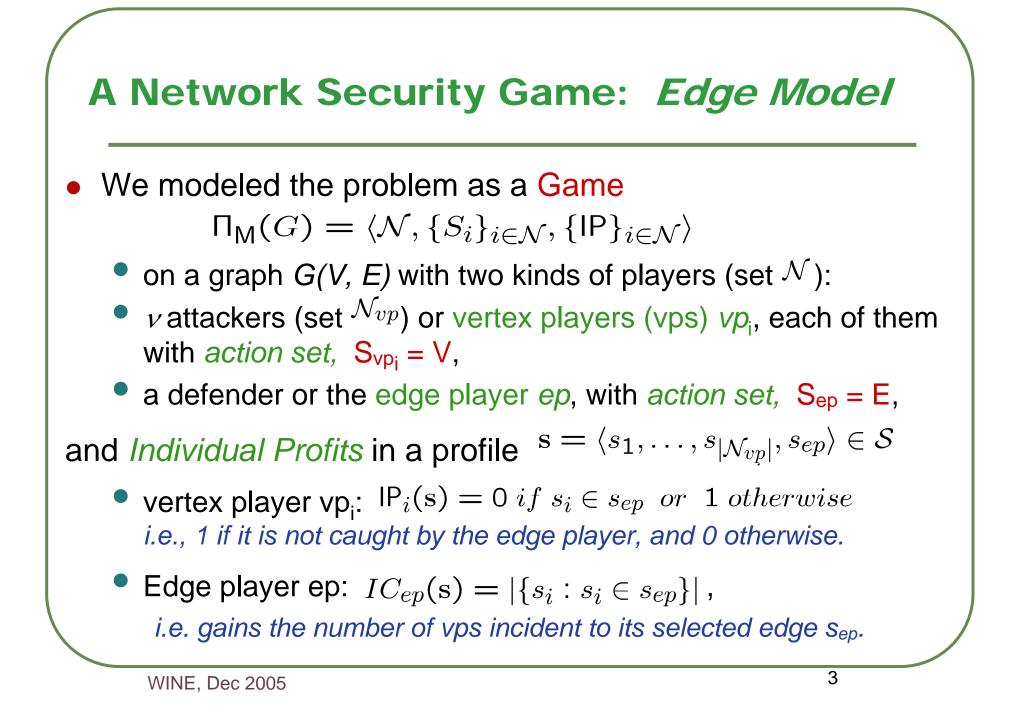
A Graph-Theoretic Network Security Game

M. Mavronicolas*, V. Papadopoulou*, A. Philippou* and P. Spirakis§

University of Cyprus, Cyprus* University of Patras and RACTI, Greece§



- Information network with
 - nodes *insecure* and vulnerable to *infection* by attackers e.g., viruses, Trojan horses, eavesdroppers, and
 - a system security software or a defender of limited power, e.g. able to clean a part of the network.
- In particular, we consider
 - a graph G with
 - v attackers each of them locating on a node of G and
 - a defender, able to clean a single edge of the graph.



Nash Equilibria in the Edge Model

- We consider pure and mixed strategy profiles.
- Study associated Nash equilibria (NE), where no player can unilaterally improve its Individual Cost by switching to another configuration.

Notation

- P_s(ep, e): probability ep chooses edge e in s
- $P_s(vp_i, v)$: probability vp_i chooses vertex v in s
- $P_s(vp, v) = \sum_{i \ge N_{vp}} P_s(vp_i, v)$: # vps located on vertex v in s
- D_s(i): the support (actions assigned positive probability) of player i2 N in s.
- $ENeigh_s(v) = \{(u,v) \in E : (u,v) \in D_s(ep)\}$
- $P_s(Hit(v)) = \sum_{e \in ENeigh(v)} P_s(ep, e)$: the hitting probability of v
- $m_{s}(v) = \sum_{i \in N_{vp}} P_{s}(vp_{i}, v)$: expected # of vps choosing v
- $m_{s}(e) = m_{s}(u) + m_{s}(v)$
- Neigh_G(X) = { $u \notin X : (u, v) \in E(G)$ }

