A Risk Management Approach to Double-virus Tradeoff Problem

JICHAO BI,(Student Member, IEEE)\(^2\), FANGFEI ZHANG,(Student Member, IEEE)\(^3\), ALI DORRI,(Student Member, IEEE)\(^5\), CHUNMING ZHANG\(^1\), AND CHEN ZHANG\(^4\)

\(^1\)School of Information Engineering, Guangdong Medical University, Dongguan, P.R. China
\(^2\)School of Big Data & Software Engineering, Chongqing University, Chongqing, P.R. China
\(^3\)School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia
\(^4\)State Grid Zhejiang Electric Power Corporation Information & Telecommunication Branch, Hangzhou, P.R. China
\(^5\)School of Computer Science and Engineering, University of New South Wales, Sydney, Australia.

Corresponding author: Chunming Zhang (e-mail: chunfei2002@163.com).

This work is supported by Natural Science Foundation of Guangdong Province, China (Grant No. 2018A0303130049).

ABSTRACT One of the major threats to cybersecurity is the emergence of new computer viruses. By emergence of new viruses, the cybersecurity companies assign a team of security experts and programmers to study the behavior of the virus and develop the corresponding antivirus program to secure networks. Sometimes, more than one new virus is identified that requires the cybersecurity team to make a tradeoff on the allocation of programmers, and thus leads to two-antivirus-program development problem under double-virus attack. In this paper, we propose DOWNHILL algorithm to address the outlined challenge. We model the time evolution of the expected state of the network as a differential dynamical system to measure the total loss caused by the viruses. Then, we propose a DOWNHILL algorithm, three heuristic algorithms and a random algorithm to solve the problem, respectively. We study the computational complexities of the proposed algorithms as well. Through numerous comparative experiments, we confirm the DOWNHILL algorithm is the most effective method to this problem. Finally, the influence of different factors on the DOWNHILL strategy and its potential total loss are also researched.

INDEX TERMS Computer virus, Antivirus, Dynamic model, Constrained optimization.

I. INTRODUCTION

In recent years, computer networks grows rapidly in a way that not only computers but also every thing around us are getting connected that provides sophisticated personalized services to the end-users, e.g., smart farming or smart energy management. Due to ever increasing connectivity of computers and networks, if a computer (or group of computers) in a network malfunction, information cascades rapidly encompass other entities in the network which impacts the functionality of the network [1]. One of the most common threats to computer networks is computer virus which can spread itself among computers in a network by exploiting hardware or software vulnerabilities, e.g., operating systems or mis-configuration of devices. The impact of computer viruses inflict huge economic loss [2]. Authors in [3]–[6] estimated the total monetary loss due to the impact of computer viruses.

Cybersecurity companies identify new viruses by analyzing the log files, operation records and reports collected from the end-users. Once a new virus is detected, the company abstracts its feature codes, store them into a database, and allocate programmers to develop the corresponding antivirus program to protect user devices from being compromised. The company first estimates the amount of required effort, which is the first requirement to design new programs according to software engineering fundamentals, and is usually measured as man-months or man-days unit [7]–[10]. We employ man-days in this paper considering the fundamental features of antivirus program, such as urgency and timeliness. Next, the number of programmers to be assigned to code the antivirus is determined.

The networks, or generally the Internet, might be attacked by two different viruses (say virus I and II ) at the same time which is known as double-virus attack. The cybersecurity company shall dedicate team to develop antivirus for each virus simultaneously to minimize the potential economic loss. Note that the number of professional team members on the company is limited. Thus, the company requires to consider a tradeoff between economic loss and number of professionals that are assigned to each virus. We refer to the problem of allocating the limited number of programmers
to balance the two conflicting demands as the two-antivirus-program development problem. The allocation to each of the programmers is also referred as allocation strategy.

Once a virus is settled in a device, it attempts to reach other devices in the network by taking advantages of network protocols, e.g., TCP/IP or SMTP. By injecting other devices, the virus can potentially cause a greater harm as compared to single device injection. The propagation of computer viruses is characterized by individual-level epidemic models [11], [12] that can accurately capture the propagation process of any virus on an arbitrary network.

