Predators: Good Will Mobile Codes Combat against Computer Viruses

Hiroshi Toyoizumi
Performance Evaluation Lab.
U. of Aizu
Aizu-wakamatsu
Fukushima, Japan
toyo@u-aizu.ac.jp

Atsuhi Kara
CITEC
U. of Aizu
Aizu-wakamatsu
Fukushima, Japan
kara@u-aizu.ac.jp

ABSTRACT

We present a mathematical analysis of a new approach to fight against computer viruses through the use of their predators. Predators are good will mobile codes which, like viruses, travel over computer networks, and replicate and multipy themselves. The only difference is that predators are specifically designed to eliminate the viruses. We model the interaction between predators and viruses by the Lotka-Volterra equations, which are widely used in mathematical biology. Using this model, we derive a method to constrain the number of predators to be as few as possible, while maintaining their power to eliminate viruses.

Keywords

computer virus, worms, mathematical biology, Lotka-Volterra equation

1. INTRODUCTION

Malicious mobile codes, known as computer viruses or worms, have become a siginificant social problem recently[17, 4]. To protect against computer viruses, there are many commercial anti-virus applications that we can install on our machines. When the anti-virus application detects a virus on the machine, it eliminates the virus. However, we need the latest update file, which lists all known virus patterns, to protect against the new viruses [1, 16]. Unfortunately, not all machines are equipped with the anti-virus application and the latest update file. Furthermore, it may take some time for users to install the latest update file. As shown in the outbreak of Code-red [2, 11] and Nimda [3], viruses with strong infection power will dominate the network within 24 hours. Actually, on July 19, 2001, within 14 hours of the debut of its first copy, Code-red virus infected more than 359,000 machines, at a rate of 2,000 machines per minute at its peak [12].

Since many of the anti-virus applications are of server-client type, the server providing the latest updates can become a bottleneck if many users try to get the latest files simultaneously, which may happen especially when there is an intensive epidemic of a new and unknown virus on the network. Moreover, the server itself can be the target of a Denial of Service (DoS) attack. In any event, it is doubtful that all users install anti-virus software, and this problem will be further compounded when all machines in the home have constant access to the Internet. To avoid having install anti-virus applications on all machines, we may install an anti-virus application on only network elements such as gateways or mailservers. This will avoid the installation problem, but once a virus penetrates into the local network, the advantage will disappear. Recently, some researchers have suggested the possibility of sending a vaccine to each machines via the same penetration method of a particular virus [8, 4]. We will extend their idea to the concept of predators, which remove viruses and replicate themselves. Since it is important to control the number of predators on the network without reducing their ability to eliminate viruses, we construct a mathematical model of the interaction between predators and viruses. There has been many such proposed mathematical models of computer viruses. For examples, in [17], viruses are discussed in the setting of computer science, whereas in [7], viruses are treated as biological objects in the natural world. In [9], the authors study the real epidemic of computer viruses. Even the spread of Code-red virus is discussed using mathematical models in [13, 15]. In this paper, following the discussion in [6], we formulate a mathematical model based on the Lotka-Volterra equations. The Lotka-Volterra equations are known to be a simple yet quite strong tool for analyzing the charcteristics of predator-prey relationships in the natural world. Motivated by an analysis of the orbit of the dynamical system of the Lotka-Volterra equations, we propose a method to select important parameters of predators, such as the optimal rates of multiplication and predation.

New Security Paradigms Workshop '02, September 23-26, 2002, Virginia Beach, Virginia.

2. VECTOR METHOD

A good will mobile code may deliver a vaccine to machines through the same mechanism as viruses. This method is known as the vector method [8, 4]. The vaccine for a virus can spread throughout the network, by penetrating machines both infected or not yet infected. However, if the good will mobile code has the strong infectiousness, the network will be flooded with such vaccines, which eventually consume precious network resources. This degradation of the network is not tolerable, especially when the network has no infected machines. Also, even though this mobile code is not intended to do any harm,

it may still cause damage to some machines. Thus, the spread of good will mobile codes should be carefully controlled.

3. PREDATOR METHOD

We introduce predators to overcome the disadvantages of vectors described in the previous section. A predator is also a good will mobile code, which has the ability to search for and eliminate viruses, as well as replicate itself, as follows:

1. Searching Viruses. By looking for the traces of the virus or the particular packets emitted when the virus tries to mulitply, a predator searches places where a virus is likely to hide.

