Inoculation Strategies for Victims of Viruses

James Aspnes[1] Kevin Chang[1] Aleksandr Yampolskiy[1]

[1]Department of Computer Science, Yale University

Symposium On Discrete Algorithms(SODA), 2005

Outline

1. The Model
   - The Basic Problem
   - Game Theoretic Model
2. Nash Equilibrium
   - Characterization
   - Computing pure Nash Equilibrium
3. Optimizing Social Cost
   - Characterization
   - Sum of Squares Partition Problem

Meet the Authors...

James Aspnes

Kevin Chang

Aleksandr Yampolskiy

Outline

1. The Model
   - The Basic Problem
   - Game Theoretic Model
2. Nash Equilibrium
   - Characterization
   - Computing pure Nash Equilibrium
3. Optimizing Social Cost
   - Characterization
   - Sum of Squares Partition Problem

The Basic Problem

- Consider System of networked computers
- If a virus infects some node it evetually infects all the unprotected neighbors of infected machine
- Best strategy to install antivirus software on all machines?
- Each machine owner decides to install antivirus software or decides to get infected.

The Basic Problem

- Consider System of networked computers
- If a virus infects some node it evetually infects all the unprotected neighbors of infected machine
- Best strategy to install antivirus software on all machines?
- Each machine owner decides to install antivirus software or decides to get infected.

The Basic Problem

- Consider System of networked computers
- If a virus infects some node it evetually infects all the unprotected neighbors of infected machine
- Best strategy to install antivirus software on all machines?
- Each machine owner decides to install antivirus software or decides to get infected.

The Basic Problem

- Consider System of networked computers
- If a virus infects some node it evetually infects all the unprotected neighbors of infected machine
- Best strategy to install antivirus software on all machines?
- Each machine owner decides to install antivirus software or decides to get infected.

The Basic Problem

- Consider System of networked computers
- If a virus infects some node it evetually infects all the unprotected neighbors of infected machine
- Best strategy to install antivirus software on all machines?
- Each machine owner decides to install antivirus software or decides to get infected.

Outline

1. The Model
   - The Basic Problem
   - Game Theoretic Model
2. Nash Equilibrium
   - Characterization
   - Computing pure Nash Equilibrium
3. Optimizing Social Cost
   - Characterization
   - Sum of Squares Partition Problem

The Model

- Players: n players corresponding to nodes in the graph
- Strategies: A node i has two possible strategies- to inoculate or to get infected denoted by a_i. Nodes strategy profile is denoted by $\vec{a} \in [0,1]^n$.
- Attack Graph: let nodes that choose to install software be $I_{\vec{a}}$. Now attack graph is $G - I_{\vec{a}}$
- Given the choices of the nodes, virus chooses a node uniformly randomly to infect. Now virus infects the unprotected nodes and all its neghbours.

The Model

- Players: n players corresponding to nodes in the graph
- Strategies: A node i has two possible strategies- to inoculate or to get infected denoted by a_i. Nodes strategy profile is denoted by $\vec{a} \in [0,1]^n$.
- Attack Graph: let nodes that choose to install software be $I_{\vec{a}}$. Now attack graph is $G - I_{\vec{a}}$
- Given the choices of the nodes, virus chooses a node uniformly randomly to infect. Now virus infects the unprotected nodes and all its neghbours.

The Model

- Players: n players corresponding to nodes in the graph
- Strategies: A node i has two possible strategies- to inoculate or to get infected denoted by a_i. Nodes strategy profile is denoted by $\vec{a} \in [0,1]^n$.
- Attack Graph: let nodes that choose to install software be $I_{\vec{a}}$. Now attack graph is $G - I_{\vec{a}}$
- Given the choices of the nodes, virus chooses a node uniformly randomly to infect. Now virus infects the unprotected nodes and all its neghbours.