In this paper, we study the outlined problem under the Susceptible-Infected (SI) model as discussed in [13], [14]. SI model captures the spreading a new virus before the corresponding antivirus program is released. We evaluate the potential loss of the network using risk analysis methods as in [15]. First, we evaluate the expected loss of the network per unit of time. As this metric is determined based on the expected state of the network, we need to model the time evolution of the expected state of the network. Following, the potential total loss of the network can be obtained by accumulating them.

The main contributions of this paper are as follows:
- We introduce an individual-level state evolution model that captures the expected state of a network under double-virus attack, and obtain the potential total loss of the network. We introduce two-antivirus-program development problem. We convert this problem to an optimization problem that can be solved mathematically.
- We propose DOWNHILL algorithm to address the two-antivirus-program development problem. Simulation results prove that DOWNHILL algorithm achieves better performance as compared with three heuristic algorithms and the random algorithm.
- We study the impact of different metrics on the DOWNHILL strategy and its potential total loss. This potentially enables cybersecurity companies to quickly action against double-virus attack.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 outlines the two-antivirus-program development problem. Section 4, we propose DOWNHILL algorithm to solve the problem outlined in Section 3. We undertake comparative experiments to study the performance of our method that are outlined in Section 5. Section 6 studies the influence of different factors on the DOWNHILL strategy and the potential total loss of DOWNHILL algorithm. Finally, Section 7 concludes the paper.

II. RELATED WORK

Computer virus propagation dynamics is a new applied mathematics that studies the steps involved in spreading a computer virus to mitigate the impact of virus attacks [16]. Originally, the study was focused on virus propagation based on homogeneous networks [17]. Later, [18], [19] set off a wave of research on virus propagation based on scale-free networks. With the advance in wireless networks, today’s computer networks may be deployed under any circumstances and at any location [20]. To understand a variety of propagation phenomena on arbitrary computer networks, multiple propagation models based on arbitrary networks are proposed [21], [22].

There exists three main methods to estimate the efforts required to develop a program [9]. These are: i) Model-based methods: that predict the demanded effort to develop a new project by analyzing the data from old projects, ii) Expert-based methods: that rely on human expertise to estimate the effort, iii) Hybrid methods: that are combination of the previous methods. Generally, each method consists of two main stages. In the first stage, an estimation regarding the size of a new product is obtained by using sizing techniques, such as function points and lines of code. In the second stage, by converting the estimation of software size into a man-months effort estimation, the time required to develop the software can be estimated [7]. In literature, the problem of combining or adjusting key cost drivers to improve the measurement of software size or complexity are also studied [23]–[27].

Risk management is the identification, evaluation, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability or impact of unfortunate events [15], [28]. To manage the risk involved with multiple cyber attacks, risk management methods are applied in cybersecurity [29], [30]. Recently individual-level epidemic are proposed as a new method to study the spreading dynamics of computer virus [31]–[33]. Individual-level epidemic is a class of epidemic models that use one or a few separate differential equations to characterize the evolution of the expected state of the network in each node. Individual-level epidemic outperforms the population-level [34]–[36] and network-level spreading [37]–[39] models. In the former, each population consists of all the nodes with the same state, while the latter considers the network topology thus some populations are introduced for each possible node degree. Epidemic models have also been applied to other areas, such as disease transmission [40]–[42] and active cyber defense [43]–[46].

In this paper, in order to minimize the network’s expected loss under double-virus attack, we apply the risk management approach to the programmer allocation problem under the individual-level SI model. We model the evolution process of the network’s expected state since the potential economic loss is closely related to the network’s expected state at all time. Our model is similar to the one in [47], which only considers one-virus attack. In contrast, we take the double-virus attack into consideration and propose the DOWNHILL algorithm to solve this tradeoff problem, which is a significant step towards studying more complicated situations and thus protecting the security of the networks against complex security threats.
III. TWO-ANTIVIRUS-PROGRAM DEVELOPMENT PROBLEM

In this section we outline the details of the Two-antivirus-program development (TAPD) problem which is defined as follow:

For a coexistent pair of computer viruses, allocate the limited programmers to different antivirus program projects to achieve the minimum potential total loss.

