 0$ is a constant. We refer to c_1 as *cost coefficient*.

(H_7) $\gamma_I = c_2\, x^\theta$, where $c_2 > 0$ is a constant. We refer to c_2 as *antivirus program injection coefficient*.

Given a time horizon $[0, T]$. To quantify the impact of a malicious code in the time horizon, let us introduce the following reasonable hypothesis.

(H_8) For $1 \le i \le N$, the economic loss per unit time of the infected node i is one unit.

Thus, the bandwidth usage fee in the time horizon $[0, T]$ is

$$P(x) = T\sum_{i=1}^{N_u} M = Tc_1\sum_{i=1}^{N_u} x = Tc_1\sum_{i=1}^{N_u}\sqrt[\theta]{\frac{\gamma_I}{c_2}}. \tag{5}$$

Then the economic loss caused by a malicious code, which is measured by the expected loss of all network users in the time horizon $[0, T]$, is

$$L(x) = \int_0^T \sum_{i=1}^{N} I_i(t)dt. \tag{6}$$

Hence, the total loss is

$$C(x) = P(x) + L(x) = T\sum_{i=1}^{N_u} M + \int_0^T \sum_{i=1}^{N} I_i(t)dt. \tag{7}$$

Based on above discussions, the MCBT problem is modelled as the following constrained minimization problem:

$$\begin{aligned} \min_{0\le x\le x_{\max}} \quad & C(x) = T\sum_{i=1}^{N_u} M + \int_0^T \sum_{i=1}^{N} I_i(t)dt, \\ s.t. \quad & \text{VAI model (4) with } \mathbf{E}(0) = \mathbf{E}_0. \end{aligned} \tag{8}$$

We refer to the problem as *MCBT model.* An MCBT model is determined by the following 13-tuple.

$$(G, U, \beta_I, \beta_P, \gamma_P, \delta, c_1, c_2, x_{\max}, T, initial, step, \mathbf{E}_0). \tag{9}$$

in which *initial* stands for the initial value of x, and *step* is the iteration step.

III. SOLUTION FOR MCBT MODEL

In the previous section, the MCBT problem is modelled as the MCBT model. Due to the inherent complexity of the objective function in the MCBT model, it seems impossible to solve the model analytically. So, we turn our attention to the numerical solutions for the MCBT model.

A. NETWORKS

First, we provide three representative networks for the following experiments.

Small-world networks are those that have a small diameter and a high clustering coefficient [27]. Empirical studies show that many real-world networks are small-world [12]. Therefore, small-world networks have been taken as a model of real-world networks. Pajek is one of the most famous pieces of software for performing social network analysis [28]. By using Pajek, setting the number of nodes as $N = 50$, we get a synthetic small-world network G_{SW}. Fig. 2(a) shows G_{SW}, where U_{SW} consists of the red nodes.

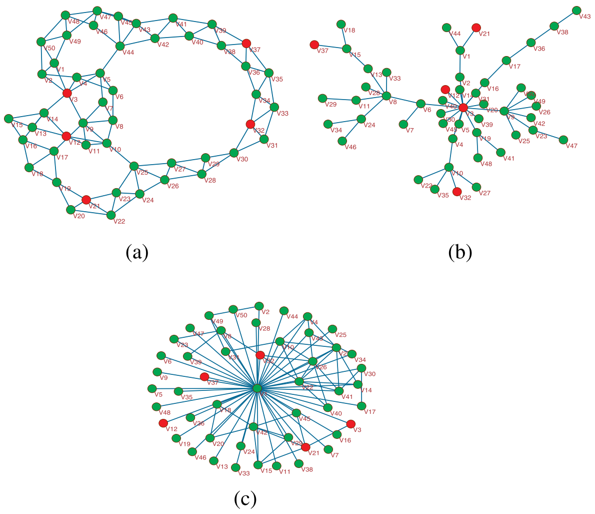

FIGURE 2. Three representative networks: (a) a synthetic small-world network G_{SW}, (b) a synthetic scale-free network G_{SF}, and (c) a subnet of Facebook network G_{FB}.

Scale-free networks are those that approximately obey a power-law degree distribution [29]. Empirical studies show that many real-world networks are scale-free [12]. Therefore, scale-free networks are adopted as another model of real-world networks. We set the number of nodes as $N = 50$, and get a synthetic scale-free network G_{SF} by Pajek, which is shown in Fig. 2(b), where U_{SF} consists of the red nodes.

Finally, we get a subnet with 50 nodes of Facebook network G_{FB} in Fig. 2(c), where U_{FB} consists of the red nodes.

B. HEURISTIC ALGORITHM FOR SOLVING MCBT MODEL

Second, in order to propose DOWNHILL algorithm, let us carry out some numerical simulations to inspect the optimal bandwidth for the MCBT model.

Experiment 1: Consider an MCBT model with $G = G_{SW}$, $U = U_{\mathrm{SW}}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.06$, $\beta_P = 0.08$, $\gamma_P = 0.02$, $c_1 = 3$, $c_2 = 2$, $\delta = 0.01$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \dots, 0.1)$. We present every x and its corresponding $C(x)$ in Fig. 2(a). As shown, $C(x)$ attains the minimum at $x = 1.4$ with $\theta = 0.5$, $C(x)$ attains the minimum at $x = 2.1$ with $\theta = 1$, and $C(x)$ attains the minimum at $x = 2.5$ with $\theta = 2$.

Experiment 2: Consider an MCBT model with $G = G_{SF}$, $U = U_{SF}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.02$, $\beta_P = 0.02$, $\gamma_P = 0.03$, $c_1 = 2$, $c_2 = 1$, $\delta = 0.03$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \dots, 0.1)$. We present every x and its corresponding $C(x)$ in Fig. 2(b). As shown, $C(x)$ attains the minimum at $x = 1.2$ with $\theta = 0.5$, $C(x)$ attains the minimum at $x = 1.9$ with $\theta = 1$, and $C(x)$ attains the minimum at $x = 2.4$ with $\theta = 2$.

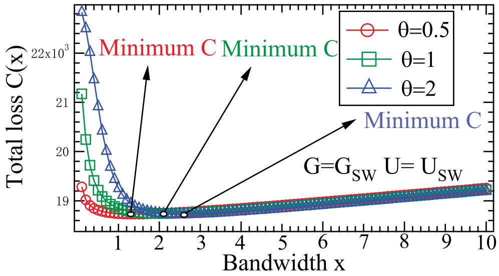

FIGURE 3. Total loss with different θ on G_{SW}.