The Model

- Players: n players corresponding to nodes in the graph
- Strategies: A node i has two possible strategies- to inoculate or to get infected denoted by a_i. Nodes strategy profile is denoted by $\vec{a} \in [0,1]^n$.
- Attack Graph: let nodes that choose to install software be $I_{\vec{a}}$. Now attack graph is $G - I_{\vec{a}}$
- Given the choices of the nodes, virus chooses a node uniformly randomly to infect. Now virus infects the unprotected nodes and all its neghbours.

Attack Graph

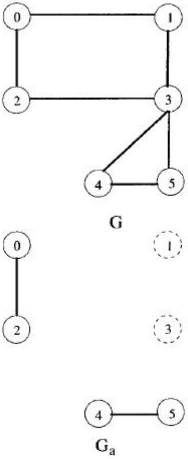

Figure 1: Sample graph G and its attack graph $G_{\vec{a}}$ for $\vec{a} = 010100$.

Attack Graph

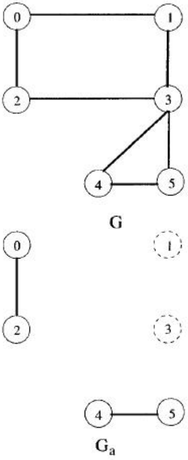

Figure 1: Sample graph G and its attack graph $G_{\vec{a}}$ for $\vec{a} = 010100$.

The Model

- Cost Model: Let C be the cost of antivirus software and L be the cost due to virus infection. Then given a strategy profile $\vec{a} \in [0,1]^n$, the cost to node i is -

  $$cost_i(\vec{a}) = a_i C + (1 - a_i) L \cdot p_i(\vec{a})$$

  Where $p_i(\vec{a})$ is the probability that node i gets infected *given that it chooses to get infected* , and $p_i(\vec{a}) = k_i/n$,Where k_i is the size of the connected component in which node i is present.
- Social cost: Given a strategy profile $\vec{a} \in [0,1]^n$, total social cost is given by -

  $$cost(\vec{a}) = \sum_{j=0}^{n-1} cost_j(\vec{a})$$

The Model

- Cost Model: Let C be the cost of antivirus software and L be the cost due to virus infection. Then given a strategy profile $\vec{a} \in [0,1]^n$, the cost to node i is -

  $$cost_i(\vec{a}) = a_i C + (1 - a_i) L \cdot p_i(\vec{a})$$

  Where $p_i(\vec{a})$ is the probability that node i gets infected *given that it chooses to get infected* , and $p_i(\vec{a}) = k_i / n$,Where k_i is the size of the connected component in which node i is present.
- Social cost: Given a strategy profile $\vec{a} \in [0,1]^n$, total social cost is given by -

  $$cost(\vec{a}) = \sum_{j=0}^{n-1} cost_j(\vec{a})$$

The Model

- Cost Model: Let C be the cost of antivirus software and L be the cost due to virus infection. Then given a strategy profile $\vec{a} \in [0,1]^n$, the cost to node i is -

$$cost_i(\vec{a}) = a_i C + (1 - a_i) L \cdot p_i(\vec{a})$$

  Where $p_i(\vec{a})$ is the probability that node i gets infected *given that it chooses to get infected* , and $p_i(\vec{a}) = k_i/n$,Where k_i is the size of the connected component in which node i is present.

- Social cost: Given a strategy profile $\vec{a} \in [0,1]^n$, total social cost is given by -

$$cost(\vec{a}) = \sum_{j=0}^{n-1} cost_j(\vec{a})$$

Outline

1. The Model
   - The Basic Problem
   - Game Theoretic Model
2. Nash Equilibrium
   - Characterization
   - Computing pure Nash Equilibrium
3. Optimizing Social Cost
   - Characterization
   - Sum of Squares Partition Problem

Mixed Equilibria

Theorem

Let $S(i)$be the expected size of the insecure component that contains node i of the attack graph $G_{\vec{a}[i/0]}$ and let $t = Cn/L$ then a strategy profile $\vec{a}$ is a Nash equilibrium if and only if
1)$\forall i$ such that $a_i = 1$, $S(i) \geq t$
2)$\forall i$ such that $a_i = 0$, $S(i) \leq t$
3)$\forall i$ such that $0 < a_i < 1$, $S(i) = t$

