ウイルス捕食プロセスを用いたコンピュータウイルス対策シミュレーション

小牧嵩征* 森口一郎**

ウイルス対策ソフトで各コンピュータに免疫を与える方法では、インターネット上からウイルスを絶滅することは困難である。これに替わる手法として、ネットワーク上のウイルスを削除してまわるプログラムをネットワーク上に放つ手法を提案し、この手法の有効性とどのような挙動を起こすのかをシミュレーションで検証した。この結果、捕食プログラムを自己消滅するようにした場合、捕食プログラムの寿命を長くしない限りウイルスの絶滅は困難なことを明らかにした。また、インターネットのようなネットワーク構造を持つ場合、各ノードのリンク数が一様なネットワークに比べ、捕食プログラムがネットワーク上に広がりやすいことがわかった。

キーワード：SIモデル，random network，scale-free network，ウイルス対策，捕食-被食関係

Anti-virus simulation using predator process

Takayuki KOMAKI and Ichirou MORIGUCHI

It has been shown that the anti-virus strategy immunizing each computer is difficult to exterminate computer viruses in the Internet. In this study, we proposed a new anti-virus strategy that releases the programs which move around in the Internet and remove viruses, and investigated the effectiveness of this strategy by simulation. In case that the anti-virus programs disappear of themselves, it was shown that exterminating computer viruses is difficult. Futhermore, it was also shown that moving anti-virus programs easily diffuse on heterogeneous networks than on homogeneous networks.

Keyword：SI model, random network, scale-free network, anti-virus, predator-prey relationships

*東京情報大学 総合情報学部 情報システム学科（2011年3月卒業），2011年4月より北陸先端科学技術大学院大学知識科学研究科に所属
Tokyo University of Information Sciences, Faculty of Informatics, Department of Information SystemSystems（graduation in March, 2011）
**東京情報大学 総合情報学部 情報システム学科
Tokyo University of Information Sciences, Faculty of Informatics, Department of Information Systems

2011年6月16日受理

1．はじめに

現在インターネット上に蔓延しているウイルスの状況を見る限り、ウイルス対策の主流となっているウイルス対策ソフトで各コンピュータに免疫を与える方法では、ウイルスの絶滅は困難ということがわかる。また、別のウイルス対策の手法として、多数のリンクを持つノード（ハブノード）に対し優先的に免疫を与える方法［1］や、あるノードを任意に選び、その隣接ノードに対し免疫を与える方法［2］などが報告されている。

本研究では、新たなウイルス対策手法としてネットワーク上にウイルスを削除してまわるプログラムを放つ手法を提案し、この手法ではウイルス蔓延度はどのように変動するのか、すでにネットワーク上に蔓延しているウイルスを絶滅可能かどうかをシミュレーションにより検証した。本論文ではウイルスを削除して回るプログラムを捕食者と呼び、捕食者が存在するノードを捕食者ノード、捕食者が感染ノードを治癒させることを捕食と呼ぶことにする。

捕食、被食のシミュレーションは通常、セル･オートマトンを用いたライフゲームで行われるが、セル･オートマトンを用いたライフゲームの空間構造は通常二次元格子状で一様であるため、インターネットのネットワーク構造に合致していない。そこで本研究では、インターネットのネットワーク構造に近いと言われているスケールフリーネットワーク（Barabási-Albertモデル、以下BA）と、BAとの比較のため、空間構造がほぼ一様なランダムネットワーク（以下RN）の2つのネットワークモデル［3］を用いてシミュレーションを行った。BAは少数のリンクを持つノードが多く存在し、ハブノードが少数存在するモデルであり、リンク数分布がべき乗則に従うモデルである。また、RNは全てのノードがほぼ同じリンク数を持ち、リンク数分布がポアソン分布に従うモデルである。今回の研究ではRN、BA共に100万ノード、平均リンク数6でシミュレーションを行った。

これらのネットワークモデルを用いてシミュレーションを行った結果、捕食者が自己消滅するようにした場合、ウイルスの絶滅は捕食プログラムの寿命を長くしない限り困難なことが明らかとなった。また、ランダムネットワークでは捕食者がネットワークに蔓延する閾値が存在し、スケールフリーネットワークは明確な閾値が確認できなかった。