We first introduce preliminary terminologies and notations in Section III-A. We define the double-virus propagation model in Section III-B. Finally, we model the TAPD problem as an optimization problem in Section III-C.

A. TERMINOLOGIES AND NOTATIONS

Consider a computer network with $N$ nodes denoted as $V = \{1, 2, \cdots, N\}$. The network topology is denoted as $G = (V, E)$ where $\{i, j\} \in E$ represents the communication link between node $i$ and node $j$. $A(G) = (a_{ij})_{N \times N}$ denotes the adjacency matrix of $G$, i.e., $a_{ij} = 1$ or 0 based on where $\{i, j\} \in E$ or not.

Suppose there are two different viruses, Virus I and II, coexisting in the same network at time $t = 0$. We define four states for the nodes in the network at any time $t \geq 0$ which are: i) Susceptible: refers to the node that is not infected with either of the two viruses, ii) I-infected: denotes the node that is infected with Virus I but not infected with Virus II, iii) II-infected: represents the node that is infected with Virus II but not infected with Virus I, and iv) both-infected: means the node that is infected with both two viruses. Let $X_i(t) = 0, 1, 2, 3$ denote that node $i$ is susceptible, I-infected, II-infected, and both-infected at time $t$, respectively. Thus, the state of the network at time $t$ can be represented by the following vector:

$$X(t) = (X_1(t), X_2(t), \cdots, X_N(t))$$

(1)

Let $S_i(t), I^1_i(t), I^2_i(t)$ and $I^3_i(t)$ denote the probability of node $i$ being susceptible, I-infected, II-infected, and both-infected at time $t$, respectively.

$$S_i(t) = \Pr\{X_i(t) = 0\}, \quad I^1_i(t) = \Pr\{X_i(t) = 1\}, \quad I^2_i(t) = \Pr\{X_i(t) = 2\}, \quad I^3_i(t) = \Pr\{X_i(t) = 3\}$$

(2)

As $S_i(t) = 1 - I^1_i(t) - I^2_i(t) - I^3_i(t)$, the expected state of the network at time $t$ can be represented as follow:

$$I(t) = (I^1_1(t), \cdots, I^1_N(t), I^2_1(t), \cdots, I^2_N(t), I^3_1(t), \cdots, I^3_N(t))$$

(3)

Remark 1. $I(0)$ may be estimated through the relevant user reports.

Let the antivirus program I and II denote the programs against Virus I and II, and $T_1$ and $T_2$ denote the complete time of program I and II, respectively. Thus, there is no node infected with Virus I at any time $t \geq T_1$, and no node infected with Virus II at any time $t \geq T_2$. So, for $1 \leq i \leq N$, we have (a) $I^1_i(t) = I^1_i(t) = 0$ for $t \geq T_1$, and (b) $I^2_i(t) = I^2_i(t) = 0$ for $t \geq T_2$.

B. A DOUBLE-VIRUS PROPAGATION MODEL

To characterize the propagation of viruses on the network, we make the following assumptions:

(H$_1$) Given that viruses can propagate in the network, any node $i$ without Virus I may get infected with the virus from an infected neighbor $j$ at the constant rate of $\beta_1$. Thus, the probability of a virus I being propagated to another node at time $t$ is $\beta_1 \sum_{j=1}^{N} a_{ij} \left( I^1_j(t) + I^3_j(t) \right)$

(H$_2$) Any node $i$ without Virus II could get infected with the virus from an infected neighbor $j$ at the constant rate of $\beta_2$. Thus, the probability of a virus II being propagated to another node at time $t$ is $\beta_2 \sum_{j=1}^{N} a_{ij} \left( I^2_j(t) + I^3_j(t) \right)$