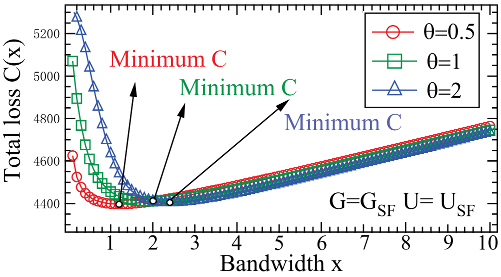

FIGURE 4. Total loss with different θ on G_{SF}.

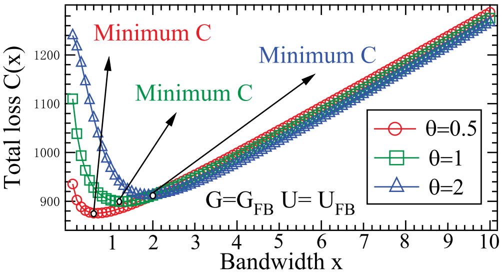

FIGURE 5. Total loss with different θ on G_{FB}.

Experiment 3: Consider an MCBT model with $G = G_{FB}$, $U = U_{FB}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.005$, $\beta_P = 0.001$, $\gamma_P = 0.05$, $c_1 = 2$, $c_2 = 1$, $\delta = 0.003$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *We present every x and its corresponding $C(x)$ in Fig. 2(c). As shown, $C(x)$ attains the minimum at $x = 0.7$ with $\theta = 0.5$, $C(x)$ attains the minimum at $x = 1.3$ with $\theta = 1$, and $C(x)$ attains the minimum at $x = 1.9$ with $\theta = 2$.*

The vertical axis of Figs. 3-5 is the total loss. From these figures, it can be seen that there exists a minimum total loss with different θ on small-world network, scale-free network and social network(i.e. a subnet of facebook).

Based on these and other similar numerical simulations, we conclude that the total loss first decreases and then increases with the increase of x. Thus, a heuristic algorithm in Algorithm 1 for solving the MCBT model is proposed.

We refer to x obtained by running DOWNHILL algorithm as *the DOWNHILL strategy*(i.e. optimal bandwidth), denoted as x^D, and refer to $C(x)$ as *the DOWNHILL total loss*(i.e. optimal total loss), denoted by $C(x^D)$.

IV. PERFORMANCE OF DOWNHILL ALGORITHM

In order to illustrate the effectiveness of DOWNHILL algorithm, we conduct some comparative experiments between DOWNHILL algorithm and random strategies on three larger realistic networks (i.e. Facebook, P2P and Email) with 500 nodes, which is denoted as $G_{facebook}$, G_{p2p}, and G_{email}, respectively.

Algorithm 1 DOWNHILL

```
Input: an MCBT model $\mathcal{M}_{VAT} = (G, U, \beta_I, \beta_P, \gamma_P, \delta, c_1, c_2, x_{\max}, T, initial, step, \mathbf{E}_0)$.
Output: $x$.
x := initial;
while $x \le x_{\max}$ and $C(x) > C(x + step)$ do
    $x := x + step$;
end while
return $x$;
```

A. COMPARATIVE EXPERIMENTS

In the following Experiment 4, we stochastically select 20 nodes as the antivirus program injection subset, denoted as $U_{facebook}$.

Experiment 4: Consider a set of MCBT models with $G = G_{facebook}$, $U = U_{facebook}$, $\theta \in \{1/3, 1, 2\}$, $\beta_I = 0.02$, $\beta_P = 0.02$, $\gamma_P = 0.03$, $c_1 = 1$, $c_2 = 1$, $\delta = 0.001$, $x_{max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *Let Z_0 represent the DOWNHILL strategy and Z_1 to Z_{100} denote the random strategies, respectively.*

In the following Experiment 5, we stochastically select 20 nodes as the antivirus program injection subset, denoted as U_{p2p}.

Experiment 5: Consider a set of MCBT models with $G = G_{p2p}$, $U = U_{p2p}$, $\theta \in \{1/3, 1, 2\}$, $\beta_I = 0.03$, $\beta_P = 0.01$, $\gamma_P = 0.05$, $c_1 = 2.5$, $c_2 = 1$, $\delta = 0.003$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *Let Z_0 represent the DOWNHILL strategy and Z_1 to Z_{100} denote the random strategies, respectively.*

In the following Experiment 6, we stochastically select 20 nodes as the antivirus program injection subset, denoted as U_{email}.

Experiment 6: Consider a set of MCBT models with $G = G_{email}$, $U = U_{email}$, $\theta \in \{1/3, 1, 2\}$, $\beta_I = 0.1$, $\beta_P = 0.002$, $\gamma_P = 0.3$, $c_1 = 1.5$, $c_2 = 2$, $\delta = 0.03$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *Let Z_0 represent the DOWNHILL strategy and Z_1 to Z_{100} denote the random strategies, respectively.*

Figs. 6-8 show the results of Experiments 4-6, respectively. It can be seen the total loss caused by DOWNHILL strategy(represented by the red circle) is smaller compared with random strategies(represented by the green circle) with different θ on Facebook, p2p, and email.

From these and other similar numerical simulations, we conclude that the DOWNHILL strategy leads to the minimum total loss compared with random strategies. Therefore, we infer DOWNHILL algorithm is the most effective algorithm for this problem.

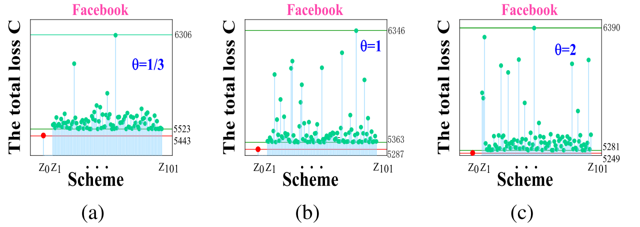

FIGURE 6. The results of Experiment 4: Total loss C vs. scheme on $G_{facebook}$ (a) with $\theta = 1/3$, (b) with $\theta = 1$, (c) with $\theta = 2$, respectively.

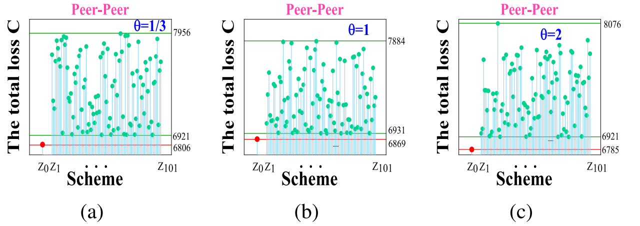

FIGURE 7. The results of Experiment 5: Total loss C vs. scheme on G_{p2p} (a) with $\theta = 1/3$, (b) with $\theta = 1$, (c) with $\theta = 2$, respectively.