2．感染、捕食プロセス

感染、捕食プロセスをシミュレーションするために、各ノードの状態変化を離散化し、stepによる時間発展を行う。感染モデルはSIモデルを採用し、捕食プロセスはSIモデルに捕食を付加したものを採用した。

2.1　感染モデル

実際の生物間の伝染病解析で用いられるモデルとしてSIモデルがあるが、このモデルはネットワーク上のコンピュータウイルス蔓延の解析にも有効なため、本研究でもこのSIモデルを採用した。このモデルはノードの状態をS（susceptible）：健康な状態、I（Infected）：ウイルスに感染した状態の2通りに分類し、図1のようにS→Iの流れでノードの状態を遷移させる。またSからIに状態が遷移する感染確率はλとする。SIモデルでは$\lambda > 0$の場合、感染ノードは治癒しないため、時間が十分に経過すれば全てのノードが感染状態になる。

2.2　捕食プロセス

捕食プロセスはSIモデルにP（predator）：捕食者が存在する状態を付加したものになる。

シミュレーションの1stepの流れは、まず、ウイルス感染しているノードが隣接しているノードに対して感染活動を行い、そのウイルスの攻撃パケットを感染ノードに隣接している捕食者が感知し、感染ノードに対して捕食者が捕食活動を行う。その後、捕食者が自己消滅確率Dで自己消滅する。現実のネットワーク上でこのような捕食プログラムを自己消滅させなかった

図1．SIモデルにおけるノードの状態遷移。

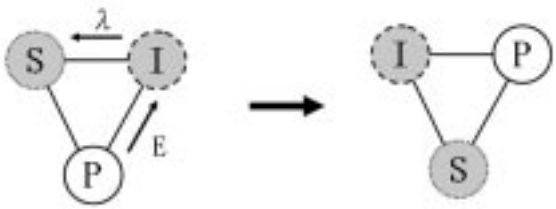

図2　SIモデルに捕食プロセスを付加した1stepの流れ。Sは健康なノード、Iは感染ノード、Pは捕食者ノードを意味する。捕食者は隣接している感染ノードに捕食活動を行い、感染ノードは隣接している健康なノードに感染活動を行う。

場合、捕食者はPCに残り続けPCに負荷をかけ続けるため駆除方法としては好ましくない。よって、シミュレーションでは捕食者は自己消滅するという条件をつけた。捕食者が自己消滅した場合、ノードの状態はS健康な状態になる。これらのプロセスを用いた場合1stepで感染者と捕食者が同時に動くプロセスになる（図２）。

3．捕食シミュレーション

シミュレーションでは、RN、BA上で捕食-被食関係が存在した場合、どのような挙動を起こすのか、ネットワークにすでに蔓延しているウイルスを絶滅できるのか検証を行った。

3.1　シミュレーション方法

捕食シミュレーションは、ウイルスが定常状態に至ってから捕食者を設置することにした。また、SIモデルでは感染確率λが０を超えている場合、感染者が治癒しないため、十分な時間が経過すれば、全てのノードが感染する。よっ

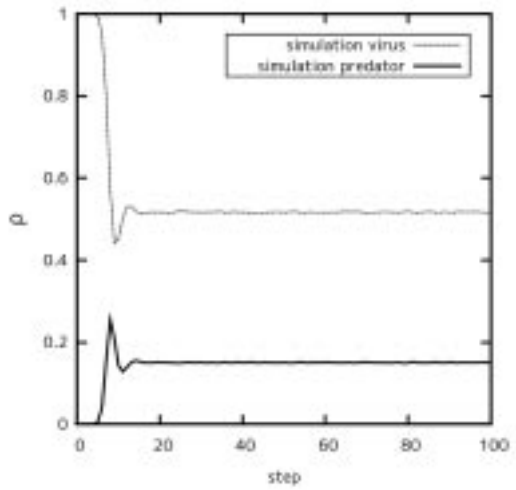

図3　RNでの捕食シミュレーションの捕食者、ウイルスのノード割合の時間変動。縦軸ρはノード割合を意味する。捕食者とウイルスが互いに影響し合い振動が発生し、その後すぐに振動が減衰し、捕食者、ウイルス割合共に定常状態に入っている。感染力λは0.5、捕食確率Eは0.5、自己消滅確率Dは1.0。

