A Graph-Theoretic Network Security Game*

Marios Mavronicolas[1], Vicky Papadopoulou[1], Anna Philippou[1], and Paul Spirakis[2]

[1] Department of Computer Science, University of Cyprus, Nicosia CY-1678, Cyprus. { mavronic, viki,annap}@ucy.ac.cy

[2] Department of Computer Engineering and Informatics, University of Patras, 265 00 Patras, Greece, & Research and Academic Computer Technology Institute, 261 10 Patras, Greece. spirakis@cti.gr

Abstract. Consider a network vulnerable to viral infection. The system security software can guarantee safety only to a limited part of the network. We model this practical network scenario as a non-cooperative multi-player game on a graph, with two kinds of players, a set of *attackers* and a *protector* player, representing the viruses and the system security software, respectively. Each attacker player chooses a node of the graph (or a set of them, via a probability distribution) to infect. The protector player chooses independently, in a basic case of the problem, a simple path or an edge of the graph (or a set of them, via a probability distribution) and cleans this part of the network from attackers. Each attacker wishes to maximize the probability of escaping its cleaning by the protector. In contrast, the protector aims at maximizing the expected number of cleaned attackers. We call the two games obtained from the two basic cases considered, as the *Path* and the *Edge* model, respectively. We are interested in the associated *Nash equilibria* on them, where no network entity can unilaterally improve its local objective. We obtain the following results:

- The problem of existence of a pure Nash equilibrium is $\mathcal{NP}$-complete for the Path model. This opposed to that, no instance of the Edge model possesses a pure Nash equilibrium, proved in [4].
- We compute, in polynomial time, mixed Nash equilibria on corresponding graph instances. These graph families include, regular graphs, graphs that can be decomposed, in polynomially time, into vertex disjoint r-regular subgraphs, graphs with perfect matchings and trees.
- We utilize the notion of *social cost* [3] for measuring system performance on such scenario; here is defined to be the utility of the protector. We prove that the corresponding *Price of Anarchy* in any mixed Nash equilibria of the game is upper and lower bounded by a linear function of the number of vertices of the graph.

* This work was partially supported by the IST Programs of the European Union under contract numbers IST-2001-33116 (FLAGS) and IST-2004-001907 (DELIS).

1 Introduction

Motivation. This work considers a problem of *Network Security*, related to the protection of a system from harmful procedures (e.g. viruses, worms). Consider an information network where the nodes of the network are insecure and vulnerable to infection such as, viruses, Trojan horses, the *attackers*. A *protector*, i.e. system security software, is available in the system but it can guarantees security only to a limited part of the network, such as a simple path or a single link of it, chosen via a probability distribution. Each harmful entity targets a location (i.e. a node) of the network via a probability distribution; the node is damaged unless it is cleaned by the system security software. Apparently, the harmful entities and the system security software have conflicting objectives. The security software seeks to protect the network as much as possible, while the harmful entities wish to avoid being caught by the software so that they be able to damage the network. Thus, the system security software seeks to maximize the expected number of viruses it catches, while each harmful entity seeks to maximize the probability it escapes from the security software.

Naturally, we model this scenario as a non-cooperative multi-player strategic game played on a graph with two kinds of players: the *vertex players* representing the harmful entities, and the *edge* or the *path player* representing each one of the above two cases for the system security software considered; where it chooses a simple path or a single edge, respectively. The corresponding games are called the *Path* and the *Edge* model, respectively. In both cases, the Individual Cost of each player is the quantity to be maximized by the corresponding entity. We are interested in the *Nash equilibria* [7, 8] associated with these games, where no player can unilaterally improve its Individual Cost by switching to a more advantageous probability distribution.