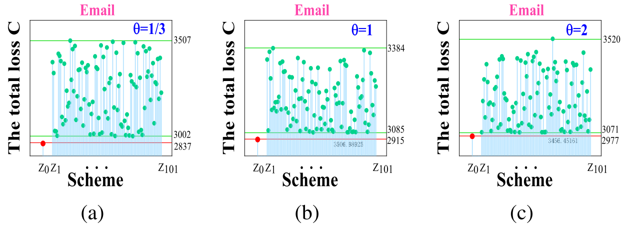

FIGURE 8. The results of Experiment 6: Total loss C vs. Scheme on G_{email} (a) with $\theta = 1/3$, (b) with $\theta = 1$, (c) with $\theta = 2$, respectively.

V. FURTHER DISCUSSIONS

This section is devoted to studying the influence of some factors on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

A. INFLUENCE OF MALICIOUS CODE INJECTION RATE

First, let us examine the influence of malicious code injection rate on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

Experiment 7: Consider a set of MCBT models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $U \in \{U_{SF}, U_{SW}, U_{FB}\}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I \in \{0.002, 0.004, \ldots, 0.04\}$, $\beta_P = 0.03$, $\gamma_P = 0.1$, $c_1 = 1$, $c_2 = 1$, $\delta = 0.02$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *By running DOWNHILL algorithm on these MCBT models, we obtain the optimal bandwidth* x^D *and the corresponding optimal total loss* $C(x^D)$. *Fig. 9(a) shows* x^D *vs.* β_I *on* G_{SW}, *Fig. 9(b) shows* $C(x^D)$ *vs.* β_I *on* G_{SW}, *Fig. 9(c) shows* x^D *vs.* β_I *on* G_{SF}, *Fig. 9(d) shows* $C(x^D)$ *vs.* β_I *on* G_{SF}, *Fig. 9(e) shows* x^D *vs.* β_I *on* G_{FB} *and Fig. 9(f) shows* $C(x^D)$ *vs.* β_I *on* G_{FB}*, respectively.*

Fig. 9 shows the results of Experiment 7. The vertical axis of Fig. 9(a), (c) and (e) is the optimal bandwidth; and the vertical axis of Fig. 9(b), (d), and (f) is the corresponding optimal total loss. In each subfigure, the red line refers to the results with $\theta = 0.5$, the green line denotes the results with $\theta = 1$, and the blue line represents the results with $\theta = 2$, respectively.

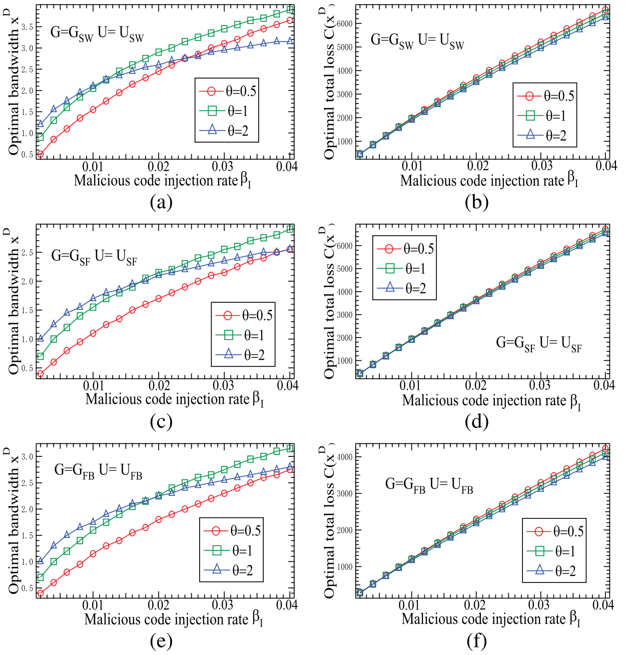

FIGURE 9. The experimental results in Experiment 7. (a) β_I vs. x^D on G_{SW}; (b) β_I vs. $C(x^D)$ on G_{SW}; (c) β_I vs. x^D on G_{SF}; (d) β_I vs. $C(x^D)$ on G_{SF}; (e) β_I vs. x^D on G_{FB}; (f) β_I vs. *$C(x^D)$ on G_{FB}.*

Based on these and other similar numerical simulations, we conclude that the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$ are both increasing with increase of malicious code injection rate. Therefore, it is better to allocate more bandwidth for antivirus programs with the rise of malicious code injection rate.

B. INFLUENCE OF MALICIOUS CODE PROPAGATION RATE

Second, let us inspect the influence of malicious code propagation rate on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

Experiment 8: Consider a set of MCBT models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $U \in \{U_{SF}, U_{SW}, U_{FB}\}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.005$, $\beta_P \in \{0.003, 0.006, \ldots, 0.06\}$, $\gamma_P = 0.05$, $c_1 = 1$, $c_2 = 1.5$, $\delta = 0.015$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *By running DOWNHILL algorithm on these MCBT models, we obtain the optimal bandwidth* x^D *and the corresponding optimal total loss* $C(x^D)$. *Fig. 10(a) shows* x^D *vs.* β_P *on* G_{SW}, *Fig. 10(b) shows* $C(x^D)$ *vs.* β_P *on* G_{SW}, *Fig. 10(c) shows* x^D *vs.* β_P *on* G_{SF}, *Fig. 10(d) shows* $C(x^D)$ *vs.* β_P *on* G_{SF}, *Fig. 10(e) shows* x^D *vs.* β_P *on* G_{FB} *and Fig. 10(f) shows* $C(x^D)$ *vs.* β_P *on* G_{FB}*, respectively.*

Fig. 10 shows the results of Experiment 8. The vertical axis of Fig. 10(a), (c) and (e) is the optimal bandwidth; and the vertical axis of Fig. 10(b), (d), and (f) is the corresponding optimal total loss. In each subfigure, the red line refers to the results with $\theta = 0.5$, the green line denotes the results

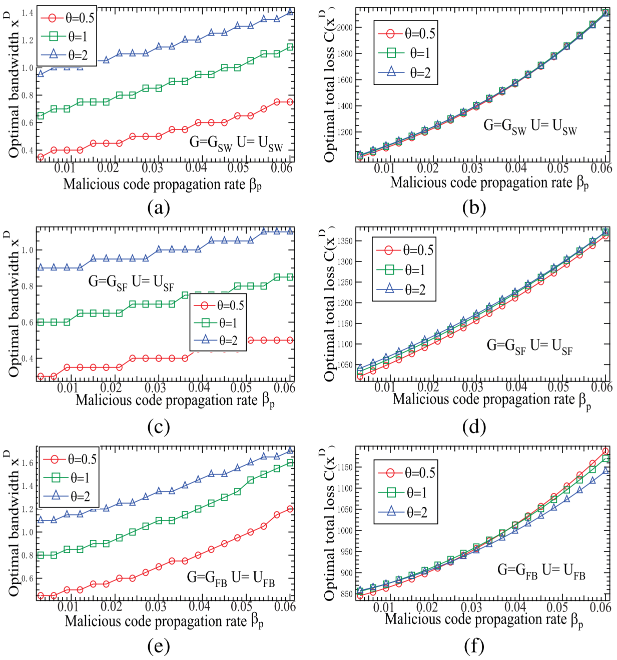

FIGURE 10. The experimental results in Experiment 8. (a) β_P vs. x^D on G_{SW}; (b) β_P vs. $C(x^D)$ on G_{SW}; (c) β_P vs. x^D on G_{SF}; (d) β_P vs. $C(x^D)$ on G_{SF}; (e) β_P vs. x^D on G_{FB}; (f) β_P vs. $C(x^D)$ on G_{FB}.

with $\theta = 1$, and the blue line represents the results with $\theta = 2$, respectively.