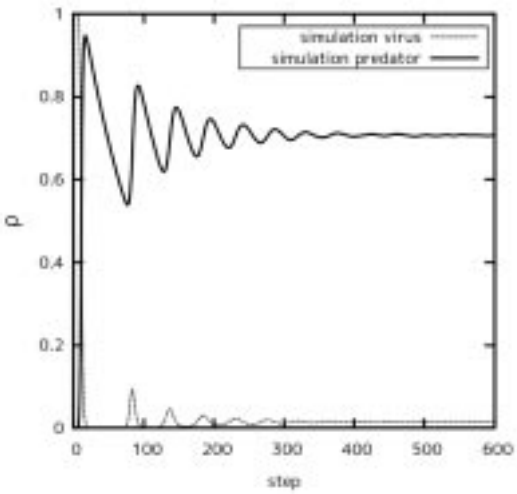

図4　RNでの捕食シミュレーションの捕食者、ウイルスのノード割合の時間変動。捕食者とウイルスが互いに影響し合い振動が発生し、その後、振動を繰り返しながら減衰し捕食者、ウイルス割合共に定常状態に入っている。感染力λは0.5、捕食確率Eは0.5、自己消滅確率Dは0.01。

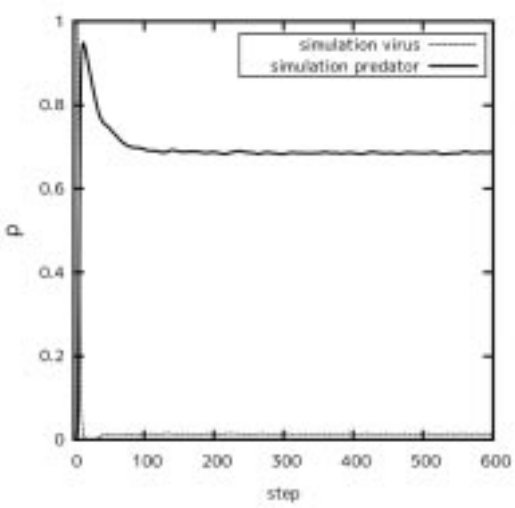

図5　BAでの捕食シミュレーションの捕食者、ウイルスのノード割合の時間変動。捕食者とウイルスが互いに影響し合い振動が発生しているが、図4に比べすぐに振動は減衰し、捕食者、ウイルス割合共に定常状態に入っている。各値は図4と同様。

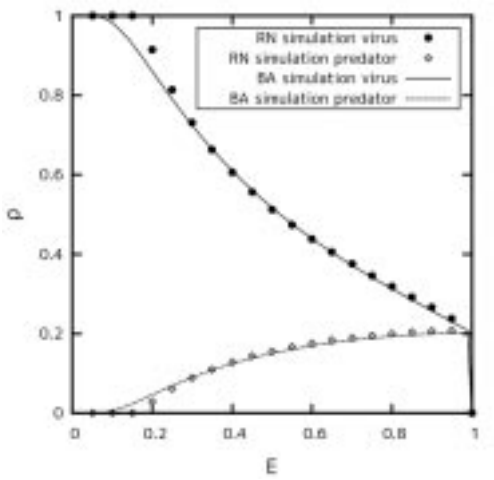

図6　RNとBAの各捕食確率Eに対する捕食者とウイルスのノード割合。自己消滅確率Eを変化させている。ウイルス、捕食者が共に定常状態になったときのウイルス、捕食者割合の50stepの平均を、その捕食確率Eでのウイルス割合、捕食者割合としている。RN、BA共に捕食確率Eが1.0のときのみウイルスが絶滅している。感染力λは0.5、自己消滅確率Dは1.0。