2. Eliminating viruses. When the predator finds a virus on a machine, the predator will enter the machine in the same way as the virus. After successfully penetrating the machine, the predator eliminates (or eats!) the virus. The predator may even take an appropriate security measure, such as installing a patch for the security hole used by the virusor even the halls that may potentially be used by viruses in the future.

3. Multiplication.
   (a) The predator who ate the virus then enters a multiplication phase. During this period, a predator replicates itself a number of times determined by the random number B, and sends the copies to other randomly selected machines via the same method as the viruses.
   (b) When the multiplication phase is completed, the predator resumes searching for food.

Using these features, predators can be constrained to replicate themselves only in the environment where viruses can breed, but to cease multiplying when there are no viruses present. Thus, we will be able to eliminate viruses while minimizing the effect of the predators on the environment.

Remark 1. *The idea of predator is not quite new. Some researchers coded a predator for Code-Red, and security professionals debated the pragmatic and ethical issues involved in releasing a predators. See the references in [10]. In this paper, we show how we can build predators effectively in the following. Ethical and leagal issues would be addressed also, but we will not approach them in this paper.*

4. MATHEMATICAL MODEL FOR COMPUTER VIRUS

The main difference between our predators and the ones found in the natural world is that ours can be controlled by specifying the multiplication rate. In the following, we will investigate the mathematical biological model of virus multiplication to find the optimal parameters for predators to eliminate viruses with maximum efficiency. First, we will show how existing biological models (see, e.g., [6, 5, 14]) are used to describe the multiplication process of malicious computer viruses.

4.1 Multiplication Process of Computer Viruses

First, we consider the case when there are only viruses on the network. Let $x(t)$ be the number of machines infected by a virus at time t. Then, $x(t)$ will satisfy the following logistic differential equation:

$$\frac{dx}{dt} = rx(1 - \frac{x}{K})$$
$$x(0) = x_0, \qquad (1)$$

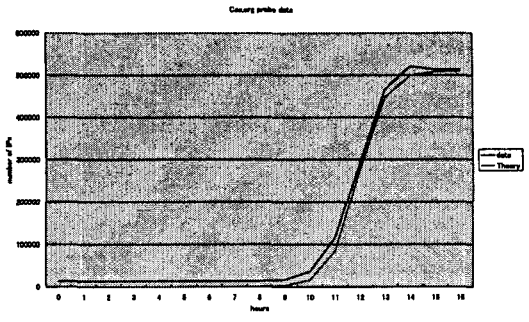

Figure 1: Number of Code-red detection per unit hour as a function of time, as compared to the logarithmic function with parameters $K = 510,000$ and $r = 1.8$/hour. Graph from Staniford [15].

where x_0 is the initial number of machines infected, r is the (intrinsic) natural multiplication rate, and K is environmental capacity. The factor $r(1-x/K)$ represents the infection rate, that is, the number of secondary infections per virus per unit time. The number of infected machines $x(t)$ increases when $x(t)$ is smaller than the environmental capacity K, but the rate of the infection decreases as $x(t)$ approaches K. It is wellknown that the solution to the logistic equation (1) is the logistic curve,

$$x(t) = \frac{K}{1 + (K/x_0 - 1)exp(-rt)}. \qquad (2)$$

Indeed, it has been reported that the actual multiplication process of Code Red virus satisfied a logistic curve [13, 15] (see Figure 1).

Also, when $x(t)$ is sufficiently small, equation (1) can be approximated by an exponential growth model,

$$\frac{dx}{dt} = rx$$
$$x(0) = x_0, \qquad (3)$$

where r represents the viral infection rate. The solution of (3) is $x(t) = x_0 e^{rt}$, that is, initially, there is an exponential outbreak of the virus.