Summary of Results. Our results are summarized as follows:

- We prove that the problem of existence of pure Nash equilibria in the Path model is $\mathcal{NP}$-complete (Theorem 1). This opposes to that, the simpler case of this model, i.e. that the Edge model posses no pure Nash equilibrium [4].
- [4] provides a graph-theoretic characterization of mixed Nash Equilibria for the Edge model. Unfortunately, this characterization only implies an exponential time algorithm for the general case. Here, we utilize the characterization in order to compute, in polynomial time, mixed Nash equilibria for specific graph instances of the game. In particular, we combine the characterization with a suitable exploration of some graph-theoretic properties of each graph family considered to obtain polynomial time mixed Nash equilibria. These graph families include, regular graphs, graphs that can be partitioned into vertex disjoint regular subgraphs, graphs with perfect matchings and trees (Theorem 3, Proposition 2, Theorems 4 and 5, respectively).
- We measure the system performance with respect to the problem considered utilizing the notion of the *social cost* [3]. Here, it is defined to be the number of attackers catch by the protector. We compute upper and lower bounds

of the social cost in any mixed Nash equilibria of the Edge model. Using these bounds, we show that the corresponding Price of Anarchy is upper and lower bounded by a linear function of the number of vertices of the graph (Theorem 6).

Due to space limits, some proofs are omitted; we include them in the full version of the paper [5].

Related Work and Significance. This work is a step further in the development of the new born area of *Algorithmic Game Theory*. It is also one of the only few works to model *network security problems* as a strategic game. Such a research line is that of *Interdependent Security* games, e.g. [2]. However, we remark that *none* of these works, with an exception of [2], study Nash equilibria on the games considered. This work is also one of the only few works that study games exploiting heavily *Graph-Theoretic* tools. In [2], the authors study a security problem and establish connections with variants of the Graph Partition problem. In [1], the authors study a two-players game on a graph, establish connections with the k-server problem. In a recent work of ours [4], we consider the simpler of the two games considered here, the Edge model. We provide a non-existence result for pure Nash equilibria of the model and a polynomial time algorithm for mixed Nash equilibria for bipartite graphs. Finally, our results contribute toward answering the general question of Papadimitriou [10] about the complexity of Nash equilibria for our special game.

2 Framework

Throughout, we consider an undirected graph $G(V, E)$, with $|V(G)| = n$ and $|E(G)| = m$. Given a set of vertices $X \subseteq V$, the graph $G \backslash X$ is obtained by removing from G all vertices of X and their incident edges. For any vertex $v \in V(G)$, denote $\Delta(v)$ the degree of vertex v in G. Denote $\Delta(G)$ the maximum degree of the graph G. A *simple* path, P, of G is a path of G with no repeated vertices, i.e. $P = \{v_1, \cdots, v_i \cdots v_k\}$, where $1 \leq i \leq k \leq n$, $v_i \in V$, $(v_i, v_{i+1}) \in E(G)$ and each $v_i \in V$ appears at most once in P. Denote $\mathcal{P}(G)$ the set of all possible simple paths in G. For a tree graph T denote $root \in V$, the root of the tree and $leaves(T)$ the leaves of the tree T. For any $v \in V(T)$, denote $parent(v)$, the parent of v in the tree and $children(v)$ its children in the tree T. For any $A \subseteq V$, let $parents(A) := \{u \in V : u = father(v),\ v \in A\}$.

2.1 Protector-Attacker models

Definition 1. *An information network is represented as an undirected graph $G(V, E)$. The vertices represent the network hosts and the edges represent the communication links. For $\mathsf{M} = \{\mathsf{P}, \mathsf{E}\}$, we define a non-cooperative game $\Pi_{\mathsf{M}} = \langle \mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{\mathsf{IC}\}_{i \in \mathcal{N}} \rangle$ as follows:*