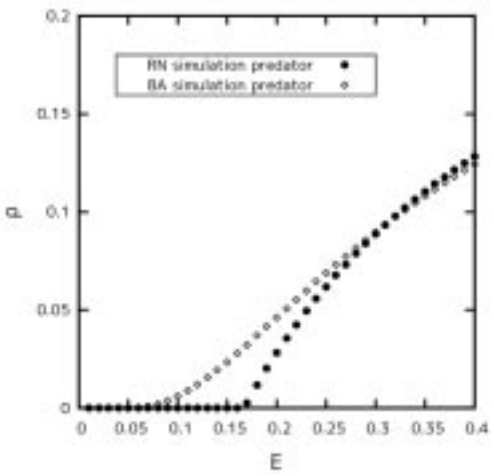

図7　図6の拡大図。BAはRNよりも捕食者がネットワークに蔓延する閾値が低くなっている。

て、λ＞0ならばウイルスの定常状態での割合はすべて1.0となるので、時間短縮のため、最初に全てのノードを感染させた状態でランダムに1つノードを選び、そのノードに捕食者を設置するシミュレーション方法を採用した。

通常、ノードとリンクを用いたネットワークシミュレーションでは構造情報を格納するために隣接行列が用いられるが、この手法では現在のコンピュータのメモリ制限により、1万ノード程度のシュミレーションが限界である。本研究ではプログラム上でノード間のリンク情報を配列へ格納する際、隣接行列を用いず、1次元行列を2つ用いてリンク情報をインデックス化してメモリの節約を行った。この手法では、1000万ノード程度のネットワークのシミュレーションが可能である。

3.2　シミュレーション結果と考察

シミュレーションの結果、RN、BA共に図3のように捕食者とウイルスが互いに影響し合い振動が発生し、その後振動が減衰し、捕食者、ウイルス共に定常状態に入った。この振動は、捕食者と被食者の増減関係を式にしたモデルで

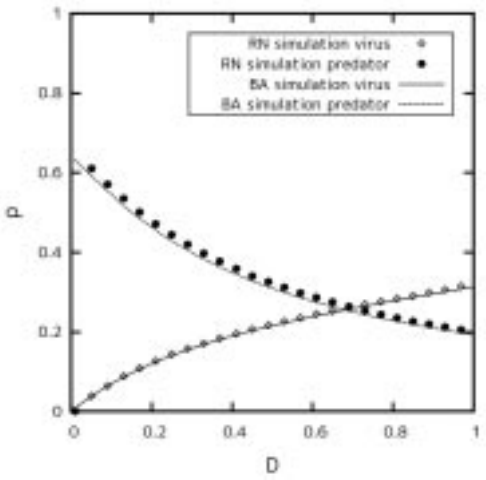

図8　RNとBAの各自己消滅確率Dに対する捕食者とウイルスのノード割合。捕食者、ウイルスの割合の平均のとり方は図6と同じ。感染力λと捕食確率Eは一定にし、自己消滅確率Dを変化させている。捕食確率Eは0.8、感染確率λは0.5。自己消滅確率Dの一番低い値は0.01。RNはD=0.01でウイルスが絶滅している。

あるLotka-Volterraモデル［4］と同じ特徴が現れていると考えられる。また、RNで捕食者の自己消滅確率Dを低くし、捕食者の寿命を長くしたところ、捕食者、ウイルス共に振動を繰り返しながら振動が減衰し、定常状態に入った（図4）。BAでも同じく捕食者の自己消滅確率Dを低くしたことろ、RNに比べ振動の減衰の仕方には変化が見られず、すぐに振動は減衰し、捕食者、ウイルス共に定常状態に入った（図5）。

RN、BA共に十分な時間が経過すれば、振動は減衰し捕食者、ウイルス割合共に定常状態に入ることがわかったため、感染力λ、捕食者自己消滅確率Dを一定にし、捕食確率Eを変化させた場合、ウイルス、捕食者の定常状態はどのように変化するのかを調べた。捕食確率Eが増加するに従い、RN、BA共にウイルスは減少し捕食者は増加する傾向が見られた（図6）。また、捕食確率Eが1.0のときRN、BA共にウイルスが絶滅した。絶滅した理由は、全てのノードが感染している初期状態かつ、隣接感染ノードを必ず捕食するというシミュレーションを行ったためである。