Based on these and other similar numerical simulations, we conclude that the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$ are both increasing with malicious code propagation rate. Therefore, we need to allocate more bandwidth for antivirus programs with the rise of malicious code propagation rate.

C. INFLUENCE OF ANTIVIRUS PROGRAM PROPAGATION RATE

Third, we consider the influence of antivirus program propagation rate on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

Experiment 9: Consider a set of MCBT models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $U \in \{U_{SF}, U_{SW}, U_{FB}\}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.1$, $\beta_P = 0.05$, $\gamma_P \in \{0.2, 0.22, \ldots, 0.58\}$, $c_1 = 2$, $c_2 = 1$, $\delta = 0.01$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *By running DOWNHILL algorithm on these MCBT models, we obtain the optimal bandwidth* x^D *and the corresponding optimal total loss* $C(x^D)$. *Fig. 11(a) shows* x^D *vs.* γ_P *on* G_{SW}, *Fig. 11(b) shows* $C(x^D)$ *vs.* γ_P *on* G_{SW}, *Fig. 11(c) shows* x^D *vs.* γ_P *on* G_{SF}, *Fig. 11(d) shows* $C(x^D)$ *vs.* γ_P *on* G_{SF}, *Fig. 11(e) shows* x^D *vs.* γ_P *on* G_{FB} *and Fig. 11(f) shows* $C(x^D)$ *vs.* γ_P *on* G_{FB}, *respectively.*

Fig. 11 shows the results of Experiment 9. The vertical axis of Fig. 11(a), (c) and (e) is the optimal bandwidth; and the vertical axis of Fig. 11(b), (d), and (f) is the corresponding optimal total loss. In each subfigure, the red line refers to the results with $\theta = 0.5$, the green line denotes the results with $\theta = 1$, and the blue line represents the results with $\theta = 2$, respectively.

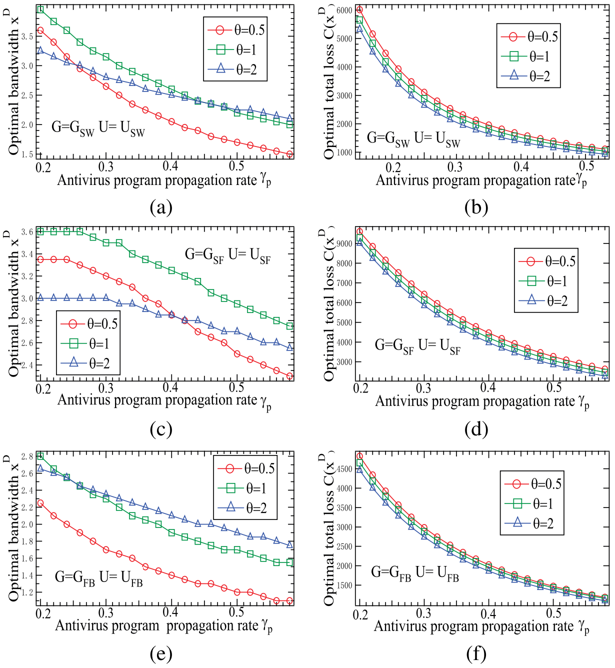

FIGURE 11. The experimental results in Experiment 9. (a) γ_P vs. x^D on G_{SW}; (b) γ_P vs. $C(x^D)$ on G_{SW}; (c) γ_P vs. x^D on G_{SF}; (d) γ_P vs. $C(x^D)$ on G_{SF}; (e) γ_P vs. x^D on G_{FB}; (f) γ_P vs. $C(x^D)$ on G_{FB}.

Based on these and other similar numerical simulations, we conclude that the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$ are both decreasing with the increase of antivirus program propagation rate. Therefore, we are supposed to allocate less bandwidth for antivirus programs with the rise of antivirus program propagation rate.

D. INFLUENCE OF ANTIVIRUS PROGRAM FAILURE RATE

Fourth, we investigate the influence of antivirus program failure rate on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

Experiment 10: Consider a set of MCBT models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $U \in \{U_{SF}, U_{SW}, U_{FB}\}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.03$, $\beta_P = 0.002$, $\gamma_P = 0.01$, $c_1 = 1$, $c_2 = 1$, $\delta \in \{0.001, 0.002, \ldots, 0.02\}$, $x_{\max} = 10$, $T = 10$, $initial = 0.05$, $step = 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *By running DOWNHILL algorithm on these MCBT models, we obtain the optimal bandwidth* x^D *and the corresponding optimal total loss* $C(x^D)$. *Fig. 12(a) shows* x^D *vs.* δ *on* G_{SW}, *Fig. 12(b) shows* $C(x^D)$ *vs.* δ *on* G_{SW}, *Fig. 12(c) shows* x^D *vs.* δ *on* G_{SF}, *Fig. 12(d) shows* $C(x^D)$ *vs.* δ *on* G_{SF}, *Fig. 12(e) shows* x^D *vs.* δ *on* G_{FB} *and Fig. 12(f) shows* $C(x^D)$ *vs.* δ *on* G_{FB}, *respectively.*

Fig. 12 shows the results of Experiment 10. The vertical axis of Fig. 12(a), (c) and (e) is the optimal bandwidth; and the vertical axis of Fig. 12(b), (d), and (f) is the corresponding optimal total loss. In each subfigure, the red line refers to the results with $\theta = 0.5$, the green line denotes the results with $\theta = 1$, and the blue line represents the results with $\theta = 2$, respectively.