The dynamic change in the expected state of the network can be formulated as below:

$$dI^1_i(t) \over dt = \beta_1 \left[ 1 - I^1_i(t) - I^2_i(t) - I^3_i(t) \right] \sum_{j=1}^{N} a_{ij} \left[ I^1_j(t) + I^3_j(t) \right]$$

$$dI^2_i(t) \over dt = \beta_2 \left[ 1 - I^1_i(t) - I^2_i(t) - I^3_i(t) \right] \sum_{j=1}^{N} a_{ij} \left[ I^2_j(t) + I^3_j(t) \right]$$

(4)

C. THE MODELING OF THE TAPD PROBLEM

To measure the expected loss caused by viruses, we introduce the following hypotheses:

(H$_3$) The economic loss per unit time caused by each node only infected with Virus I is $c_1$ USD.

(H$_4$) The economic loss per unit time caused by each node only infected with Virus II is $c_2$ USD.

(H$_5$) The economic loss per unit time caused by each node infected with both viruses is $c_3$ USD, which is the sum of $c_1$ and $c_2$.

(H$_6$) The effort of the antivirus program I is $W_1$ man-days.

(H$_7$) The effort of the antivirus program II is $W_2$ man-days.

(H$_8$) There are totally $n$ programmers available.

Remark 2. The economic loss per unit time, such as $c_1$ and $c_2$, may be estimated by feedbacks collected from users. The effort, such as $W_1$ and $W_2$, may be estimated through project
management approaches. The number of programmers currently available is specified by the company.

Let \( n_1 \) denote the number of programmers assigned to develop antivirus program I, where \( 1 \leq n_1 \leq n - 1 \). Thus, the number of programmers assigned to develop antivirus program II is \( n_2 = n - n_1 \). So, \( T_1 = \frac{W_1}{n_1} \), \( T_2 = \frac{W_2}{n - n_1} \). Combining the above, the expected loss caused by viruses is

\[
\mathcal{L}(n_1) = \int_0^{\max\left\{ \frac{W_1}{n_1}, \frac{W_2}{n-n_1} \right\}} \sum_{i=1}^{N} \left[ c_1 I_1^i(t) + c_2 I_2^i(t) + c_3 I_3^i(t) \right] dt.
\]

Therefore, the TAPD problem is modeled as the following discrete optimization problem.

\[
\min_{1 \leq n_1 \leq n-1} \mathcal{L}(n_1)
\]

\[
= \int_0^{\max\left\{ \frac{W_1}{n_1}, \frac{W_2}{n-n_1} \right\}} \sum_{i=1}^{N} \left[ c_1 I_1^i(t) + c_2 I_2^i(t) + c_3 I_3^i(t) \right] dt.
\]

We refer to this problem as the TAPD model which is characterized by the 9-tuple

\[
\mathcal{M}_{TAPD} = (G, \beta_1, \beta_2, c_1, c_2, W_1, W_2, n, I_0).
\]

IV. TAPD MODEL

In this section we outline TAPD model and propose a solution to solve the problem. As the TAPD model is subject to a complex differential dynamical system, it seems infeasible to solve the model analytically. Now, let us turn our attention to the numerical solution of the TAPD model. First, in Section IV-A we discuss three representative networks that are used in simulation studies. Next, in Section IV-B we present DOWNSHILL algorithm for solving the TAPD model. Finally, in Section IV-C we study the performance of the proposed algorithm and prove that the proposed heuristic algorithm solves the TAPD model.

A. NETWORKS

In this section, we outline the representative networks employed in this paper.

According to empirical research, the majority of the real-world networks can be categorized in two main categories:

i) Scale-free networks that approximately follow power-law degree distributions [18], [19]. We employed Pajek software [48] that produces a synthetic scale-free network \( G_{SF} \) with 100 nodes as which is shown in Fig. 1(a).

ii) Small-world network where each network admit a relatively small diameter [19], [49]. By using Pajek, we get a synthetic small-world network \( G_{SW} \) with 100 nodes, which is shown in Fig. 1(b).