RNで捕食者がネットワーク上に蔓延する閾値が見られたが、BAでは明確な閾値が確認できなかった（図7）。BAの捕食者蔓延の閾値がRNと違い確認できなかった要因として、ハブノードが捕食者をネットワーク上に拡散し、捕食者がネットワーク上に蔓延しやすくなっているためと考えられる。スケールフリーネットワーク上でのSISモデルウイルス感染では、ハブノードがウイルスをネットワークに拡散し、感染力λの閾値が0になる特徴をもっているが［1］、これと同じような現象が起き、捕食者がネットワーク上に蔓延しやすくなったと考えられる。捕食者の自己消滅確率Dの値を1.0と固定し、捕食確率Eを変えウイルス捕食シミュレーションを行った結果、捕食確率Eを1.0にし、捕食者を設置するときの初期感染割合を1.0にするという特殊な条件下のみウイルスが絶滅した。このような特殊な条件ではなく、ウイルスをネットワーク上から絶滅させるために、捕食確率Eと感染力λを一定にし、捕食者の自己消滅確率Dを変化させシミュレーションを行った。その結果、図8のようにRN、BA共に自己消滅確率Dの値をかなり低くしない限り、感染者を絶滅に追い込めないことがわかった。

4．捕食シミュレーション理論式

SIモデルに捕食者を付加したRNでの理論式を立て、捕食者がネットワーク上に蔓延する閾値を調べた。RNは各ノードのリンク数のばらつきが小さいため、全てのノードのリンク数は平均リンク数<k>と近似して式を立てることができる。しかし、BAでは各ノードのリンク数に大きくばらつきがあり、このような近似が行なえないため、本研究ではRNのみ理論式を立てた。また、Runge-Kutta法で数値解を求め、RNで行なったシミュレーション結果との比較を行った。

4.1　SIモデルに捕食を付加した理論式

シミュレーション結果を理論的に解析するために、SIモデルに捕食者を付加したRNでの理論式を式（1）、（2）のように立てた。また、RNはリンク数分布がポアソン分布となり、各ノードのリンク数は平均リンク数まわりの分散が小さいため、全てのノードのリンク数は平均リンク数<k>と近似した。

$$\frac{dp(t)}{dt} = -Dp(t) + E\left[1 - \{1 - p(t)\}^{\langle k \rangle}\right] i(t) \quad (1)$$

$$\frac{di(t)}{dt} = \lambda\left[1 - \{1 - \mathrm{i}(t)\}^{\langle k \rangle}\right]\{1 - p(t) - i(t)\} - \mathrm{E}\left[1 - \{1 - p(t)\}^{\langle k \rangle}\right] i(t) \quad (2)$$

$i(t)$は時間tでの感染者のノード割合、$p(t)$は時間tでの捕食者のノード割合を示す。また、式（1）の左辺は捕食者の変化割合を示している。右辺第１項は捕食者が自己消滅確率Dで減少していくことを示し、右辺第２項は感染ノードが隣接している捕食者ノードから捕食されることを示している。式（2）の左辺は感染者の変化割合を示す。また、右辺第１項は健康なノードが隣接している感染ノードから感染することを示し、右辺第２項は感染ノードが周りの捕食者ノードから捕食され減少していくことを示している。

4.2　RNで捕食者がネットワーク上に蔓延する閾値

RNで捕食者がネットワーク上に蔓延する一番低い値で捕食者が蔓延した場合、捕食者の割合は図７のように小さいと考えられる。よって、式（1）を捕食者の割合が小さいと仮定し、$\{1-p(t)\}^{\langle k \rangle}$を$p(t)$でマクローリン展開し、一次までの項を式（1）に代入すると

$$\frac{dp(t)}{dt} = -Dp(t) + E\langle \mathrm{k} \rangle i(t) p(t) \quad (3)$$

となる。この式（3）から捕食者がネットワーク上に蔓延する閾値を考えていく。式（3）で十分に時間が経ちウイルス、捕食者の蔓延度が定常状態になったとすると

$$\frac{dp(t)}{dt} = 0 \quad \begin{matrix} p(t \to \infty) = p \\ i(t \to \infty) = i \end{matrix} \quad (4)$$