– *The set of players is* $\mathcal{N} = \mathcal{N}_{vp} \cup \mathcal{N}_p$, *where* $\mathcal{N}_{vp}$ *is a finite set of* vertex *players* vp_i, $i \geq 1$, $p = \{pp, ep\}$ *and* $\mathcal{N}_p$ *is a singleton set of a player* p *which is either (i) a* path *player and* $p = pp$ *or (ii) an* edge *player and* $p = ep$, *in the case where* $\mathsf{M} = \mathsf{P}$ *or* $\mathsf{M} = \mathsf{E}$, *respectively.*
– *The strategy set* S_i *of each player* vp_i, $i \in \mathcal{N}_{vp}$, *is* V*; the strategy set* S_p *of the player* p *is either (i) the set of paths of* G, $\mathcal{P}(G)$ *or (i)* E, *when* $\mathsf{M} = \mathsf{P}$ *or* $\mathsf{M} = \mathsf{E}$, *respectively. Thus, the strategy set* $\mathcal{S}$ *of the game is* $\left(\underset{i \in N_{vp}}{\times} S_i \right) \times S_p$ *and equals to* $|V|^{|\mathcal{N}_{vp}|} \times |\mathcal{P}(G)|$ *or* $|V|^{|\mathcal{N}_{vp}|} \times |E|$, *when* $\mathsf{M} = \mathsf{P}$ *or* $\mathsf{M} = \mathsf{E}$, *respectively.*
– *Take any* strategy profile $\mathbf{s} = \langle s_1, \ldots, s_{|\mathcal{N}_{vp}|}, s_p \rangle \in \mathcal{S}$, *called a* configuration.
  - *The* Individual Cost *of vertex player* vp_i *is a function* $\mathsf{IC}_i : \mathcal{S} \to \{0, 1\}$ *such that* $\mathsf{IC}_i(\mathbf{s}) = \begin{cases} 0, & s_i \in s_p \\ 1, & s_i \notin s_p \end{cases}$ *; intuitively,* vp_i *receives* 1 *if it is not caught by the player* p, *and* 0 *otherwise.*
  - *The* Individual Cost *of the player* p *is a function* $\mathsf{IC}_p : \mathcal{S} \to \mathbb{N}$ *such that* $\mathsf{IC}_p(\mathbf{s}) = |\{s_i : s_i \in s_p\}|$.

We call the games obtained as the Path *or the* Edge *model, for the case where* $\mathsf{M} = \mathsf{P}$ *or* $\mathsf{M} = \mathsf{E}$, *respectively.*

The configuration $\mathbf{s}$ is a *pure Nash equilibrium* [7, 8] (abbreviated as *pure NE*) if for each player $i \in \mathcal{N}$, it minimizes IC_i over all configurations $\mathbf{t}$ that differ from $\mathbf{s}$ only with respect to the strategy of player i. We consider *mixed strategies* for the Edge model. In the rest of the paper, unless explicitly mentioned, when referring to mixed strategies, these apply on the Edge model. A *mixed strategy* for a vertex player (resp., edge player) is a probability distribution over vertices (resp., over edges) of G. A *mixed strategy profile* $\mathbf{s}$ is a collection of mixed strategies, one for each player. Denote $P_{\mathbf{s}}(ep, e)$ the probability that edge player ep chooses edge $e \in E(G)$ in $\mathbf{s}$; denote $P_{\mathbf{s}}(vp_i, v)$ the probability that player vp_i chooses vertex $v \in V$ in $\mathbf{s}$. Denote $P_{\mathbf{s}}(vp, v) = \sum_{i \in \mathcal{N}_{vp}} P_{\mathbf{s}}(vp_i, v)$ the probability that vertex v is chosen by some vertex player in $\mathbf{s}$. The *support* of a player $i \in \mathcal{N}$ in the configuration $\mathbf{s}$, denoted $D_{\mathbf{s}}(i)$, is the set of pure strategies in its strategy set to which i assigns strictly positive probability in $\mathbf{s}$. Denote $D_{\mathbf{s}}(vp) = \bigcup_{i \in \mathcal{N}_{vp}} D_{\mathbf{s}}(i)$. Let also $ENeigh_{\mathbf{s}}(v) = \{(u, v) \in E : \ (u, v) \in D_{\mathbf{s}}(ep)\}$. Given a mixed strategy profile $\mathbf{s}$, we denote $(\mathbf{s}_{-x}, [y])$ a configuration obtained by $\mathbf{s}$, where all but player x play as in $\mathbf{s}$ and player x plays the pure strategy y.