In addition to the above networks, we get a subnet of 100 nodes of the Facebook network \( G_{FB} \), shown in Fig. 1(c).

![Three representative networks](image)

**FIGURE 1**: Three representative networks: (a) a synthetic scale-free \( G_{SF} \), (b) a synthetic small-world \( G_{SW} \), and (c) a subnet of Facebook network \( G_{FB} \).

B. THE DOWNSHILL ALGORITHM FOR SOLVING THE TAPD MODEL

In this section, inspired by [47] we conduct numerical simulations with MATLAB to inspect the optimal solution for the TAPD model. This is necessary to design the DOWNSHILL algorithm.

**Experiment 1. Consider the TAPD model with \( G = G_{SF} \), \( \beta_1 = 0.015, \beta_2 = 0.01, c_1 = 2, c_2 = 3, W_1 = 80, W_2 = 100, n = 60, I(0) = (0.1, \ldots, 0.1) \). We present every \( n_1 \) and its corresponding \( \mathcal{L}(n_1) \) in Fig. 2(a).**

**Experiment 2. Consider the TAPD model with \( G = G_{SW} \), \( \beta_1 = 0.01, \beta_2 = 0.02, c_1 = 3, c_2 = 2, W_1 = 80, W_2 = 100, n = 60, I(0) = (0.1, \ldots, 0.1) \). We present every \( n_1 \) and its corresponding \( \mathcal{L}(n_1) \) in Fig. 2(b).**

**Experiment 3. Consider the TAPD model with \( G = G_{FB} \), \( \beta_1 = 0.03, \beta_2 = 0.01, c_1 = 1, c_2 = 3, W_1 = 100, W_2 = 80, n = 60, I(0) = (0.1, \ldots, 0.1) \). We present every \( n_1 \) and its corresponding \( \mathcal{L}(n_1) \) in Fig. 2(c).**
C. THREE HEURISTIC ALGORITHMS AND A RANDOM ALGORITHM

In this section, we outline the details of three heuristic algorithms and a random algorithm proposed in [32] that are used as baselines to analyze the performance of DOWNHILL.

The first heuristic algorithm is to evenly assign the programmers to the antivirus programs and its computational complexity is \( O(1) \).

\[
\mathbf{n} = \left( \left\lfloor \frac{\pi}{2} \right\rfloor, n - \left( \left\lfloor \frac{\pi}{2} \right\rfloor \right) \right). \tag{8}
\]

We refer to the allocation strategy based on the above algorithm as the uniform (UN) strategy.

The second heuristic algorithm is to assign the programmers to the antivirus program, which is linearly proportional to its effort. The computational complexity of this algorithm is \( O(1) \)

\[
\mathbf{n} = \left( \left\lfloor \frac{\pi w_1}{w_1 + w_2} \right\rfloor, n - \left( \left\lfloor \frac{\pi w_1}{w_1 + w_2} \right\rfloor \right) \right). \tag{9}
\]

We refer to the allocation strategy based on the second heuristic algorithm as the antivirus development effort first (WF) strategy.

The third heuristic algorithm is to assign the programmers to the antivirus program, which is linearly proportional to its infection force. The computational complexity of this algorithm is \( O(1) \).

\[
\mathbf{n} = \left( \left\lfloor \frac{\pi \beta_1}{\beta_1 + \beta_2} \right\rfloor, n - \left( \left\lfloor \frac{\pi \beta_1}{\beta_1 + \beta_2} \right\rfloor \right) \right). \tag{10}
\]

We refer to the allocation strategy based on the third heuristic algorithm as the virus infection force first (IF) strategy.

The random algorithm is to randomly assign the programmers to the antivirus programs and its computational complexity is \( O(1) \).

V. PERFORMANCE EVALUATION FOR DOWNHILL ALGORITHM

This section presents the performance evaluation results based on the simulation output. Assume that \( n_1 \) represent the number of programmers allocated to develop antivirus program I of a specific strategy, and \( L(n_1) \) denote the corresponding total loss of this strategy. Section V-A compares DOWNHILL with the heuristic algorithms, and Section V-B compares DOWNHILL with the random algorithm.