Based on these and other similar numerical simulations, we conclude that the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$ are both increasing with

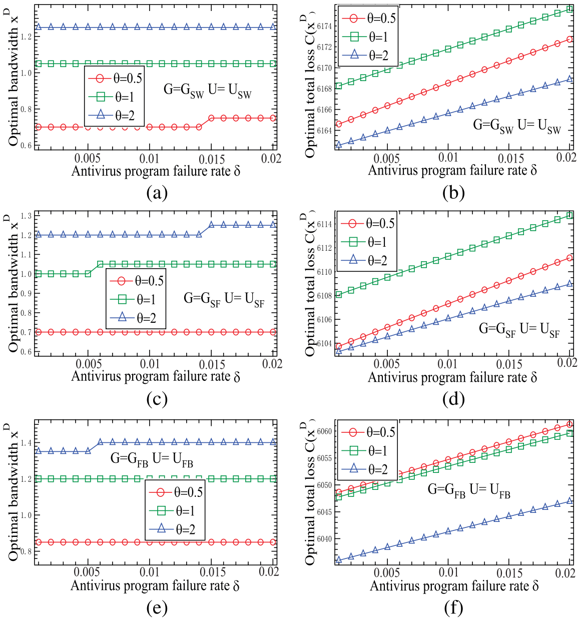

FIGURE 12. The experimental results in Experiment 10. (a) δ vs. x^D on G_{SW}; (b) δ vs. $C(x^D)$ on G_{SW}; (c) δ vs. x^D on G_{SF}; (d) δ vs. $C(x^D)$ on G_{SF}; (e) δ vs. x^D on G_{FB}; (f) δ vs. $C(x^D)$ on G_{FB}.

the increase of antivirus program failure rate. Therefore, we should allocate more bandwidth for antivirus programs with the rise of antivirus program failure rate.

E. INFLUENCE OF COST COEFFICIENT

Now, we investigate the influence of cost coefficient on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

Experiment 11: Consider a set of MCBT models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $U \in \{U_{SF}, U_{SW}, U_{FB}\}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.08$, $\beta_P = 0.002$, $\gamma_P = 0.009$, $\delta = 0.02$,$c_1 \in \{0.1, \ldots, 0.2, 2\}$, $c_2 = 1$, $x_{\max} = 10$, $T = 10$, *initial* $= 0.05$, *step* $= 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *By running DOWNHILL algorithm on these MCBT models, we obtain the optimal bandwidth* x^D *and the corresponding optimal total loss* $C(x^D)$. *Fig. 13(a) shows* x^D *vs.* c_1 *on* G_{SW}, *Fig. 13(b) shows* $C(x^D)$ *vs.* c_1 *on* G_{SW}, *Fig. 13(c) shows* x^D *vs.* c_1 *on* G_{SF}, *Fig. 13(d) shows* $C(x^D)$ *vs.* c_1 *on* G_{SF}, *Fig. 13(e) shows* x^D *vs.* c_1 *on* G_{FB} *and Fig. 13(f) shows* $C(x^D)$ *vs.* c_1 *on* G_{FB}, *respectively.*

Fig. 13 shows the results of Experiment 11. The vertical axis of Fig. 13(a), (c) and (e) is the optimal bandwidth; and the vertical axis of Fig. 13(b), (d), and (f) is the corresponding optimal total loss. In each subfigure, the red line refers to the results with $\theta = 0.5$, the green line denotes the results with $\theta = 1$, and the blue line represents the results with $\theta = 2$, respectively.

Based on these and other similar numerical simulations, we conclude that the optimal bandwidth x^D is decreasing with the increase of cost coefficient while the corresponding optimal total loss $C(x^D)$ is increasing. Therefore, we have allocate less bandwidth for antivirus programs with the rise of cost coefficient.

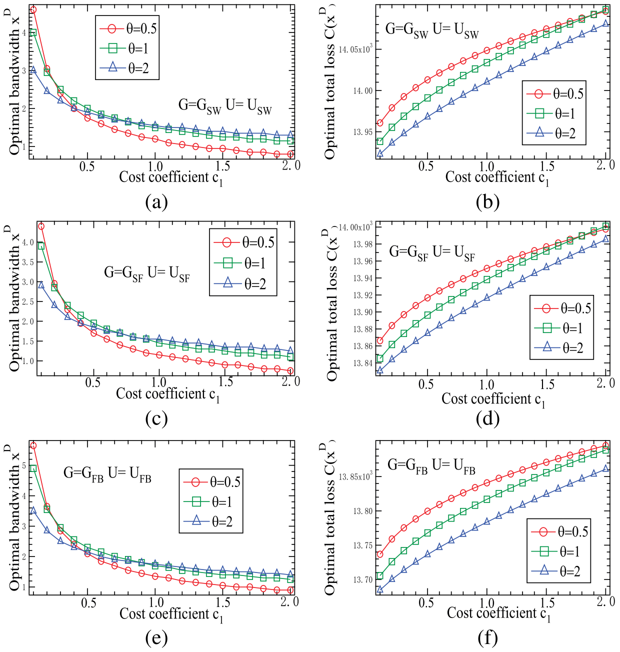

FIGURE 13. The experimental results in Experiment 11. (a) c_1 vs. x^D on G_{SW}; (b) c_1 vs. $C(x^D)$ on G_{SW}; (c) c_1 vs. x^D on G_{SF}; (d) c_1 vs. $C(x^D)$ on G_{SF}; (e) c_1 vs. x^D on G_{FB}; (f) c_1 vs. $C(x^D)$ on G_{FB}.

F. INFLUENCE OF ANTIVIRUS PROGRAM INJECTION COEFFICIENT

Next, we study the influence of antivirus program injection coefficient on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

Experiment 12: Consider a set of MCBT models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $U \in \{U_{SF}, U_{SW}, U_{FB}\}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.01$, $\beta_P = 0.002$, $\gamma_P = 0.1$, $c_1 = 1.5$, $c_2 \in \{0.1, 0.2, \ldots, 2\}$, $\delta = 0.002$, $x_{\max} = 10$, $T = 10$, *initial* $= 0.05$, *step* $= 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *By running DOWNHILL algorithm on these MCBT models, we obtain the optimal bandwidth* x^D *and the corresponding optimal total loss* $C(x^D)$, *respectively. Fig. 14(a) shows* x^D *vs.* c_2 *on* G_{SW}, *Fig. 14(b) shows* $C(x^D)$ *vs.* c_2 *on* G_{SW}, *Fig. 14(c) shows* x^D *vs.* c_2 *on* G_{SF}, *Fig. 14(d) shows* $C(x^D)$ *vs.* c_2 *on* G_{SF}, *Fig. 14(e) shows* x^D *vs.* c_2 *on* G_{FB} *and Fig. 14(f) shows* $C(x^D)$ *vs.* c_2 *on* G_{FB}, *respectively.*

Fig. 14 shows the results of Experiment 12. The vertical axis of Fig. 14(a), (c) and (e) is the optimal bandwidth; and the vertical axis of Fig. 14(b), (d), and (f) is the corresponding optimal total loss. In each subfigure, the red line refers to the results with $\theta = 0.5$, the green line denotes the results with $\theta = 1$, and the blue line represents the results with $\theta = 2$, respectively.