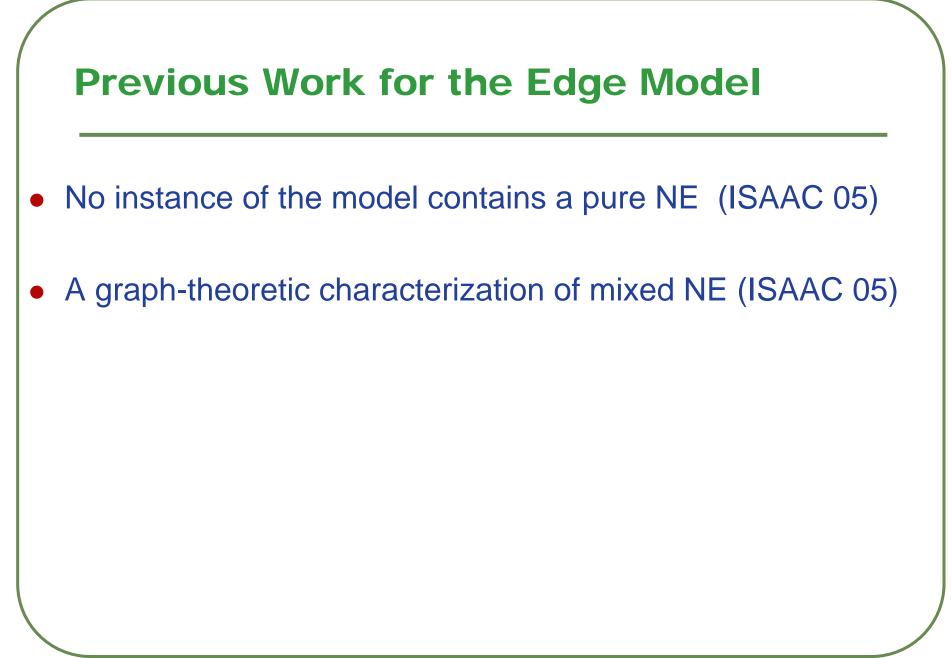


vertex players vp;

$$\mathsf{P}_i(\mathbf{s}) = \sum_{v \in V} P_{\mathbf{s}}(vp_i, v) \cdot (1 - P_{\mathbf{s}}(Hit(v)))$$
(1)

• edge player *ep*:

$$\mathsf{IP}_{ep}(\mathbf{s}) = \sum_{e=(u,v)\in E} P_{\mathbf{s}}(ep,e) \cdot (m_{\mathbf{s}}(u) + m_{\mathbf{s}}(v)) \tag{2}$$



Summary of Results

- Polynomial time computable mixed NE on various graph instances:
 - regular graphs,
 - graphs with, polynomial time computable, *r*-regular factors
 - graphs with perfect matchings.
- Define the Social Cost of the game to be
 - the expected number of attackers catch by the protector
- The Price of Anarchy in any mixed NE is
 - upper and lower bounded by a linear function of the number of vertices of the graph.
- Consider the generalized variation of the problem considered, the Path model
 - The existence problem of a pure NE is NP-complete

Significance

- The *first* work (with an exception of ACY04) to model *network security problems* as strategic game and study its associated Nash equilibria.
- One of the few works highlighting a fruitful interaction between *Game Theory* and *Graph Theory*.
- Our results contribute towards answering the general question of Papadimitriou about the complexity of Nash equilibria for our special game.
- We believe Matching Nash equilibria (and/or extensions of them) will find further applications in other network games.

Pure and Mixed Nash Equilibria

- Theorem 1. [ISAAC05] If G contains more than one edges, then Π(G) has no pure Nash Equilibrium.
- Theorem 2. [ISAAC05] (characterization of mixed NE)
- A mixed configuration s is a Nash equilibrium for any $\Pi(G)$ if and only if:
- **1.** $D_s(ep)$ is an edge cover of G and
- **2.** $D_s(vp)$ is a vertex cover of the graph obtained by $D_s(ep)$.
- **3.** (a) $P(Hit(v)) = P_s(Hit(u)) = min_v P_s(Hit(v)), 8 u, v 2 D_s(vp),$ (b) $\sum_{e \ 2 \ D_s(ep)} P_s(ep, e) = 1$
- 4. (a) $m_s(e_1)=m_s(e_2)=max_e m_s(e)$, 8 e_1 , e_2 2 $D_s(ep)$ and (b) $\sum_{v \ge V(Ds(ep))} m_s(v)=v$.