4.2 Interaction between Viruses and Predators

The interaction of viruses and predators can be modeled by the Lotka-Volterra system, which is widely used in mathematical biology (see [6], p.35). Let $x(t)$ be the number of machines infected by a virus at time t and $y(t)$ be the number of predators at time t. We assume the vector $(x(t), y(t))$ satisfies the following system of

differential equations:

$$\frac{dx}{dt} = rx - axy$$
$$\frac{dy}{dt} = bxy$$
$$(x(0), y(0)) = (x_0, y_0), \tag{4}$$

where r is the viral multiplication rate. The virus not only multiplicates, but also is eliminated by predators. The rate of elimination is proportional to the population of both viruses and predators, and is given by axy, where a is the predatory rate. On the other hand, the multiplication of predators will be affected by the virus and predator population, and is given by bxy, where b is the predator multiplication rate. Also, we set (x_0, y_0) to be the initial state of the system.

Remark 2. *According to the discussion in Section 3, the multiplication rate b is given by*

$$b = aE[B], \tag{5}$$

where B is a random variable that represents the number of multiplications permitted for the predator who eliminates a virus.

In case of the Lotka-Volterra equations (4), we cannot obtain their explicit solution. However, we can use a conserved quantity to analyze the orbits of the system in xy plane.

Lemma 1 (Conservation). *Let $V(x,y) = bx - r\log y + ay$. Then there exists a constant independent of x and y satisfying*

$$V(x,y) = C. \tag{6}$$

PROOF. Differentiating $V(x,y)$ with respect to t and applying (4), we have

$$\begin{aligned}\frac{d}{dt}V(x,y) &= b\frac{dx}{dt} - \frac{r}{y}\frac{dy}{dt} + a\frac{dy}{dt} \\ &= b(rx - axy) + bxy(a - r/y) \\ &= 0.\end{aligned}$$

Before discussing in detail the orbits in xy plane, we need to recall the definition of the product logarithm functions. Consider the following implicit equation for w,

$$z = we^w \tag{7}$$

(see Figure 2). For $-e^{-1} \le z < 0$, (7) has two real solutions w_{-1}, w_0, with $w_{-1} \le w_0$, called the product logarithm functions and denoted by

$$w_{-1} = \mathrm{plog}_{-1}(z)$$
$$w_0 = \mathrm{plog}_0(z).$$

Note that if we extend the product logarithm functions to imaginary numbers, they correspond to branches of the standard logarithm function: the value w_0 is the primary branch and w_{-1} is the (-1)-st branch. Also, it is not difficult to derive the following properties for the product logarithm functions.

Lemma 2 (Product Logarithm Functions). *For $-e^{-1} \le z < 0$, (1) the product logarithm functions are real, (2) $\mathrm{plog}_{-1}(z) \le -1 \le \mathrm{plog}_0(z)$, and (3) $\mathrm{plog}_{-1}(z)$ is decreasing with respect to z.*

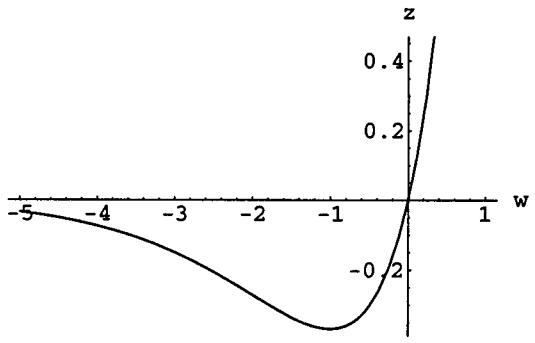

Figure 2: Curve $z = we^w$.

Through the use of the product logarithm functions, the virus-predator orbit can be estimated.

Theorem 1 (Orbit of the Predator Model). *Let y_T be the earliest time by which all viruses have been eliminated. Then we have*

$$y_T = -\frac{r}{a}\mathrm{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r - bx_0/r}\right). \tag{8}$$

Also, in the xy-plane, the orbit of the system (4) is

$$(x(y), y) = \left(\frac{1}{b}\{r\log(y/y_0) - a(y - y_0)\} + x_0, y\right), \tag{9}$$

for $y \in [y_0, y_T]$.