A mixed strategic profile $\mathbf{s}$ induces an *Expected Individual Cost* IC_i for each player $i \in \mathcal{N}$, which is the expectation, according to $\mathbf{s}$, of its corresponding Individual Cost (defined previously for pure strategy profiles). The mixed strategy profile, denoted as $\mathbf{s}^*$, is a *mixed Nash equilibrium* [7, 8] (abbreviated as mixed NE) if for each player $i \in \mathcal{N}$, it maximizes IC_i over all configurations $\mathbf{t}$ that differ from $\mathbf{s}$ only with respect to the mixed strategy of player i. Denote $BR_{\mathbf{s}}(x)$ the set of *best response (pure) strategies* of player x in a mixed strategy profile $\mathbf{s}$, that is, $\mathsf{IC}_x(\mathbf{s}_{-x}, y) \geq \mathsf{IC}_x(\mathbf{s}_{-x}, y')$, $\forall \ y \in BR_{\mathbf{s}}(x)$ and $y' \notin BR_{\mathbf{s}}(x)$, $y' \in S_x$, where S_x is the strategy set of player x (see also [9, chapter 3]). A *fully* mixed strategy profile is one in which each player plays with positive probability all strategies of its strategy set.

For the rest of this section, fix a mixed strategy profile $\mathbf{s}$. For each vertex $v \in V$, denote $Hit(v)$ the event that the edge player hits vertex v. So, $P_{\mathbf{s}}(Hit(v)) = \sum_{e \in ENeigh(v)} P_{\mathbf{s}}(ep, e)$. Define the minimum hitting probability $P_{\mathbf{s}}$ as $\min_v P_{\mathbf{s}}(Hit(v))$. For each vertex $v \in V$, denote $m_{\mathbf{s}}(v)$ the expected number of vertex players choosing v (according to $\mathbf{s}$). For each edge $e = (u, v) \in E$, denote $m_{\mathbf{s}}(e)$ the expected number of vertex players choosing either u or v; so, $m_{\mathbf{s}}(e) = m_{\mathbf{s}}(u) + m_{\mathbf{s}}(v)$. It is easy to see that for each vertex $v \in V$, $m_{\mathbf{s}}(v) = \sum_{i \in \mathcal{N}_{vp}} P_{\mathbf{s}}(vp_i, v)$. Define the maximum expected number of vertex players choosing e in $\mathbf{s}$ as $\max_e m_{\mathbf{s}}(e)$. We proceed to calculate the Expected Individual Costs for any vertex player $vp_i \in \mathcal{N}_{vp}$ and for the edge player.

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

$$\mathsf{IC}_{ep}(\mathbf{s}) = \sum_{e=(u,v) \in E(G)} P_{\mathbf{s}}(ep, e) \cdot m_{\mathbf{s}}(e) = \sum_{e=(u,v) \in E(G)} P_{\mathbf{s}}(ep, e) \cdot (\sum_{i \in N_{vp}} P_{\mathbf{s}}(vp_i, u) + P_{\mathbf{s}}(v_i, v)) \tag{2}$$

Social Cost and Price of Anarchy. We utilize the notion of *social cost* [3] for evaluating the system performance.

Definition 2. *For model* M, $\mathsf{M} = \{\mathsf{P}, \mathsf{E}\}$, *we define the* social cost *of configuration* $\mathbf{s}$ *on instance* $\Pi_{\mathsf{M}}(G)$, $\mathsf{SC}(\Pi_{\mathsf{M}}, \mathbf{s})$, *to be the sum of vertex players of* Π_{M} *arrested in* $\mathbf{s}$. *That is,* $\mathsf{SC}(\Pi_{\mathsf{M}}, \mathbf{s}) = \mathsf{IC}_p(\mathbf{s})$ *(*$p = \{pp, vp\}$*, when* $\mathsf{M} = \mathsf{P}$ *and* $\mathsf{M} = \mathsf{E}$*, respectively). The system wishes to maximize the social cost.*