Background

- Definition 1. A graph G is polynomially computable r-factor graph if its vertices can be partitioned, in polynomial time, into a sequence G_{r_1}, \dots, G_{r_k} of k *r-regular* vertex disjoint subgraphs, for an integer *k*, $1 \cdot k \cdot n, G_r' = \{Gr_1 \cup \dots \cup Gr_k\}$ the graph obtained by the sequence.
- A *two-factor* graph is can be recognized and decomposed into a sequence C₁, ..., C_k, 1 · k · n, in polynomial time (via Tutte's reduction).

Polynomial time NE : Regular Graphs

Theorem 1. For any $\Pi(G)$ for which G is an r-regular graph, a mixed NE can be computed in constant time O(1).

Proof.

Construct profile s^r on $\Pi(G)$:

For any $i \in \mathcal{N}_{vp}$, $P_{\mathbf{s}^{\mathbf{r}}}(vp_i, v) := \frac{1}{n}$, $\forall v \in V(G)$ and then set, $\mathbf{s}^{\mathbf{r}}_j := \mathbf{s}^{\mathbf{r}}_i$, $\forall j \neq i, j \in \mathcal{N}_{vp}$. Set $P_{\mathbf{s}^{\mathbf{r}}}(ep, e) := \frac{1}{m}$, $\forall e \in E$.

 \Rightarrow 8 v 2 V, $P_{s}(Hit(v)) = | ENeigh(v) | / m$

 \Rightarrow 8 v2 V and vp_i, IC_i (s^r-i, [v]) = 1- r/m

• Also, 8 e 2 E, $m(v) = v \phi(1/n)$. Thus, 8 e 2 E, $IC_{ep}(s^{r}-ep,[e]) = 2\phi v/n$

 \Rightarrow s^r is a NE.

Polynomial time NE : r-factor Graphs

 Corollary 1. For any Π(G), such that G is a polynomial time computable r- factor graph, a mixed NE can be computed in polynomial time O(T(G)), where O(T(G)) is the time needed for the computation of G_r['] from G.

Polynomial time NE : Graphs with Perfect Matchings

Theorem 2. For any $\Pi(G)$ for which G has a perfect matching, a mixed NE can be computed in polynomial time, $O(n^{1/2} \phi m)$.

Proof.

- Compute a perfect matching of G, M using time O(n^{1/2}¢m).
- Construct the following profile s^{t} on $\Pi(G)$:

For any $i \in \mathcal{N}_{vp}$, $P_{\mathbf{sf}}(vp_i, v) := \frac{1}{n}$, $\forall v \in V(G)$ and set $\mathbf{sf}_j := \mathbf{sf}_i$, $\forall j \neq i, j \in \mathcal{N}_{vp}$. Set $P_{\mathbf{sf}}(ep, e) := \frac{1}{|M|}$, $\forall e \in E$.

- 8 v 2 V, $P_s(Hit(v)) = 1/|M|$
- $\Rightarrow 8 \ v2 \ V \ and \ vp_i, \ \ IC_i (s^r_{-i}, [v]) = 1 1/|M| = 1 2/n$
- Also, 8 e 2 E, $m(v) = v \phi(1/n)$. Thus, 8 e 2 E, $IC_{ep}(s^{r}-ep,[e]) = 2\phi v/n$