Based on these and other similar numerical simulations, we conclude that the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$ are both decreasing with the increase of antivirus program injection coefficient. Therefore, we need to allocate less bandwidth for antivirus programs with the rise of antivirus program injection coefficient.

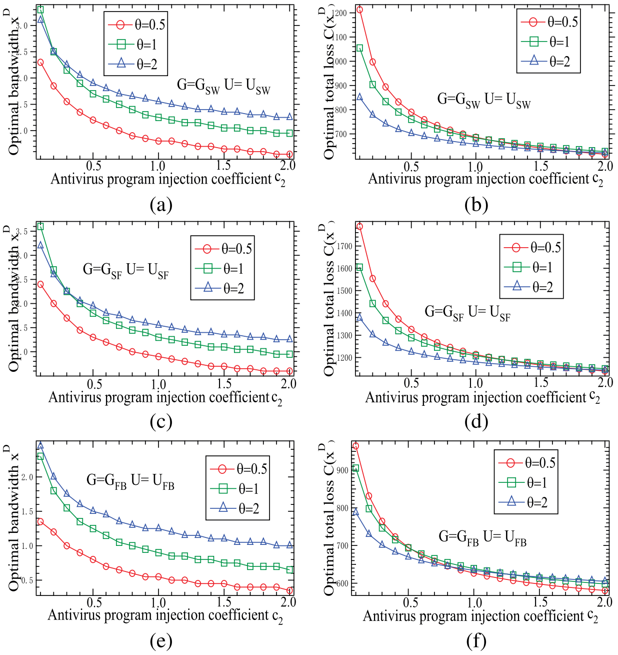

FIGURE 14. The experimental results in Experiment 12. (a) c_2 vs. x^D on G_{SW}; (b) c_2 vs. $C(x^D)$ on G_{SW}; (c) c_2 vs. x^D on G_{SF}; (d) c_2 vs. $C(x^D)$ on G_{SF}; (e) c_2 vs. x^D on G_{FB}; (f) c_2 vs. $C(x^D)$ on G_{FB}.

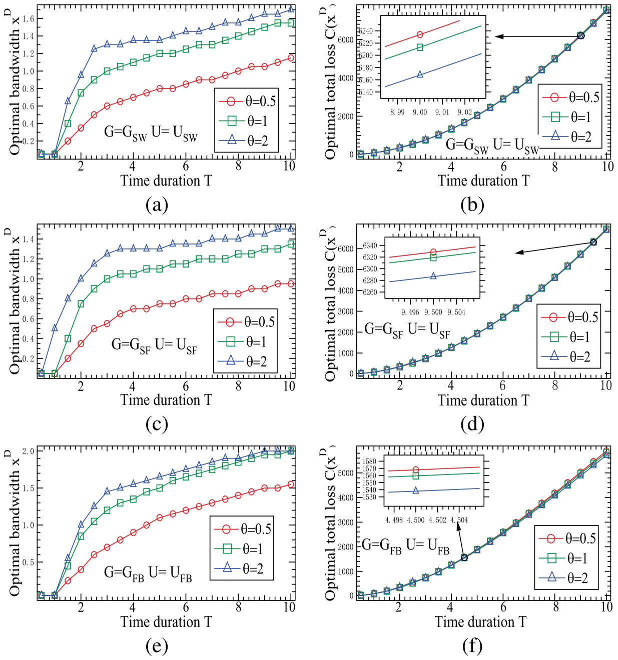

FIGURE 15. The experimental results in Example 13. (a) T vs. x^D on G_{SW}; (b) T vs. $C(x^D)$ on G_{SW}; (c) T vs. x^D on G_{SF}; (d) T vs. $C(x^D)$ on G_{SF}; (e) T vs. x^D on G_{FB}; (f) T vs. $C(x^D)$ on G_{FB}.

G. INFLUENCE OF TIME DURATION

Finally, we study the influence of time duration on the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$, respectively.

Experiment 13: Consider a set of MCBT models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $U \in \{U_{SF}, U_{SW}, U_{FB}\}$, $\theta \in \{0.5, 1, 2\}$, $\beta_I = 0.035$, $\beta_P = 0.02$, $\gamma_P = 0.045$, $c_1 = 2$, $c_2 = 1$, $\delta = 0.02$, $x_{\max} = 10$, $T \in \{0.5, 1, \ldots, 10\}$, *initial* $= 0.05$, *step* $= 0.05$, $\mathbf{I}_0 = (0.1, \ldots, 0.1)$. *By running DOWNHILL algorithm on these MCBT models, we obtain the optimal bandwidth* x^D *and the corresponding optimal total loss* $C(x^D)$*, respectively. Fig. 15(a) shows* x^D *vs.* T *on* G_{SW}*, Fig. 15(b) shows* $C(x^D)$ *vs.* T *on* G_{SW}*, Fig. 15(c) shows* x^D *vs.* T *on* G_{SF}*, Fig. 15(d) shows* $C(x^D)$ *vs.* T *on* G_{SF}*, Fig. 15(e) shows* x^D *vs.* T *on* G_{FB} *and Fig. 15(f) shows* $C(x^D)$ *vs.* T *on* G_{FB}*, respectively.*

Fig. 15 shows the results of Experiment 13. The vertical axis of Fig. 15(a), (c) and (e) is the optimal bandwidth; and the vertical axis of Fig. 15(b), (d), and (f) is the corresponding optimal total loss. In each subfigure, the red line refers to the results with $\theta = 0.5$, the green line denotes the results with $\theta = 1$, and the blue line represents the results with $\theta = 2$, respectively.

Based on these and other similar numerical simulations, we conclude that the optimal bandwidth x^D and the corresponding optimal total loss $C(x^D)$ are both increasing with the increase of time duration. Therefore, we would better allocate more bandwidth for antivirus programs with the rise of time duration.

VI. CONCLUDING REMARKS

Antivirus program plays a key role in mitigating the impact of malicious codes. We address the problem of weighing the economic loss caused by malicious codes and the bandwidth assigned to transmit antivirus programs(i.e. the MCBT problem). First, a novel malicious code and antivirus program interacting model is proposed. On this basis, the MCBT problem is modelled as an optimization problem (the MCBT model). We get some optimal bandwidth by solving the MCBT model. Based on this, we propose a heuristic algorithm named DOWNHILL, which performs better compared with random strategies. Finally, the influence of some factors on the optimal bandwidth and the corresponding optimal total loss is uncovered through numerical simulations.

There are some relevant research topics towards this direction. In this work, the bandwidth is simply assumed to be fixed. In reality, we can flexibly change the bandwidth to achieve a better tradeoff, and this can be done in the framework of evolutionary algorithms [30], [31].The methodology developed in this work can be applied to situations where wireless sensor network [32], [33] and rumor spreading [34], [35].