Mixed Equilibria

Theorem

Let $S(i)$be the expected size of the insecure component that contains node i of the attack graph $G_{\vec{a}[i/0]}$ and let $t = Cn/L$ then a strategy profile $\vec{a}$ is a Nash equilibrium if and only if
1)$\forall i$ *such that* $a_i = 1,\ S(i) \geq t$
2)$\forall i$ *such that* $a_i = 0,\ S(i) \leq t$
3)$\forall i$ *such that* $0 < a_i < 1,\ S(i) = t$

Pure Equilibria

Theorem

Let $t = Cn/L$ then a strategy profile $\vec{a}$ is a pure Nash equilibrium if and only if
1)Every component in the attack graph $G_{\vec{a}}$ has size at most t
2)Inserting any secure node $j \in I_{\vec{a}}$ and its edges into $G_{\vec{a}}$ yields a component of size at least t.

Pure Equilibria

Theorem

Let $t = Cn/L$ then a strategy profile $\vec{a}$ is a pure Nash equilibrium if and only if
1)Every component in the attack graph $G_{\vec{a}}$ has size at most t
2)Inserting any secure node $j \in I_{\vec{a}}$ and its edges into $G_{\vec{a}}$ yields a component of size at least t.

Outline

1. The Model
   - The Basic Problem
   - Game Theoretic Model
2. Nash Equilibrium
   - Characterization
   - Computing pure Nash Equilibrium
3. Optimizing Social Cost
   - Characterization
   - Sum of Squares Partition Problem

Constructive proof of Existance

Theorem

Starting From any pure strategy profile $\vec{a}$, if at each step some participant with a suboptimal strategy switches its strategy , the system converges to pure Nash Equilibrium in no more than 2n steps.

Proof.

Define a potential function as -

$$\Phi(\vec{a}) = \sum_{A \in S_{big}(\vec{a})} |A| - \sum_{A \in S_{small}(\vec{a})} |A|$$

□

Constructive proof of Existance

Theorem

Starting From any pure strategy profile $\vec{a}$, if at each step some participant with a suboptimal strategy switches its strategy , the system converges to pure Nash Equilibrium in no more than 2n steps.

Proof.

Define a potential function as -

$$\Phi(\vec{a}) = \sum_{A \in S_{big}(\vec{a})} |A| - \sum_{A \in S_{small}(\vec{a})} |A|$$

$\square$

Constructive proof of Existance

Theorem

Starting From any pure strategy profile $\vec{a}$, if at each step some participant with a suboptimal strategy switches its strategy , the system converges to pure Nash Equilibrium in no more than 2n steps.

Proof.

Define a potential function as -

$$\Phi(\vec{a}) = \sum_{A \in S_{big}(\vec{a})} |A| - \sum_{A \in S_{small}(\vec{a})} |A|$$

□

Extremal Nash Equiliria

Theorem

Computing the pure nash equilibrium with lowest cost and highest cost are $NP-hard$ problems

Proof.

For lowest cost reduce *Vertex Cover* to decision problem *Does there exist Nash Equilibrium with cost less than c?*
For highest cost reduce *Independent Dominating Set* to decision problem *Does there exist Nash Equilibrium with cost greater than c?...* □

Extremal Nash Equiliria

Theorem

Computing the pure nash equilibrium with lowest cost and highest cost are $NP - hard$ problems

Proof.

For lowest cost reduce *Vertex Cover* to decision problem *Does there exist Nash Equilibrium with cost less than c?*
For highest cost reduce *Independent Dominating Set* to decision problem *Does there exist Nash Equilibrium with cost greater than c?...*

Extremal Nash Equiliria

Theorem

Computing the pure nash equilibrium with lowest cost and highest cost are $NP-hard$ problems

Proof.