となり、式（3）は、

$$0 = -pD + E\langle k \rangle ip \quad (5)$$

のようになる。これを変形させ、

$$i = \frac{D}{E\langle k \rangle} \quad (6)$$

とする。シミュレーションでは感染割合iの初期値は1.0としていた。よって、捕食者が蔓延する条件は、感染者の割合が1.0未満になる条件と同じである。よって、その条件$i<1$をつけると、

$$\frac{E}{D} > \frac{1}{\langle k \rangle} \quad (7)$$

となり、捕食者は「捕食確率E/自己消滅確率D」が「１/平均リンク数」を超えていれば蔓延するという臨界値が得られる。シミュレーションでは平均リンク数を６とし、自己消滅確率Dを1.0としているため、捕食確率Eが約0.167を超えていれば捕食者はネットワーク上に蔓延する。RNで捕食確率Eを0.01ずつ変化させシミュレーションを行った結果、0.17から捕食者がネットワーク上に蔓延し、理論値と近い値となった（図７）。

4.3　シミュレーション結果と理論式の数値解法の比較

式（2）、（3）を直接解くのは困難なため、Runge-Kutta法で数値解を求め、シミュレーション結果との比較を行った。捕食確率Eが高くなるにつれ、式（2）、（3）の数値解の感染割合に比べ、シミュレーション結果の感染割合が小さくなっていくことがわかった（図９）。これは、理論式では感染者に捕食者が隣接している

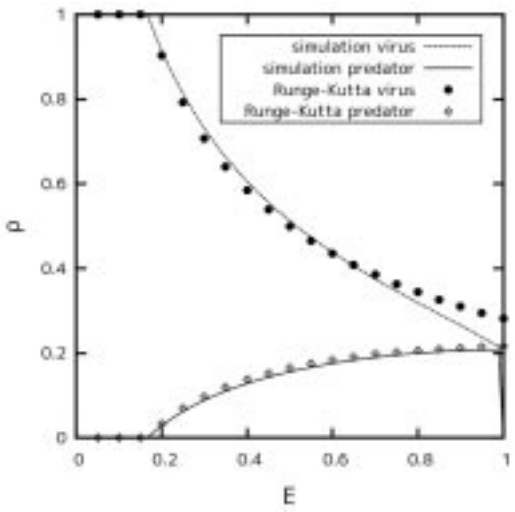

図9　RNでのシミュレーションの各捕食確率Eに対する捕食者、ウイルスの割合と、式（1）、(2)をRunge-Kutta法で数値解法した各捕食確率Eに対する捕食者とウイルスのノード割合。シミュレーション方法、各値は図6と同じ。捕食確率Eが高くなるにつれ、シミュレーションのウイルスのノード割合と理論式のウイルスのノード割合との差異が広がっている。

割合をネットワーク全体の捕食者割合と同値としているのに対し、シミュレーションでは捕食確率Eが高くなるにつれ、捕食者が感染者を追跡し、結果として感染者に隣接している捕食者の割合がネットワーク全体の捕食者割合に比べ大きくなっているためと考えられる（図10)。

この現象を確認するために、シミュレーションで感染ノードの隣接ノードには、捕食者がどの程度の割合で存在するかを調べた。感染者に隣接している捕食者割合の平均が、ネットワーク全体での捕食者割合と同値だった場合、捕食者は感染ノードの位置に関係なく散らばっていて、追跡現象は起きていないと判断できる。RNでシミュレーションを行った結果、捕食確率Eが高くなるにつれ、感染ノードに捕食者ノードが隣接している割合が、実際の捕食者割合よりも大きくなっていた（図11)。このことから、捕食者が感染者を追跡する現象が起きてい