Definition 3. *For model* M, $\mathsf{M} = \{\mathsf{P}, \mathsf{E}\}$, *the* price of anarchy, $\mathsf{r}(\mathsf{M})$ *is*

$$\mathsf{r}(\mathsf{M}) = \max_{\Pi_{\mathsf{M}}(G), \mathbf{s}^*} \frac{\max_{\mathbf{s} \in \mathcal{S}} \mathsf{SC}(\Pi_{\mathsf{M}}(G), \mathbf{s})}{\mathsf{SC}(\Pi_{\mathsf{M}}(G), \mathbf{s}^*)}$$

2.2 Background from Graph Theory

Throughout this work, we consider the (undirected) graph $G = G(V, E)$.

Definition 4. *A graph* G *is* polynomially computable r-factor *graph if its vertices can be partitioned, in polynomial time, into a sequence* $G_{r_1} \cdots G_{r_k}$ *of* k r*-regular disjoint subgraphs, for an integer* k, $1 \leq k \leq n$. *That is,* $\bigcup_{1 \leq i \leq k} V(G_{r_i}) = V(G)$, $V(G_{r_i}) \bigcap V(G_{r_j}) = \emptyset$ *and* $\Delta_{G_{r_i}}(v) = r$, $\forall \ \leq i, j \leq k \leq n$, $\forall v \in V$. *Denote* $G'_r = \{G_{r_1} \bigcup \cdots \bigcup G_{r_k}\}$ *the graph obtained by the sequence.*

A graph G is *r-regular* if $\Delta(v) = r$, $\forall v \in V$. A *hamiltonian* path of a graph G is a simple path containing all vertices of G. A set $M \subseteq E$ is a *matching* of G if no two edges in M share a vertex. A *vertex cover* of G is a set $V' \subseteq V$ such that for every edge $(u, v) \in E$ either $u \in V'$ or $v \in V'$. An *edge cover* of G is a set $E' \subseteq E$ such that for every vertex $v \in V$, there is an edge $(v, u) \in E'$.

A matching M of G that is also an edge cover of the graph is called *perfect* matching. Say that an edge $(u, v) \in E$ (resp., a vertex $v \in V$) is *covered* by the vertex cover V' (resp., the edge cover E') if either $u \in V'$ or $v \in V'$ (resp., if there is an edge $(u, v) \in E'$). A set $IS \subseteq V$ is an *independent set* of G if for all vertices $u, v \in IS$, $(u, v) \notin E$.

A *two-factor* graph is a *polynomially computable* r*-factor* graph with $r = 2$. It can be easily seen that there exist exponential many such graph instances. Moreover, these graphs can be recognized in polynomial time and decomposed into a sequence $C_1, \ldots, C_k$, $k \leq n$, in polynomial time via Tutte's reduction [11]. Thus, the class of *polynomially computable* r*-factor* graphs contains an exponential number of graph instances. The problem of finding a maximum matching of any graph can be solved in polynomial time [6].

3 Nash Equilibria in the Path Model

We characterize pure Nash Equilibria of the Path model.

Theorem 1. *For any graph G, $\Pi_{\mathsf{P}}(G)$ has a pure NE if and only if G contains a hamiltonian path.*

Proof. Assume that G contains a hamiltonian path. Then, consider any configuration $\mathbf{s}$ of $\Pi_{\mathsf{P}}(G)$ in which the path player *pp* selects such a path. Observe that path's player selection includes all vertices of G, that the player is satisfied in $\mathbf{s}$. Moreover, any player vp_i, $i \in \mathcal{N}_{vp}$ cannot increase its individual cost since, for all $v \in V(G)$, v is caught by *pp* and, consequently, $\mathsf{IC}_i(\mathbf{s}_{-i}, [v]) = 0$. Thus, $\mathbf{s}$ is a pure NE for $\Pi_{\mathsf{P}}(G)$.