A. COMPARATIVE EXPERIMENTS BETWEEN THE DOWNHILL ALGORITHM AND THE HEURISTIC ALGORITHMS

Experiment 4. Consider a set of TAPD models with \( G \in \{G_{SF}, G_{SW}, G_{FB}\} \), \( \beta_1 = 0.01 \), \( \beta_2 = 0.05 \), \( c_1 = 1 \), \( c_2 = 2 \), \( W_1 = 25 \), \( W_2 = 20 \), \( n = 20 \), \( I_0 = (0.1, \cdots, 0.1) \).
As shown in Fig. 3, the value of $n_1^F$ varies in different allocation strategies. DOWHNILL algorithm has the lowest $L(n_1^F)$ compared to other methods. Based on this and 100,000 other similar simulations with different parameters, we conclude that the DOWHNILL algorithm could lead to the minimum potential total loss compared with the three heuristic algorithms.

**B. COMPARATIVE EXPERIMENTS BETWEEN THE DOWHNILL ALGORITHM AND THE RANDOM ALGORITHM**

**Experiment 5.** Consider a set of TAPD models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 = 0.03$, $\beta_2 = 0.01$, $c_1 = 1$, $c_2 = 3$, $W_1 = 100$, $W_2 = 80$, $n = 100$, $I_0 = (0.1, \cdots, 0.1)$.

In this experiment, let $x_0$ refer to the DOWHNILL strategy and $x_1$ to $x_{100}$ denote the strategies generated by running the random algorithm.

**VI. EFFECT OF DIFFERENT FACTORS ON DOWHNILL STRATEGIES**

In this section, we study the effect of multiple factors on the number of programmers assigned to develop antivirus program I, $n_1$ and the DOWHNILL potential total loss, $L(n_1^F)$, through numerical simulations. We study the impact of changing the infection rate (see Section VI-A), loss per unit of time (see Section VI-B), effort (see Section VI-C), and number of available programmers (see Section VI-D).
A. THE IMPACT OF THE INFECTION RATE

In this subsection, we investigate the impact of different infection rates, $\beta_1$ and $\beta_2$.

**Experiment 6.** (a) Consider a set of TAPD models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 \in \{0.01, 0.02, \ldots, 0.1\}$, $\beta_2 = 0.05$, $c_1 = 1, c_2 = 2, W_1 = 25, W_2 = 20, n = 19, I_0 = (0.1, \ldots, 0.1)$. By running the DOWNHILL algorithm on these TAPD models, we obtain $n_1^P$ and $L(n_1^P)$. We present $n_1^P$ vs. $\beta_1$ and $L(n_1^P)$ vs. $\beta_2$ in Fig. 5(a) and Fig. 5(b), respectively.

(b) Consider a set of TAPD models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 = 0.1$, $\beta_2 \in \{0.025, 0.05, \ldots, 0.25\}$, $c_1 = 1, c_2 = 2, W_1 = 20, W_2 = 25, n = 17, I_0 = (0.1, \ldots, 0.1)$. By performing the DOWNHILL algorithm on these TAPD models, we obtain $n_1^P$ and $L(n_1^P)$. We present $n_1^P$ vs. $\beta_2$ and $L(n_1^P)$ vs. $\beta_2$ in Fig. 5(c) and Fig. 5(d), respectively.

![Figure 5](image)

**FIGURE 5:** The results of Experiment 6: (a) $n_1^P$ vs. $\beta_1$, (b) $L(n_1^P)$ vs. $\beta_1$, (c) $n_1^P$ vs. $\beta_2$, and (d) $L(n_1^P)$ vs. $\beta_2$. As shown $n_1^P$ and $L(n_1^P)$ increase with $\beta_1$, $n_1^P$ decreases with $\beta_2$ and $L(n_1^P)$ increases with $\beta_2$.