PROOF. First we prove (9). From Lemma 1, there exits a constant C independent of x and y such that

$$V(x,y) = bx - r\log y + ay = C.$$

Evaluating this function for the initial condition (x_0, y_0) gives $C = -r\log y_0 + ay_0 + bx_0$, and thus

$$bx - r\log y + ay = -r\log y_0 + ay_0 + bx_0. \tag{10}$$

Solving (10) with respect to x, we obtain the function $x(y)$ given in (9). Next we prove (8) by showing that y_T is well defined, $y_T \ge y_0$, and $x(y_T) = 0$. Set $z = -\frac{ay_0}{r}e^{-ay_0/r - bx_0/r}$. By Lemma 2, y_T is real if $-e^{-1} \le z < 0$. In general, since $e^{-1} \ge se^{-s}$ for $s \ge 0$, substituting $s = ay_0/r$ yields

$$\begin{aligned}-e^{-1} &\le -\frac{ay_0}{r}e^{-ay_0/r} \\ &\le -\frac{ay_0}{r}e^{-ay_0/r - bx_0/r} \\ &= z < 0.\end{aligned} \tag{11}$$

Thus, y_T is real and well defined. Now we will show $y_T \ge y_0$. Since the product logarithm function is decreasing (Lemma 2), we have

$$\begin{aligned}y_T &= -\frac{r}{a}\mathrm{plog}_{-1}(-\frac{a}{r}y_0 e^{-ay_0/r - bx_0/r}) \\ &\ge -\frac{r}{a}\mathrm{plog}_{-1}(-\frac{a}{r}y_0 e^{-ay_0/r}).\end{aligned}$$

By the definition of the product logarithm functions, if $-ay_0/r > -1$,

$$\begin{aligned}-\frac{r}{a}\mathrm{plog}_{-1}(-\frac{a}{r}y_0 e^{-ay_0/r}) &\ge -\frac{r}{a}\mathrm{plog}_0(-\frac{a}{r}y_0 e^{-ay_0/r}) \\ &= y_0.\end{aligned}$$

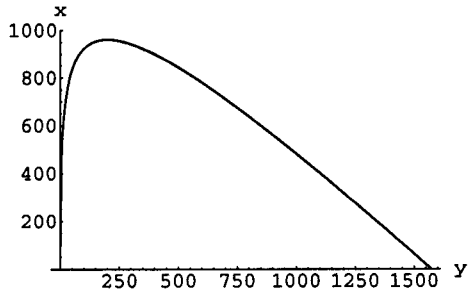

Figure 3: An orbit in virus-predator space. $(x_0, y_0) = (100, 1), r = 2, a = 1/100, b = 1/100$.

On the other hand, if $-ay_0/r \leq -1$,

$$-\frac{r}{a}\text{plog}_{-1}(-\frac{a}{r}y_0 e^{-ay_0/r}) = y_0.$$

In either case, we have $y_T \geq y_0$. Finally, we will show $x(y_T) = 0$, that is at y_T all viruses have been eliminated. From (8),

$$-\frac{a}{r}y_T = \text{plog}_{-1}(-\frac{a}{r}y_0 e^{-ay_0/r - bx_0/r}).$$

By the definition of the product logarithm functions, we have

$$y_0 e^{-ay_0/r - bx_0/r} = y_T e^{-ay_T/r}.$$

Taking logarithm of both sides yields

$$\log y_0 - ay_0/r - bx_0/r = \log y_T - ay_T/r,$$

and thus,

$$x(y_T) = \frac{1}{b}(r\log(y_T/y_0) - a(y_T - y_0)) + x_0 = 0.$$

Remark 3. *Let $y_{max} = r/a$. If $y_0 < y_{max}$, the population of the virus $x(y)$ increases for $y \in [y_0, y_{max}]$, reaches its maximum at y_{max}, and then decreases for $y > y_{max}$. If $y_0 \geq y_{max}$, $x(y)$ is monotone decreasing.*

4.3 Examples of the Orbit

Figure 3 illustrates an example of changing virus and predator populations. We assume that the predatory rate and multiplication rate of predator are significantly smaller than the viral multiplication rate, since as compared to the multiplication of viruses, it is not so easy for predators to find viruses on the network.

Initially there are not that many predators on the network, so the virus population increases along with the number of predators, as pointed out in Remark 3. However, once the predator population exceeds y_{max}, the viruses are captured by the predators at a faster rate than the viruses can multiply, and thus the virus population begins to decrease until finally it has been extinguished at time y_T. Also, the difference between Figures 1 and 3 reveals the impact of predators.