For the contrary, assume that $\Pi_{\mathsf{P}}(G)$, contains a pure NE, $\mathbf{s}^*$, but the graph G does not contain a hamiltonian path. Then, the strategy of the path player, $\mathbf{s}^*_{pp}$, is not a hamiltonian path of G. Thus, there must exist a set of vertices $U \subseteq V$ such that, for any $u \in U$, $u \notin \mathbf{s}^*_{pp}$. Since $\mathbf{s}^*$ is a NE, for all players vp_i, $i \in \mathcal{N}_{vp}$, it must be that $\mathbf{s}^*_i \in U$. Therefore, there is no vertex player located on path $\mathbf{s}^*_{pp}$ which implies that *pp* is not satisfied in $\mathbf{s}^*$; it could increase its individual cost by selecting any path containing at least one vertex player. Thus $\mathbf{s}^*$ is not a NE, which gives a contradiction. $\square$

Corollary 1. *The problem of deciding whether there exists a pure NE for any $\Pi_{\mathsf{P}}(G)$ is $\mathcal{NP}$-complete.*

4 Nash Equilibria in the Edge Model

We proceed to study Nash equilibria in the Edge model. In [4, Theorem 1] it was proved that if G contains more than one edges, then $\Pi_{\mathsf{E}}(G)$ has no pure Nash Equilibrium. For mixed NE, it was proved that:

Theorem 2 (Characterization of Mixed NE). *[4] A mixed strategy profile* $\mathbf{s}$ *is a Nash equilibrium for any $\Pi(G)$ if and only if:*

1. $D_{\mathbf{s}}(ep)$ *is an* edge cover *of* G *and* $D_{\mathbf{s}}(vp)$ *is a* vertex cover *of the graph obtained by* $D_{\mathbf{s}}(ep)$*.*
2. (a) $P_{\mathbf{s}}(Hit(v)) = P_{\mathbf{s}}(Hit(u)) = \min_v P_{\mathbf{s}}(Hit(v)),\ \forall\ u, v \in D_{\mathbf{s}}(vp)$ *and (b)* $\sum_{e \in D_{\mathbf{s}}(ep)} P_{\mathbf{s}}(ep, e) = 1$*.*
3. (a) $m_{\mathbf{s}}(e_1) = m_{\mathbf{s}}(e_2) = \max_e m_{\mathbf{s}}(e),\ \ \forall\ e_1 = (u_1, v_1), e_2 = (u_2, v_2) \in D_{\mathbf{s}}(ep)$ *and (b)* $\sum_{v \in V(D_{\mathbf{s}}(ep))} m_{\mathbf{s}}(v) = \nu$*.*

Here, we provide a estimation on the payoffs of the vertex players in any Nash equilibrium.

Lemma 1. *For any* $\Pi_{\mathsf{E}}(G)$*, a mixed NE,* $\mathbf{s}^*$*, satisfies* $\mathsf{IC}_i(\mathbf{s}^*) = \mathsf{IC}_j(\mathbf{s}^*)$ *and* $1 - \frac{2}{|D_{\mathbf{s}^*}(vp)|} \leq \mathsf{IC}_i(\mathbf{s}^*) \leq 1 - \frac{1}{|D_{\mathbf{s}^*}(vp)|}$*,* $\forall i, j \in \mathcal{N}_{vp}$*.*

4.1 Mixed Nash Equilibria in Various Graphs

Regular, Polynomially Computable *r*-factor and Two-factor graphs

Theorem 3. *For any* $\Pi_{\mathsf{E}}(G)$ *for which* G *is an* r*-regular graph, a mixed NE can be computed in constant time* $O(1)$*.*