From this and 100,000 other similar simulations with different parameters, we conclude that $n_1^P$ is increasing with $\beta_1$ and decreasing with $\beta_2$; and $L(n_1^P)$ is rising with $\beta_1$ and $\beta_2$. This conforms to the intuition.

B. THE IMPACT OF THE LOSS PER UNIT OF TIME

In this subsection, we study the impact of the losses per unit of time, $c_1$ and $c_2$.

**Experiment 7.** (a) Consider a set of TAPD models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 = 0.01, \beta_2 = 0.015, c_1 \in \{1, 2, \ldots, 10\}, c_2 = 5, W_1 = 90, W_2 = 70, n = 40, I_0 = (0.1, \ldots, 0.1)$. By running the DOWNHILL algorithm on these TAPD models, we obtain $n_1^P$ and $L(n_1^P)$. We present $n_1^P$ vs. $c_1$ and $L(n_1^P)$ vs. $c_1$ in Fig. 6(a) and Fig. 6(b), respectively.

(b) Consider a set of TAPD models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 = 0.03, \beta_2 = 0.04, c_2 = 5, c_2 = \in \{1, 2, \ldots, 10\}, W_1 = 110, W_2 = 120, n = 40, I_0 = (0.1, \ldots, 0.1)$. By executing the DOWNHILL algorithm on these TAPD models, we obtain $n_1^P$ and $L(n_1^P)$. We present $n_1^P$ vs. $c_2$ and $L(n_1^P)$ vs. $c_2$ in Fig. 6(c) and Fig. 6(d), respectively.

![Figure 6](image)

**FIGURE 6:** The results of Experiment 7: (a) $n_1^P$ vs. $c_1$, (b) $L(n_1^P)$ vs. $c_1$, (c) $n_1^P$ vs. $c_2$ and (d) $L(n_1^P)$ vs. $c_2$. As shown $n_1^P$ and $L(n_1^P)$ increase with $c_1$, $n_1^P$ decreases with $c_2$ and $L(n_1^P)$ increases with $c_2$.

From this and 100,000 other similar simulations with different parameters, we conclude that $n_1^P$ is increasing with $c_1$ and decreasing with $c_2$; and $L(n_1^P)$ is rising with $c_1$ and $c_2$. This conforms to the intuition.

C. THE IMPACT OF EFFORT

In this subsection, we inspect the impact of effort, $W_1$ and $W_2$.

**Experiment 8.** (a) Consider a set of TAPD models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 = 0.03, \beta_2 = 0.05, c_2 = 1, W_1 \in \{10, 20, \ldots, 100\}, W_2 = 50, n = 25, I_0 = (0.1, \ldots, 0.1)$. By executing the DOWNHILL algorithm on these TAPD models, we obtain $n_1^P$ and $L(n_1^P)$. We present $n_1^P$ vs. $W_1$ and $L(n_1^P)$ vs. $W_1$ in Fig. 7(a) and Fig. 7(b), respectively.

(b) Consider a set of TAPD models with $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 = 0.015, \beta_2 = 0.025, c_2 = 1, W_1 = 50, W_2 \in \{10, 20, \ldots, 100\}, n = 25, I_0 = (0.1, \ldots, 0.1)$. By running the DOWNHILL algorithm on these TAPD models, we obtain $n_1^P$ and $L(n_1^P)$. We present $n_1^P$ vs. $W_2$ and $L(n_1^P)$ vs. $W_2$ in Fig. 7(c) and Fig. 7(d), respectively.
The results of Experiment 9: (a) $n^P_1$ vs. $W_1$, (b) $\mathcal{L}(n^P_1)$ vs. $W_1$, (c) $n^P_2$ vs. $W_2$ and (d) $\mathcal{L}(n^P_2)$ vs. $W_2$. As shown $n^P_1$ and $\mathcal{L}(n^P_1)$ increase with $W_1$, $n^P_2$ decreases with $W_2$ and $\mathcal{L}(n^P_2)$ increases with $W_2$.