APPENDIX

By these hypotheses, we get a differential dynamical system as follows. Let $\Delta t > 0$ denote a very small time interval. Hypotheses (H_1)-(H_5) imply the following relations.

$$\Pr\{X_i(t+\Delta t) = 1 \mid X_i(t) = 0\} = \Delta t\left[\beta_I + \beta_P \sum_{j=1}^{N} a_{ij} I_j(t)\right] + o(\Delta t),$$

$$\Pr\{X_i(t+\Delta t) = 2 \mid X_i(t) = 0\} = \Delta t \gamma_I + o(\Delta t),\ i \in U,$$

$$\Pr\{X_i(t+\Delta t) = 2 \mid X_i(t) = 0\} = \Delta t\left[\gamma_P \sum_{j=1}^{N} a_{ij} \gamma_j(t)\right] + o(\Delta t),\ i \in V - U,$$

$$\Pr\{X_(t+\Delta t)=2 \mid X_i(t)=1\} = \Delta t\gamma_I + o(\Delta t), i \in U,$$

$$\Pr\{X_(t+\Delta t)=2 \mid X_i(t)=1\} = \Delta t\left[\gamma_p \sum_{j=1}^{N} a_{ij}\gamma_j(t)\right] + o(\Delta t), i \in V-U,$$

$$\Pr\{X_i(t+\Delta t)=0 \mid X_i(t)=1\} = o(\Delta t),$$

$$\Pr\{X_i(t+\Delta t)=0 \mid X_i(t)=2\} = \Delta t\delta + o(\Delta t),$$

$$\Pr\{X_i(t+\Delta t)=1 \mid X_i(t)=2\} = o(\Delta t).$$

As a result, we have

$$\Pr\{X_i(t+\Delta t)=0 \mid X_i(t)=0\}$$
$$= 1-\Delta t\left[\beta_I+\beta_p\sum_{j=1}^{N} a_{ij}I_j(t)\right] - \Delta t\gamma_I + o(\Delta t), i \in U,$$

$$\Pr\{X_i(t+\Delta t)=0 \mid X_i(t)=0\}$$
$$= 1-\Delta t\left[\beta_I+\beta_p\sum_{j=1}^{N} a_{ij}I_j(t)\right]$$
$$-\Delta t\left[\gamma_p\sum_{j=1}^{N} aijP_j(t)\right] + o(\Delta t), \quad i \in V-U,$$

$$\Pr\{X_i(t+\Delta t)=1 \mid X_i(t)=1\}$$
$$= 1-\gamma_I\Delta t + o(\Delta t), i \in U,$$

$$\Pr\{X_i(t+\Delta t)=1 \mid X_i(t)=1\}$$
$$= 1-\Delta t\left[\gamma_p\sum_{j=1}^{N} a_{ij}P_j(t)\right] + o(\Delta t), i \in V-U,$$

$$\Pr\{X_i(t+\Delta t)=2 \mid X_i(t)=2\}$$
$$= 1-\delta\Delta t + o(\Delta t).$$

By the total probability formula and transposing the terms $I_i(t)$ and $P_i(t)$ from the right to the left, and dividing both sides by Δt, we get

$$\frac{I_i(t+\Delta t)-I_i(t)}{\Delta t} = \left[\beta_I+\beta_P\sum_{j=1}^{N} a_{ij}I_j(t)\right] \times [1-I_i(t)-P_i(t)] - \gamma_I I_i(t) + \frac{o(\Delta t)}{\Delta t}, \quad t \geq 0, i \in U,$$

$$\frac{P_i(t+\Delta t)-P_i(t)}{\delta t} = \gamma_I[1-P_i(t)] - \delta P_i(t) + \frac{o(\Delta t)}{\Delta t}, \quad t \geq 0, i \in U,$$

$$\frac{I_i(t+\Delta t)-I_i(t)}{\Delta t} = \left[\beta_I+\beta_P\sum_{j=1}^{N} a_{ij}I_j(t)\right] \times [1-I_i(t)-P_i(t)] - \gamma_P I_i(t)\sum_{j=1}^{N} a_{ij}P_j(t) + \frac{o(\Delta t)}{\Delta t}, \quad t \geq 0, i \in V-U,$$

$$\frac{P_i(t+\Delta t)-P_i(t)}{\delta t} = \gamma_P[1-P_i(t)]\sum_{j=1}^{N} a_{ij}P_j(t) - \delta P_i(t) + \frac{o(\Delta t)}{\Delta t}, \quad t \geq 0, i \in V-U.$$

Letting $\Delta t \to 0$, we get the following dynamical model.

$$\begin{cases} \dfrac{dI_i(t)}{dt} = \left[\beta_I+\beta_P\sum_{j=1}^{N} a_{ij}I_j(t)\right][1-I_i(t)-P_i(t)] \\ \quad -\gamma_I I_i(t), t \geq 0, i \in U, \\ \dfrac{dP_i(t)}{dt} = \gamma_I[1-P_i(t)] - \delta P_i(t), t \geq 0, i \in U, \\ \dfrac{dI_i(t)}{dt} = \left[\beta_I+\beta_P\sum_{j=1}^{N} a_{ij}I_j(t)\right][1-I_i(t)-P_i(t)] \\ \quad -\gamma_P I_i(t)\sum_{j=1}^{N} a_{ij}P_j(t), t \geq 0, i \in V-U, \\ \dfrac{dP_i(t)}{dt} = \gamma_P[1-P_i(t)]\sum_{j=1}^{N} a_{ij}P_j(t) - \delta P_i(t), \\ \quad t \geq 0, i \in V-U. \end{cases}$$

where $E(0) = E_0, t \geq 0, 1 \leq i \leq N$.

REFERENCES

[1] B. A. Forouzan and F. Mosharraf, *Computer Networks: A Top-Down Approach*. New York, NY, USA: McGraw-Hill, 2012.

[2] S. Mohurle and M. Patil, "A brief study of wannacry threat: Ransomware attack 2017," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, May 2017.

[3] S. Mansfield-Devine, "Leaks and ransoms—The key threats to healthcare organisations," *Netw. Secur.*, vol. 2017, no. 6, pp. 14–19, Jun. 2017.

[4] P. Szor, *The Art of Computer Virus Research and Defense: Art Comp Virus Res Defense_p1*. London, U.K.: Pearson, 2005.

[5] J. Bi, F. Zhang, A. Dorri, C. Zhang, and C. Zhang, "A risk management approach to double-virus tradeoff problem," *IEEE Access*, vol. 7, pp. 144472–144480, 2019.

[6] Z. Ma, Y. Zhou, and J. Wu, *Modeling and Dynamics of Infectious Diseases*, vol. 11. Singapore: World Scientific, 2009.