The Lotka-Volterra equations for viruses and predators involves several parameters: (1) the initial condition (x_0, y_0), (2) the virus multiplication rate r, (3) the predatory rate a, and (4) the predator multiplication rate b. In the following subsections, we will see how varying these parameters affects the orbits of the Lotka-Volterra equations.

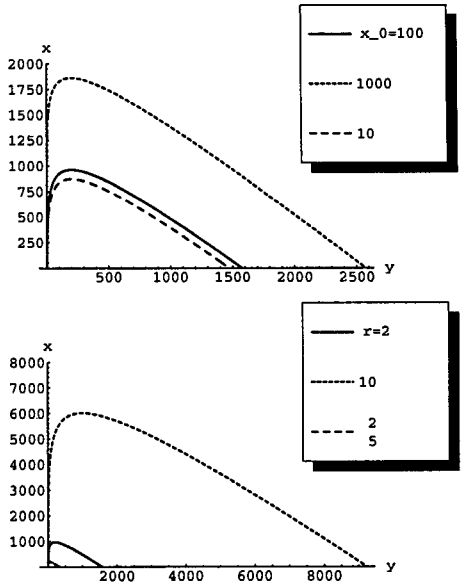

Figure 4: Orbits for various initial virus populations (upper figure) and virus multiplication rates (lower figure), with other parameters fixed as $y_0 = 1, r = 2, a = 1/100, b = 1/100$.

4.3.1 Initial Condition and Virus Multiplication Rate

Figure 4 demonstrates the impact on the orbit when we vary the initial virus population, and also when we vary the virus multiplication rate. The shape of orbit is quite similar to Figure 3. However, we can see that if the initial virus population is large, more predators are required to extinguish the viruses. Hence, it is important to act as soon as possible after a virus is detected.

Moreover, when the virus multiplication rate is large, the virus population increases more rapidly than it does for the case of a large initial virus population. Thus, those viruses with strong multiplication ability will be still more terrible.

4.3.2 Predatory Rate and Predator Multiplication Rate

Figure 5 shows the effect of varying the predatory rate and the predator multiplication rate. A large predatory rate can restrain virus population growth, as well as predator population growth; thus, predators should be manufactured so as to have the largest possible predatory rate.

When the predator multiplication rate is large, the virus can increase only during the initial stage. However, a serious outbreak of predators will result if their multiplication rate is too large. Thus, we need to control the predator multiplication rate carefully. In the following section, we discuss this matter in more detail.

5. DESIGNING PREDATORS

Unlike in the natural world, we can design predators that are perfect for fighting computer viruses. When designing the predator, particular care must be taken in choosing the predatory rate a and the multiplication rate b. For example, if b is very large, viruses can

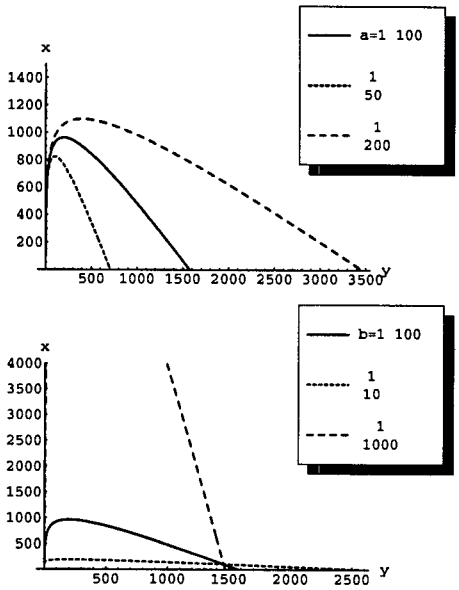

Figure 5: Orbits for various predatory rates (upper figure) and predator multiplication rates (lower figures) with other parameters fixed as $(x_0, y_0) = (100, 1), r = 2$.

be eliminated rapidly, but at the same time there will be an outbreak of predators, degrading the network. On the other hand, a smaller b will allow viruses to dominate the network.

5.1 Predatory Rate

By Theorem 1, the number of viruses can be expressed as

$$x(y) = \frac{1}{b}\{r\log(y/y_0) - a(y-y_0)\} + x_0.$$

Since $y \geq y_0$, $x(y)$ is decreasing with respect to a for all y. Thus, as seen in Section 4.3.2, the predatory rate should be as large as possible.