From this and 100,000 other similar simulations with different parameters, we conclude that $n^P_1$ is increasing with $W_1$ and decreasing with $W_2$; and $\mathcal{L}(n^P_1)$ is rising with $W_1$ and $W_2$. This conforms to the intuition.

D. THE IMPACT OF THE NUMBER OF AVAILABLE PROGRAMMERS

In this subsection, we inspect the impact of the number of available programmers.

**Experiment 9.** Consider a set of TAPD models in which $G \in \{G_{SF}, G_{SW}, G_{FB}\}$, $\beta_1 = 0.01$, $\beta_2 = 0.02$, $c_1 = 3$, $c_2 = 2$, $W_1 = 70$, $W_2 = 90$, $n \in \{10, 20, \ldots , 100\}$, $I_0 = (0.1, \ldots , 0.1)$. By running the DOWNHILL algorithm on these TAPD models, we obtain $n^P_1$ and $\mathcal{L}(n^P_1)$. We present $n^P_1$ vs. $n$ and $\mathcal{L}(n^P_1)$ vs. $n$ in Fig. 8(a) and Fig. 8(b), respectively.

From this and 100,000 other similar simulations with different parameters, we conclude that $n_1$ is increasing with $n$; and $\mathcal{L}(n^P_1)$ is decreasing with $n$. This conforms to the intuition.

**VII. CONCLUSION**

In the framework of risk management, this paper studied the computer virus response problem. On this basis, we modeled the problem of tackling with new viruses as a two-antivirus-program development problem. The latter is defined as how to allocate the limited programmer resources to develop different antivirus programs to achieve the minimum potential total loss (caused by viruses). Through modeling and analysis of the time evolution of the network’s expected state, we quantified the network’s potential loss caused by Virus I and II, respectively. Then, we obtained the potential total loss by adding them together. We presented DOWNHILL algorithm to solve this problem. The simulation results prove that the proposed DOWNHILL algorithm achieves better performance as compared to three heuristic algorithms and the random algorithm. We also studied the impact of multiple factors on DOWNHILL strategy. To the best of our knowledge, this work is the first work that studies the double-virus tradeoff problem.

As future research directions, the model can be extended to more sophisticated virus spreading models, such as the impulsive spreading, the stochastic spreading, and the spreading models on time-varying networks. Artificial intelligence algorithms can also be applied to solve the same problem studied in this paper [50]–[52]. The social and juristic obstacles in solving the problem also can be studied [53].

**REFERENCES**


8 VOLUME 4, 2016


[33] FANGFEI ZHANG received the M.S. degree in informatics from Northeastern University, USA, in 2017, the B.S. degree from Shandong University of Finance and Economics, China, in 2014. She is currently pursuing the Ph.D. degree with the school of big data and software engineering, Chongqing University, China. Her research interests include epidemic dynamics, cybersecurity and smart grid.

JICHAO BI received the B.S. degree from Shenyang normal university in 2014. He is currently pursuing the Ph.D. degree with the school of big data and software engineering, Chongqing University, China. His research interests include epidemic dynamics, cybersecurity and smart grid.
ALI DORRI is a Research Fellow at Queensland University of Technology (QUT), Brisbane, Australia. He received his Ph.D. degree from the University of New South Wales (UNSW), Sydney, Australia. He was also a Postgraduate research student at CSIRO, Australia. His research interest includes blockchain applications and challenges in adopting blockchain in large scale networks including the Internet of Things, smart cities, smart grid, and e-health. He has published over 20 peer-reviewed papers.

CHUNMING ZHANG received the B.S. degree from the School of Computer, Shaanxi University of Science and Technology, in 2005, and the Ph.D. degree from the College of Computer Science, Chongqing University, in 2012. He is currently an Associate Professor of computer science at Guangdong Medical University. He has published over 30 papers in peer-reviewed international journals. His research interests include computer virus spreading and cybersecurity.

CHEN ZHANG received the B.S. degree from Xidian University in 2013, and received the M.S. degree from Zhejiang University in 2016. His research interests include cloud computing and information security.