[7] L.-C. Chen and K. Carley, "The impact of countermeasure propagation on the prevalence of computer viruses," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 823–833, Apr. 2004.

[8] J. Goldenberg, Y. Shavitt, E. Shir, and S. Solomon, "Distributive immunization of networks against viruses using the 'honey-pot' architecture," *Nature Phys.*, vol. 1, no. 3, pp. 184–188, Dec. 2005.

[9] A. Misra, M. Verma, and A. Sharma, "Capturing the interplay between malware and anti-malware in a computer network," *Appl. Math. Comput.*, vol. 229, pp. 340–349, Feb. 2014.

[10] L.-X. Yang and X. Yang, "A novel virus-patch dynamic model," *PLoS ONE*, vol. 10, no. 9, Sep. 2015, Art. no. e0137858.

[11] W. Liu and S. Zhong, "Web malware spread modelling and optimal control strategies," *Sci. Rep.*, vol. 7, Feb. 2017, Art. no. 42308.

[12] R. Albert and A.-L. Barabási "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, p. 47, Jan. 2002.

[13] R. Pastor-Satorras and A. Vespignani, "Epidemic dynamics in finite size scale-free networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, Mar. 2002, Art. no. 035108.

[14] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Rev. Mod. Phys.*, vol. 87, no. 3, pp. 925–979, Aug. 2015.

[15] J. Ren, Y. Xu, and J. Liu, "Investigation of dynamics of a virus-antivirus model in complex network," *Phys. A, Stat. Mech. Appl.*, vol. 421, pp. 533–540, Mar. 2015.

[16] I. Stojmenović, *Handbook of Wireless Networks and Mobile Computing*. Hoboken, NJ, USA: Wiley, 2002.

[17] A. Boukerche, *Handbook of Algorithms for Wireless Networking and Mobile Computing*. Boca Raton, FL, USA: CRC Press, 2005.

[18] J. Rodriguez, *Fundamentals of 5G Mobile Networks*. Hoboken, NJ, USA: Wiley, 2015.

[19] P. Van Mieghem, J. Omic, and R. Kooij, "Virus spread in networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 1–14, Feb. 2009.

[20] P. Van Mieghem, "The N-intertwined SIS epidemic network model," *Computing*, vol. 93, nos. 2–4, pp. 147–169, Dec. 2011.

[21] J. Bi, X. Yang, Y. Wu, Q. Xiong, J. Wen, and Y. Y. Tang, "On the optimal dynamic control strategy of disruptive computer virus," *Discrete Dyn. Nature Soc.*, vol. 2017, pp. 1–14, Mar. 2017.

[22] J. Bi, L.-X. Yang, X. Yang, Y. Wu, and Y. Y. Tang, "A tradeoff between the losses caused by computer viruses and the risk of the manpower shortage," *PLoS ONE*, vol. 13, no. 1, Jan. 2018, Art. no. e0191101.

[23] L.-X. Yang, P. Li, Y. Zhang, X. Yang, Y. Xiang, and W. Zhou, "Effective repair strategy against advanced persistent threat: A differential game approach," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1713–1728, Jul. 2019.

[24] W. Liu, "Modeling ransomware spreading by a dynamic node-level method," *IEEE Access*, vol. 7, pp. 142224–142232, 2019.

[25] J. Balthrop, S. Forrest, M. E. Newman, and M. M. Williamson, "Technological networks and the spread of computer viruses," *Science*, vol. 304, no. 5670, pp. 527–529, Apr. 2004.

[26] W. J. Stewart, *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[27] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[28] W. De Nooy, A. Mrvar, and V. Batagelj, *Exploratory Social Network Analysis With Pajek: Revised and Expanded Edition for Updated Software*, vol. 46. Cambridge, U.K.: Cambridge Univ. Press, 2018,

[29] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[30] F. Luo, Z. Y. Dong, Y. Chen, and J. Zhao, "Natural aggregation algorithm: A new efficient metaheuristic tool for power system optimizations," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Nov. 2016, pp. 186–192.

[31] F. Luo, J. Zhao, and Z. Y. Dong, "A new metaheuristic algorithm for real-parameter optimization: Natural aggregation algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 94–103.

[32] M. Sayad Haghighi, S. Wen, Y. Xiang, B. Quinn, and W. Zhou, "On the race of worms and patches: Modeling the spread of information in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2854–2865, Dec. 2016.

[33] S. He, J. Chen, F. Jiang, D. K. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1931–1942, Oct. 2013.

[34] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2789–2800, Mar. 2017.

[35] W. Liu, X. Wu, W. Yang, X. Zhu, and S. Zhong, "Modeling cyber rumor spreading over mobile social networks: A compartment approach," *Appl. Math. Comput.*, vol. 343, pp. 214–229, Feb. 2019.

JICHAO BI (Student Member, IEEE) currently pursuing the Ph.D. degree with the School of Big Data and Software Engineering, Chongqing University, China. He was a Visiting Ph.D. Student with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia, from 2018 to 2019. He has published three academic articles in peer-reviewed international journals. His research interests include network dynamics, cybersecurity, and smart grid.

XIAOFAN YANG (Member, IEEE) received the B.Sc. degree from the Department of Mathematics, Sichuan University, in 1985, the M.Sc. degree from the Department of Applied Mathematics, Chongqing University, in 1988, and the Ph.D. degree from the Department of Computer Science, Chongqing University, in 1994. He joined Chongqing University, in 1987. He visited the University of Reading, Reading, U.K., from 1998 to 1999, Hong Kong Baptist University, in 2005, 2007, and 2009, and University of Macau, in 2016 and 2017. He is currently a Professor of computer science with Chongqing University. He has published more than 150 articles in peer-reviewed international journals, and more than 20 students have received the Ph.D. degree under his supervision. His research interests include computer virus spreading, cybersecurity and fault tolerant computing, and applied nonlinear dynamics.

WANPING LIU (Member, IEEE) received the B.Sc. degree in mathematical science and the Ph.D. degree in computer science from Chongqing University, China, in 2009 and 2014, respectively. He was a Visiting Ph.D. Student with the University of Waterloo, Canada, from 2012 to 2013, and a Postdoctoral Research with the University of Electronic Science and Technology of China, from 2016 to 2019. He is currently an Associate Professor with the Chongqing University of Technology, China. He has published more than 60 publications in related research areas. His research interests include mathematical modeling, complex systems, and network science.

DA-WEN HUANG received the B.Sc. degree from the Department of Mathematics, College of Science, China Three Gorges University, in 2014, and the M.Sc. degree from the School of Mathematics and Computational Sciences, Xiangtan University, in 2017. He is currently pursuing the Ph.D. degree with Chongqing University. He has published four academic articles in peer-reviewed international journals. His research interests include cybersecurity, wireless sensor networks, network dynamics, and data mining.