5.2 Multiplication Rate of Predator

Assume the predatory rate a is fixed to be as large as possible. We seek to determine an optimal predator multiplication rate b. If we suppose that the side effect of predators on the network is the same as that of viruses, then our objective will be to minimize the maximum of the combined total population of viruses and predators. Define the $L^\infty[0,\infty)$ norm for the total population of viruses and predators by

$$\|x+y\|_\infty = \sup_{t\in[0,\infty)} (x(t)+y(t)), \tag{12}$$

and let b^* be the optimal multiplication rate that minimizes $\|x+y\|_\infty$. We will seperately derive the optimal solution for different values of the initial predator population y_0.

5.2.1 *Large* y_0

First, assume $y_0 > r/a$. Since $x(t)$ is decreasing with respect to t for any b, the optimal multiplication rate is $b^* = 0$. In other words, it is not necessary for the predator to have the ability to multiply.

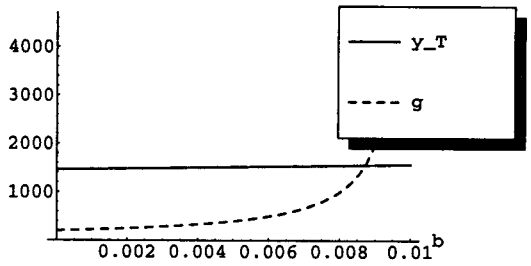

Figure 6: $y_T(b)$ and $g(b)$ have a unique intersection on $[0,a)$. $(x_0, y_0) = (100, 1), r = 2, a = 1/100$.

Theorem 2 (Optimal multiplication rate 1). *If $y_0 > r/a$, then $b^* = 0$, that is, no multiplicative ability for predators is required.*

5.2.2 *Small* y_0

Assume $y_0 \leq r/a$. If we let $f(y) = x(y) + y$, then $f(y_T) = y_T$ and

$$\|x+y\|_\infty = \sup_{y\in[y_0,y_T]} f(y). \tag{13}$$

Since f is continuous, the supremum of (13) is attained in $[y_0, y_T]$. Now, we will investigate this maximum in considerable detail. From Theorem 1,

$$f(y) = \frac{1}{b}\{r\log(y/y_0) - a(y-y_0)\} + x_0 + y. \tag{14}$$

It is easy to see that $f'(y) = 0$ at $y = r/(a-b)$, and that f is concave. Thus, we need to determine whether y_T or $y = r/(a-b)$ is larger.

Consider y_T as a function of b, denoted by $y_T(b)$, and let $g(b) = r/(a-b)$. We will compare the two functions $y_T(b)$ and $g(b)$ (see Figure 6).

Lemma 3. *The two functions $y_T(b)$ and $g(b)$ have a unique intersection on $[0,a)$, that is, there exists a unique $b_1 \in [0,a)$ such that*

$$y_T(b_1) = g(b_1). \tag{15}$$

PROOF. Differentiating $y_T(b)$ twice, we have

$$y_T''(b) = \frac{-x_0^2 \mathrm{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r - bx_0/r}\right)}{ar\{1+\mathrm{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r-bx_0/r}\right)\}^3} < 0.$$

Thus $y_T(b)$ is concave. Also, since $y_T(0) \geq r/a = g(0)$ and $g(b) \to \infty$ as $b \to a$, $y_T(b)$ and $g(b)$ have a unique intersection on $[0,a)$.

Lemma 4. *If $y_0 \leq r/a$,*

$$\|x+y\|_\infty = \begin{cases} f(r/(a-b)) & b \in [0, b_1] \\ y_T & b \in [b_1, \infty). \end{cases} \tag{16}$$

PROOF. Note that $f(y)$ is concave. If $b \leq b_1$, $g(b) = r/(a-b) < y_T(b)$. By assumption, $y_0 \leq r/a \leq r/(a-b)$, Hence, $r/(a-b) \in [y_0, y_T(b)]$, and the maximum of $f(y)$ is $f(r/(a-b))$. If $b > b_1$, $r/(a-b) > y_T(b)$. Thus, $f(y)$ is increasing in $[y_0, y_T]$, and its maximum is $f(y_T) = y_T$.