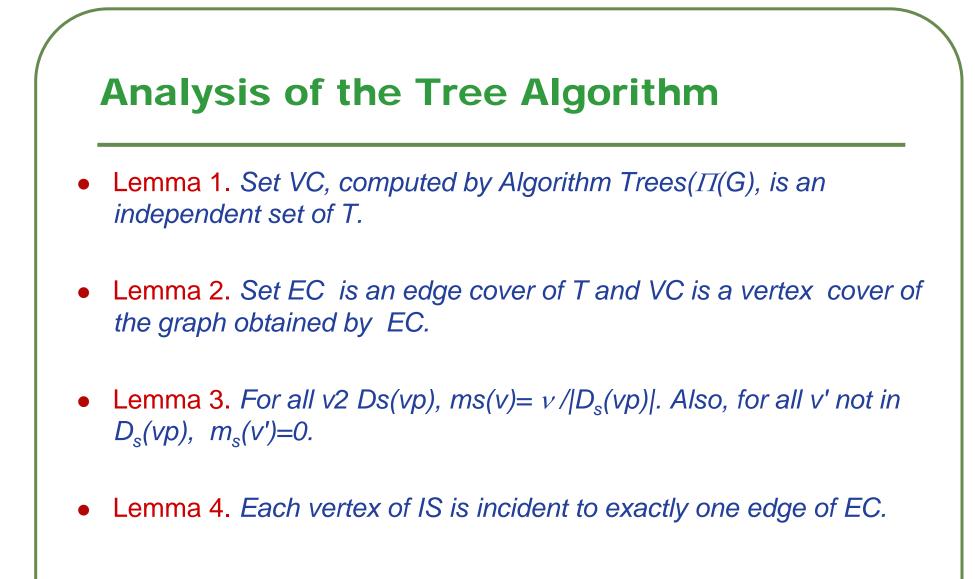
 \Rightarrow S^f VISINE, NEC 2005

Polynomial time NE : Trees

Algorithm Trees($\Pi(T)$) Input: $\Pi(T)$ Output: a NE on $\Pi(T)$

- **1**. Initialization: VC:=;, EC:=;, r:=1, $T_r := T$.
- 2. Repeat until $T_r ==;$
 - a) Find the leaves of the tree T_r , leaves(T_r) and add leaves(T_r) in VC.
 - b) For each v 2 leaves(T_r), add (v,parent_{Tr}(v) in EC
 - c) Update tree: $T_r = T_r \setminus \text{leaves}(T_r) \setminus \text{parents}(\text{leaves}(T_r))$
- 3. Set s^t: For any $i \in \mathcal{N}_{VP}$, set $D_{s^t}(vp_i) := VC$ and $D_{s^t}(ep) := EC$. Then set $D_{s^t}(vp_j) := D_{s^t}(vp_i)$, $\forall j \neq i, j \in \mathcal{N}_{VP}$.

and apply the uniform distribution on support of each player.



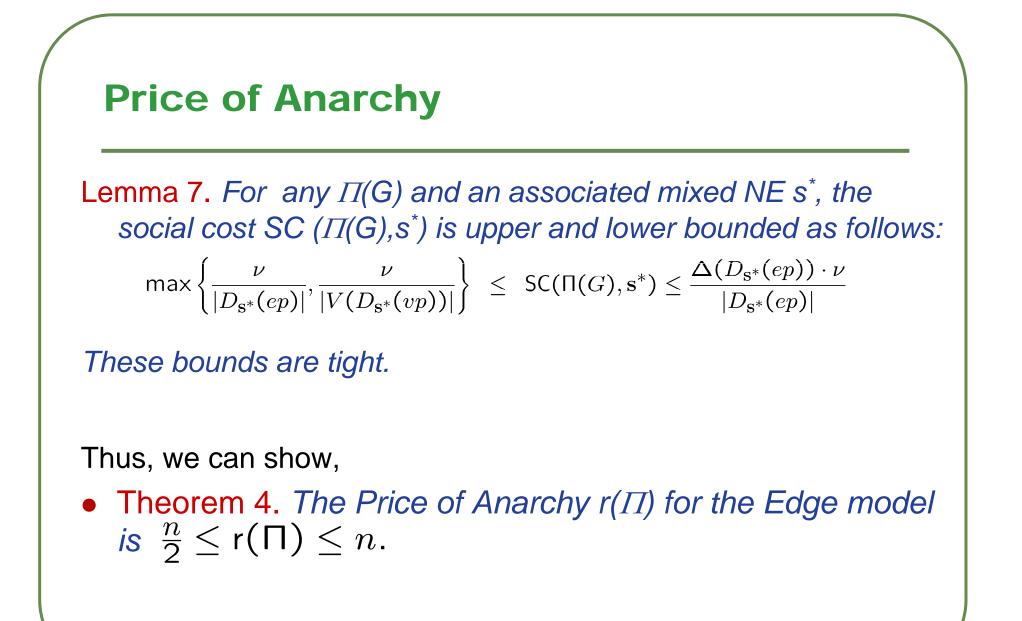
Analysis of the Algorithm (Cont.)