For lowest cost reduce *Vertex Cover* to decision problem *Does there exist Nash Equilibrium with cost less than c?*
For highest cost reduce *Independent Dominating Set* to decision problem *Does there exist Nash Equilibrium with cost greater than c?...* □

Outline

1. The Model
   - The Basic Problem
   - Game Theoretic Model
2. Nash Equilibrium
   - Characterization
   - Computing pure Nash Equilibrium
3. Optimizing Social Cost
   - Characterization
   - Sum of Squares Partition Problem

Characterization and NP-hardness

Theorem

If $\vec{a}$ is an optimum strategy, then every component in attack graph $G_{\vec{a}}$ has size at most $\max(1, (t+1)/2)$

.

Theorem

It is NP-hard to compute a socially optimal strategy

Characterization and NP-hardness

Theorem

If $\vec{a}$ is an optimum strategy, then every component in attack graph $G_{\vec{a}}$ has size at most $\max(1,(t+1)/2)$

.

Theorem

It is NP-hard to compute a socially optimal strategy

Characterization and NP-hardness

Theorem

If $\vec{a}$ is an optimum strategy, then every component in attack graph $G_{\vec{a}}$ has size at most $\max(1, (t+1)/2)$

.

Theorem

It is NP-hard to compute a socially optimal strategy

Reduction to sum of squares

Reduction

The Optimization can be restated as, find the set of nodes $I_{\vec{a}}$ to secure such that the objective function-

$$C\,|I_{\vec{a}}| + \frac{L}{n} \sum_{V \in \phi(I_{\vec{a}})} |V|^2$$

is minimized. Now assume that we know the value of $|I_{\vec{a}}|$ that optimizes this function then the problem reduces minimizing the second term.

Reduction to sum of squares

Reduction

The Optimization can be restated as, find the set of nodes $I_{\vec{a}}$to secure such that the objective function-

$$C\,|I_{\vec{a}}| + \frac{L}{n} \sum_{V \in \phi(I_{\vec{a}})} |V|^2$$

is minimized. Now assume that we know the value of $|I_{\vec{a}}|$that optimizes this function then the problem reduces minimizing the second term.

Outline

1. The Model
   - The Basic Problem
   - Game Theoretic Model
2. Nash Equilibrium
   - Characterization
   - Computing pure Nash Equilibrium
3. Optimizing Social Cost
   - Characterization
   - Sum of Squares Partition Problem

Sum of Squares Partition

Sum of Squares Partition Problem

By removing a set F of at most m nodes , partition the graph into disconnected components $H_1, H_2, ..., H_k$ such that $\sum_i |H_i|^2$ is minimum.

Theorem

Let OPT be the optimum objective function value for the sum-of-squares partition problem on G with removal of at most m nodes, then we can find a set of $O(log^2 n)m$ nodes , such that their removal creates disconnected components $H_1, H_2, ..., H_k$ such that $\sum_i |H_i|^2 \leq O(1) \cdot OPT$.

Sum of Squares Partition

Sum of Squares Partition Problem

By removing a set F of at most m nodes , partition the graph into disconnected components $H_1, H_2, \ldots, H_k$ such that $\sum_i |H_i|^2$ is minimum.

Theorem

Let OPT be the optimum objective function value for the sum-of-squares partition problem on G with removal of at most m nodes, then we can find a set of $O(\log^2 n)m$ nodes , such that their removal creates disconnected components $H_1, H_2, \ldots, H_k$ such that $\sum_i |H_i|^2 \leq O(1) \cdot OPT$.

Sum of Squares Partition

Sum of Squares Partition Problem

By removing a set F of at most m nodes , partition the graph into disconnected components $H_1, H_2, \ldots, H_k$ such that $\sum_i |H_i|^2$ is minimum.

Theorem

Let OPT be the optimum objective function value for the sum-of-squares partition problem on G with removal of at most m nodes, then we can find a set of $O(\log^2 n)m$ *nodes , such that their removal creates disconnected components* $H_1, H_2, \ldots, H_k$ *such that* $\sum_i |H_i|^2 \leq O(1) \cdot OPT$.