Proof. Construct the following configuration $\mathbf{s}^r$ on $\Pi_{\mathsf{E}}(G)$:

$$\begin{aligned}&\text{For any } i \in \mathcal{N}_{vp},\ P_{\mathbf{s}^r}(vp_i, v) := \tfrac{1}{n},\ \forall v \in V(G) \text{ and then set, } \mathbf{s}^r_j := \mathbf{s}^r_i,\\ &\forall j \neq i, j \in \mathcal{N}_{vp}\ .\ \text{Set}\ \ P_{\mathbf{s}^r}(ep, e) := \tfrac{1}{m},\ \forall e \in E.\end{aligned} \tag{3}$$

Obviously, $\mathbf{s}^r$ is a valid (*fully*) mixed strategy profile of $\Pi_{\mathsf{E}}(G)$. We prove that $\mathbf{s}^r$ is a mixed NE for $\Pi_{\mathsf{E}}(G)$. Recall that in any r-regular graph, $m = r \cdot n/2$. By eq. (1) and the construction of $\mathbf{s}^r$, we get, for any $v, u \in V(= D_{\mathbf{s}^r}(vp_i))$, $i \in \mathcal{N}_{VP}$

$$\begin{aligned} IC_i(\mathbf{s}^r_{-i}, [v]) &= 1 - P_{\mathbf{s}}(Hit(v)) = 1 - \frac{|ENeigh(v)|}{m} = 1 - \frac{|ENeigh(u)|}{m} \\ &= IC_i(\mathbf{s}^r_{-i}, [u]) = 1 - \frac{r}{m} = 1 - \frac{2}{n}. \end{aligned} \tag{4}$$

The above result combined with the fact that $D_{\mathbf{s}^r}(vp_i) = V = S_i$, concludes that any vp_i is satisfied in $\mathbf{s}^r$. Now consider the edge player; for any $e = (u, v),\ e' = (u', v') \in E$, by eq. (2) and the construction of $\mathbf{s}^r$, we get

$$\begin{aligned} IC_{ep}(\mathbf{s}^r_{-ep}, [e]) &= \sum_{i \in \mathcal{N}_{VP}} (P_{\mathbf{s}^r}(vp_i, v) + P_{\mathbf{s}^r}(vp_i, u)) = \sum_{i \in \mathcal{N}_{VP}} (P_{\mathbf{s}^r}(vp_i, v') + P_{\mathbf{s}^r}(vp_i, u')) \\ &= IC_{ep}(\mathbf{s}^r_{-ep}, [e']) = \sum_{i \in \mathcal{N}_{VP}} 2 \cdot \frac{1}{n} = \frac{2\nu}{n} \end{aligned} \tag{5}$$

The above result combined with the fact that $D_{\mathbf{s}^r}(ep) = E = S_{ep}$, concludes that ep is also satisfied in $\mathbf{s}^r$ and henceforth $\mathbf{s}^r$ is a mixed NE of $\Pi_{\mathsf{E}}(G)$. It can be easily seen that the time complexity of the assignment $O(1)$. $\square$

Corollary 2. *For any $\Pi_{\mathsf{E}}(G)$ for which G is contains an r-regular factor subgraph, 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.*

Proof. Compute an r-regular factor of G, G_r in polynomial time, denoted as $O(T(G))$. Then apply the mixed strategy profile $\mathbf{s}^r$ described in Theorem 3 on the graph G_r. See [5] for a full proof. □

Proposition 1. *For any $\Pi_{\mathsf{E}}(G)$ for which G is a* two-factor *graph, a mixed NE can be computed in polynomial time, $O(T(G))$, where $O(T(G))$ is the (polynomial) time needed for the decomposition of G into vertex disjoint cycles.*

Perfect Graphs

Theorem 4. *For any $\Pi_{\mathsf{E}}(G)$ for which G has a perfect matching, a mixed NE can be computed in polynomial time, $O(\sqrt{n} \cdot m)$.*