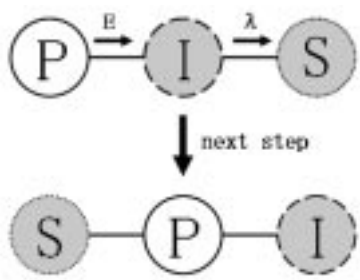

図10　シミュレーションでは捕食者がウイルスを追跡する。矢印のように捕食確率Eが高いほど、次のstepで感染するノードに捕食者が隣接する確率が高くなる。

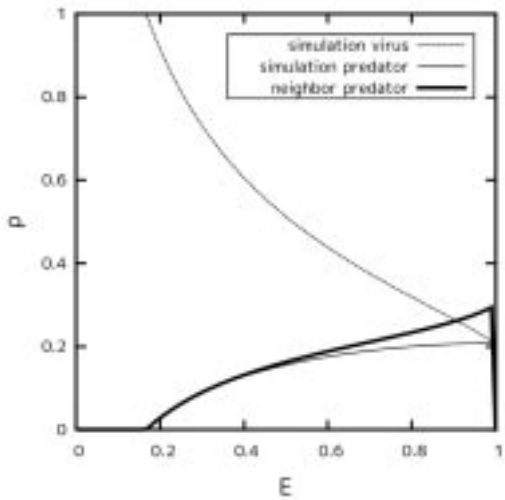

図11　RNでのシミュレーションの各捕食確率Eに対する捕食者とウイルスのノード割合。neighbor predatorはウイルスの隣接ノードに存在する捕食者の割合を各感染ノードごとに出し、その平均を取ったものとなっている。感染力λと自己消滅確率Dは一定にし捕食確率Eを変化させている。各値は図6と同じ。捕食確率Eが高くなるにつれ、感染者に隣接している捕食者の割合がネットワーク全体での捕食者の割合よりも大きくなっている。

ることが明らかとなった。同じように、BAでシミュレーションを行った結果、捕食確率Eが低いところでも、実際の捕食者割合よりも感染

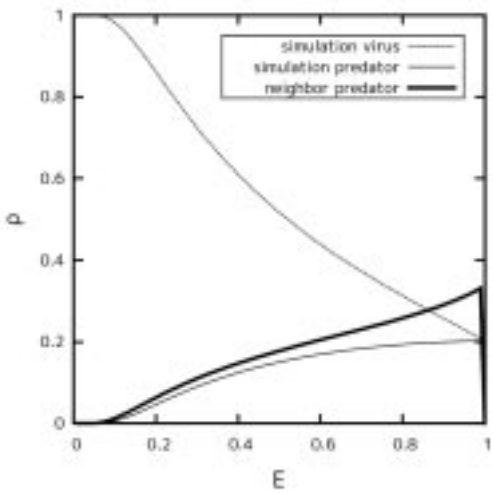

図12　BAでのシミュレーションの各捕食確率Eに対する捕食者とウイルスのノード割合。図11のRNでのシミュレーションをBAで行った結果となっている。RNと同じように捕食確率Eが高くなるにつれ、感染者に隣接している捕食者の割合とネットワーク全体での捕食者の割合との差が広がっている。

ノードに隣接している捕食者割合が上回っていた。しかし、RNと同じように、捕食確率Eが高くなるにつれ、感染ノードに捕食者ノードが隣接している割合と実際の捕食者割合との差が広がっている（図12)。この理由として、BAはハブノードが存在し、ハブノードに捕食者が存在する場合や､ハブノードが感染している場合、感染ノードと捕食者が隣接している割合が大きくなるため、BAで捕食確率Eが低いところでも、実際の捕食者割合よりも感染ノードに隣接している捕食者割合が上回ったと考えられる。これらのことから、BAでもRNと同じように追跡現象が起きていることが明らかとなった。

5．おわりに

本研究ではRN、BA上で、ウイルスが感染活動を行い、その後、捕食者が捕食活動を行うという感染、捕食プロセスを用いてシミュレーションを行った。その結果、RNに比べBAは捕食者が蔓延しやすいことを明らかにした。これは、BAのハブノードがネットワーク上に捕食者を拡散したためと考えられる。また、RN、BA共に捕食確率を高くするか、捕食者の寿命を極端に長くしない限り、ウイルスの絶滅は難しいということがわかった。

本研究ではすでに蔓延しているウイルスを捕食プログラムによって絶滅させることができるかどうかを検証した。今後は、より確実にウイルスを絶滅させることのできるような捕食プロセスが必要であると考えられる。

【参考文献】

[1] Romualdo Pastor-Satorras and Alessandro Vespignani, “Evolution and Structure of the internet”, pp.180-210, CAMBRIDGE（2004).
[2] Reuven Cohen, Shlomo Havlin and Daniel ben-Avraham, “Efficient Immunization Strategies for Computer Networks and Populations”, Phys. Rev . Lett. vol.91, 247901（2003).
[3] 編著．林 幸雄．「ネットワーク科学の道具箱」, pp.4-13, 近代科学社（2007).
[4] 巌佐 庸．「数理生物学入門」, pp.34-39, 共立出版（1990).