Extensions and Open Issues

- Friendship Nash Equilibrium[1] and existence of pure Nash Equilibria in presence of friendship factor
- To reach the Nash Equilibrium each node needs to know the structure of complete graph. How to model this problem so that reaching NE does not require the knowledge of complete graph.
- Consider a model in which players have to report to the central enforcer who their neighbors are...
- Better Approximation algorithm for the sum of squares partition problem.

[1] On Windfall of Friendship : Inoculation Strategies on Social Network, Stefan Schmid et. al. in EC 2008

Extensions and Open Issues

- Friendship Nash Equilibrium[1] and existence of pure Nash Equilibria in presence of friendship factor
- To reach the Nash Equilibrium each node needs to know the structure of complete graph. How to model this problem so that reaching NE does not require the knowledge of complete graph.
- Consider a model in which players have to report to the central enforcer who their neighbors are...
- Better Approximation algorithm for the sum of squares partition problem.

[1] On Windfall of Friendship : Inoculation Strategies on Social Network, Stefan Schmid et. al. in EC 2008

Extensions and Open Issues

- Friendship Nash Equilibrium[1] and existence of pure Nash Equilibria in presence of friendship factor
- To reach the Nash Equilibrium each node needs to know the structure of complete graph. How to model this problem so that reaching NE does not require the knowledge of complete graph.
- Consider a model in which players have to report to the central enforcer who their neighbors are...
- Better Approximation algorithm for the sum of squares partition problem.

[1] On Windfall of Friendship : Inoculation Strategies on Social Network, Stefan Schmid et. al. in EC 2008

Extensions and Open Issues

- Friendship Nash Equilibrium[1] and existence of pure Nash Equilibria in presence of friendship factor
- To reach the Nash Equilibrium each node needs to know the structure of complete graph. How to model this problem so that reaching NE does not require the knowledge of complete graph.
- Consider a model in which players have to report to the central enforcer who their neighbors are...
- Better Approximation algorithm for the sum of squares partition problem.

[1] On Windfall of Friendship : Inoculation Strategies on Social Network, Stefan Schmid et. al. in EC 2008

Extensions and Open Issues

- Friendship Nash Equilibrium[1] and existence of pure Nash Equilibria in presence of friendship factor
- To reach the Nash Equilibrium each node needs to know the structure of complete graph. How to model this problem so that reaching NE does not require the knowledge of complete graph.
- Consider a model in which players have to report to the central enforcer who their neighbors are...
- Better Approximation algorithm for the sum of squares partition problem.

[1] On Windfall of Friendship : Inoculation Strategies on Social Network, Stefan Schmid et. al. in EC 2008

For Further Reading

- On Windfall of Friendship : Inoculation Strategies on Social Networks , Stefan Schmid et al. Electronic Commerce 2008.
- When Selfish meets Evil:Byzantine Players in Virus Inoculation Game, Stefan Schmid et al. PODC 2006
- Inoculation Strategies for victims of viruses and sum of squares partition problem, James Aspnes et al. Technical Report , Yale University July 2004

Friendship Nash Equilibrium

A potential function

What about Potential Function

$$\Phi(\vec{a}) = \sum_{i=1}^{k} (\alpha_i - \beta_i)$$

Where , $k = \#of\ components\ in\ G_{\vec{a}}$
$\alpha_i = \#of\ nodes\ in\ component\ i\ for\ whom\ the\ component\ is\ too\ large,$
$\beta_i = \#of\ nodes\ in\ component\ i\ for\ whom\ the\ component\ is\ too\ small$

Friendship Nash Equilibrium

A potential function

What about Potential Function

$$\Phi(\vec{a}) = \sum_{i=1}^{k} (\alpha_i - \beta_i)$$

Where , $k = \#of\ components\ in\ G_{\vec{a}}$
$\alpha_i = \#of\ nodes\ in\ component\ i\ for\ whom\ the\ component\ is\ too\ large,$
$\beta_i = \#of\ nodes\ in\ component\ i\ for\ whom\ the\ component\ is\ too\ small$