Theorem 3 (Optimal Multiplication Rate 2). *If we can pump predators into the network only as much as $y_0 \le r/a$, then the optimal predator multiplication rate b^*, that minimizes $\|x+y\|_\infty$ is obtained by*

$$b^* = \frac{a[1+\text{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r}\right)]}{\text{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r}\right)}. \tag{17}$$

PROOF. Since $y_T(b)$ is increasing, $y_T(b) \ge y_T(b_1)$ for $b \ge b_1$. Furthermore from (15), $f(r/(a-b_1)) = f(y_T(b_1)) = y_T(b_1)$, and thus, for $b \ge b_1$, $f(r/(a-b_1)) \le y_T(b)$. By Lemma 4, we know that b^* should be at most b_1. Now we find $b \in [0, b_1]$ that minimizes $f(r/(a-b))$. If $h(b) = f(r/(a-b))$, then we have

$$h'(b) = \frac{a(r-(a-b)y_0) - r(a-b)\log(r/((a-b)y_0))}{b^2(a-b)}. \tag{18}$$

Thus, solving $h'(b) = 0$ yields

$$\frac{a}{a-b}r - ay_0 = r\log(r/(a-b)) - r\log(y_0).$$

Rearrange this equation to obtain

$$-\frac{ay_0}{r}e^{-ay_0/r} = -\frac{a}{a-b}e^{-a/(a-b)}.$$

Now recalling (11) and using the definition of plog_{-1}, we have

$$\frac{a}{a-b} = -\text{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r}\right).$$

Thus, b^* defined by (17) satisfies $h'(b) = 0$. Also, by using similar arguments, it can be shown that $h'(b) < 0$ for $b > b^*$ and $h'(b) > 0$ for $b < b^*$, hence b^* minimizes $h(b)$. Finally, it remains to check that $b^* \le b_1$. From Lemma 3, we have

$$\begin{aligned}\frac{a}{a-b_1} &= -\text{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r-b_1x_0/r}\right)\\ &\ge -\text{plog}_{-1}\left(-\frac{ay_0}{r}e^{-ay_0/r}\right)\\ &= \frac{a}{a-b^*} \ge 0.\end{aligned}$$

Thus, $b_1 \ge b^*$.

Remark 4. *Generally, the number of machines infected by a virus is unknown. However, since b^* in (17) is independent of the initial virus population, we can design predators by analyzing the behavior of an individual virus.*

Now, we will investigate the optimal predator multiplication rate given in Theorem 3. In Figure 7, we compare the orbit for the optimal b^* with other orbits for nonoptimal b. We note a viral outbreak for the smaller predator multiplication rate, whereas for the predator multiplication rate that is larger than b^*, the population of the predator continues to grow even while the virus population decreases.

Figure 8 shows how varying the virus multiplication and predatory rates affects b^*. Even for higher virus multiplication rates, b^* is relatively stable. The predator multiplication rate should be $b = 0.01$ up to $r = 10$. Also, it can be said that the predator multiplication rate should be chosen based on the predatory rate a.

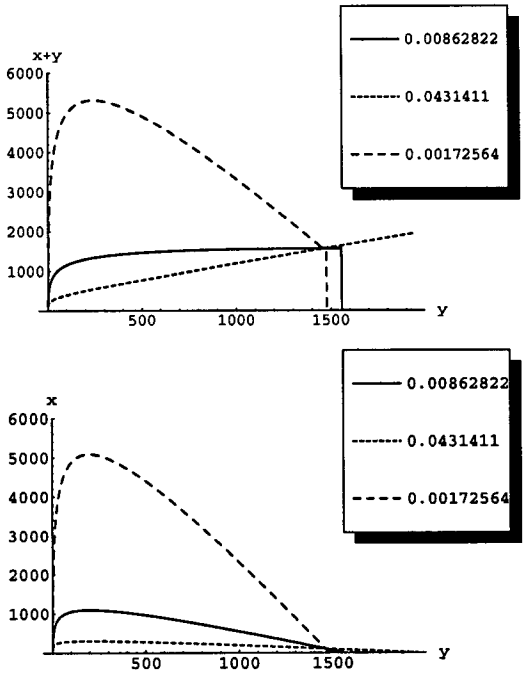

Figure 7: Orbits for optimal predator multiplication rate $b^* = 0.0086$ compared with those for $b = 5b^*, b^*/5$ with $(x_0, y_0) = (100, 1), r = 2, a = 1/100$. The upper figure is the graph of $f(y) = x(y) + y$ and the lower is the xy orbit.