By Lemmas 2 and 4, we get,

• Lemma 5. For all $v \in D_{s^t}(vp)$, $P_s(Hit(v)) = \frac{1}{|D_{s^t}(ep)|}$. Also, for all $v' \notin D_{s^t}(vp)$, $P_s(Hit(v')) \ge \frac{1}{|D_{s^t}(ep)|}$.

Thus,

Theorem 3. For any $\Pi(T)$, where T is a tree graph, algorithm *Trees*($\Pi(T)$) computes a mixed NE in polynomial time O(n).



Path Model

 If we let the protector to be able to select a single path of G instead of an edge, called the path player (pp)

 \Rightarrow The Path Model

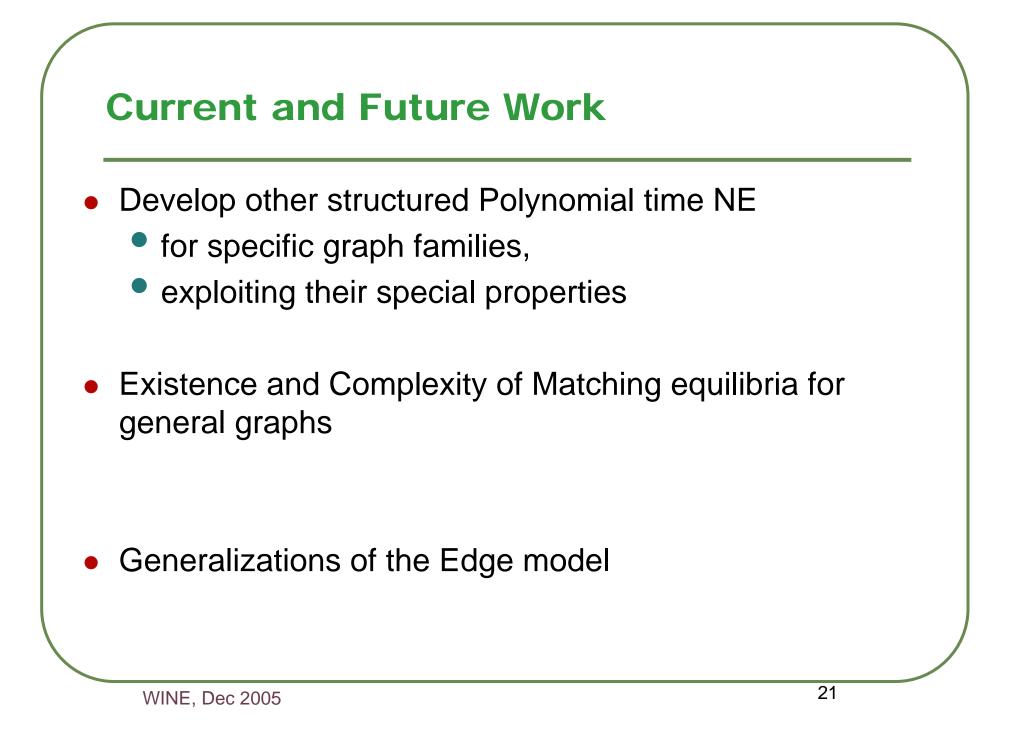
• Theorem. For any graph G, $\Pi(G)$ has a pure NE if and only if G contains a hamiltonian path.

Proof.

- Assume in contrary: $\Pi(G)$ contains a pure NE s but G is not hamiltonian.
- There exists a set of nodes U of G not contained in pp's action, s_{pp} .
- \Rightarrow for all players vp_i, i 2 N_{vp}, it holds s_i 2 U
- \Rightarrow Path player gains nothing, while he could gain more.
- \Rightarrow s is NOT a pure NE of Π (G), contradiction.

Path Model

• Corollary. The existence problem of pure NE for the Path model is *NP*-complete.



Thank you for your Attention !