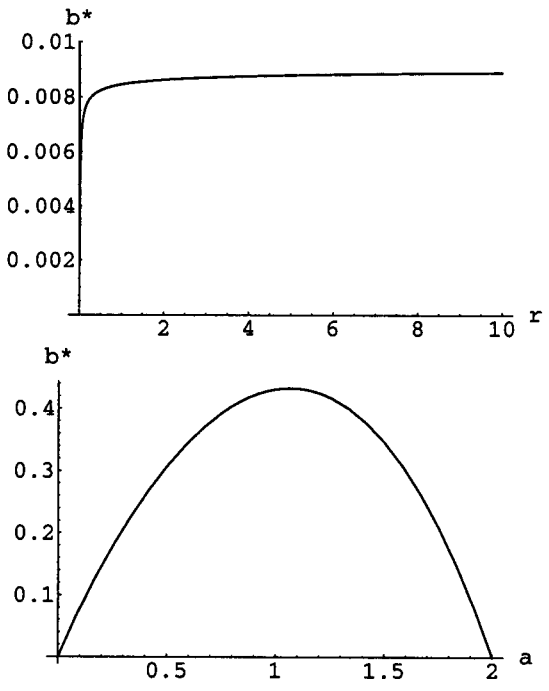

Figure 8: Upper figure: virus multiplication rate r versus the optimal predator multiplication rate b^* with $y_0 = 1, a = 1/100$. Lower figure: predatory rate a versus the optimal predator multiplication rate b^* with $y_0 = 1, r = 2$.

6. EXAMPLE OF PREDATOR FOR CODE-RED

Suppose we were able to build a predator for Code-red on July 19, 2001. Since the Code-red virus has the multiplication rate $r = 2$ (new infections/hour) as shown in Figure 1, the optimal multiplication rate for its predators would be $b = 1/100$, given that the predatory rate $a = 1/100$. In this case, the total number of viruses and predators could have been limited to as few as 2,000 in the whole world, as shown in Figure 8, instead of 359,000 machines.

7. REFERENCES

[1] Network Associates. McAfee. http://www.nai.com/japan/.

[2] CAIDA. CAIDA analysis of code-red. http://www.caida.org/analysis/security/code-red/.

[3] CERT/CC. CERTR advisory CA-2001-26 nimda worm, September 2001.

[4] R. A. Grimes. *Malicious Mobile Code*. O'Reilly and Associates, 2001.

[5] Stanley I. Grossman and William R. Derrick. *Introduction to Differential Equations With Boundary Value Problem*. Longman, January 1999.

[6] You Iwasa. *Mathematical Biology*. Kyouritsu, 1999.

[7] S. Jones and C. White. The ipm model of computer virus management. *Computers and Security*, 9(5):411–418., 1990.

[8] Atsuhi Kara. On the use of intrusion technologies to distribute non-malicious programs to vulnerable computers. Technical report, University of Aizu, 2001.

[9] Jeffrey O. Kephart, Steve R. White, and David M. Chess. Computers and epidemiology. *IEEE Spectrum*, pages 20–26, MAY 1993.

[10] Brian McWilliams. New worms seek and destroy code red. http://www.commoncriteria.org/news/newsarchive/Sept01/sept02.htm, Sep 2001.

[11] Carolyn Meinel. Code red for the web. *Scientific American*, pages 36–43, October 2001.

[12] David Moore. The spread of the code-red worm (CRv2), July 2001.

[13] Security.NL. Code red worm stats. http://www.security.nl/misc/codered-stats/, 2001.

[14] Karl Sigmund and Josef Hofbauer. *The Theory of Evolution and Dynamical Systems*. Cambridge University Press, 1988.

[15] Stuart Staniford. Analysis of spread of july infestation of the code red worm. http://www.silicondefense.com/cr/.

[16] Symantec. http://www.symantec.co.jp/.

[17] Harold Thimbleby, Stuart Anderson, and Paul Cairns. A framework for modelling Trojans and computer virus infection. *The Computer Journal*, 41(7):445–458, 1998.