Proof. Compute a perfect matching M of G using a known such algorithm (e.g. [6] and requiring time $O(\sqrt{n} \cdot m)$). Construct the following configuration $\mathbf{s}^p$ on $\Pi_{\mathsf{E}}(G)$:

$$\text{For any } i \in \mathcal{N}_{vp},\ P_{\mathbf{s}^p}(vp_i, v) := \tfrac{1}{n},\ \forall v \in V(G) \text{ and set } \mathbf{s}_j^p := \mathbf{s}_i^p,\ \forall j \neq i, j \in \mathcal{N}_{vp}\ .\ \text{Set } P_{\mathbf{s}^p}(ep, e) := \tfrac{1}{|M|},\ \forall e \in E. \tag{6}$$

Obviously, $\mathbf{s}^p$ is a valid mixed strategy profile of Π_{E}. Note that $|M| = n/2$. We first prove that any $i \in \mathcal{N}_{vp}$ is satisfied in $\mathbf{s}^p$. Note that each vertex of G is hit by exactly one edge of $D_{\mathbf{s}^p}(ep)$. Thus, by eq. (1), for any $i \in \mathcal{N}_{vp}$, $v, u \in V$,

$$\begin{aligned} IC_i(\mathbf{s}_{-i}^p, [v]) &= 1 - P_{\mathbf{s}^p}(Hit(v)) = 1 - P_{\mathbf{s}^p}(Hit(u)) = IC_i(\mathbf{s}_{-i}^p, [u]) \\ &= 1 - \frac{1}{|M|} = 1 - \frac{2}{|n|} \end{aligned}$$

The above result combined with the fact that $D_{\mathbf{s}^p}(vp_i) = V = S_i$ concludes that any vp_i is satisfied in $\mathbf{s}^p$. Now, as it concerns the edge player, note that $\mathsf{IC}_{ep}(\mathbf{s}_{-ep}^p, [e])$ depends only on the strategies of the vertex players in $\mathbf{s}^p$. Furthermore, these strategies are the same as the strategies of the vertex players on configuration $\mathbf{s}^r$ of Theorem 3. Henceforth, using the same arguments as in the theorem we conclude that the edge player is satisfied in $\mathbf{s}^p$. Since both kinds of players are satisfied in $\mathbf{s}^p$, the profile is a mixed NE for Π_{E}. For the time complexity of the assignment, see [5]. □

Trees In Figure 1 we present in pseudocode an algorithm, called $\mathsf{Trees}(\Pi_{\mathsf{E}}(T))$, for computing mixed NE for trees graph instances. Note that in [4], a polynomial time algorithm for finding NE in bipartite graphs is presented. Thus, the same algorithm can apply for trees, since trees are bipartite graphs. However, that algorithm computes a NE of $\Pi_{\mathsf{E}}(T)$ in time $O(n^{2.5}/\sqrt{\log n})$, while the algorithm presented here computes a NE in linear time $O(n)$.

Algorithm $\mathsf{Trees}(\Pi_{\mathsf{E}}(T))$

1. Initialization: $VC := \emptyset$, $EC := \emptyset$, $r := 1$, $T_r := T$.
2. Repeat until $T_r == \emptyset$
   (a) Find the leaves of the tree T_r, $leaves(T_r)$.
   (b) Set $VC := VC \cup leaves(T_r)$.
   (c) For each $v \in leaves(T_r)$ do:
      If $parent_{T_r}(v) \neq \emptyset$, then $EC := EC \cup \{(v, parent_{T_r}(v)))\}$,
      else $EC := EC \cup \{(v, u)\}$, for any $u \in children_T(v)$.
   (d) Update tree: $T_{r+1} := T_r \backslash leaves(T_r) \backslash parents(leaves(T_r))$. Set $r := r + 1$.
3. Define a configuration $\mathbf{s}^t$ with the following support:
   For any $i \in \mathcal{N}_{VP}$, set $D_{\mathbf{s}^t}(vp_i) := VC$ and $D_{\mathbf{s}^t}(ep) := EC$. Then set $D_{\mathbf{s}^t}(vp_j) := D_{\mathbf{s}^t}(vp_i)$, $\forall\, j \neq i$, $j \in \mathcal{N}_{VP}$.
4. Determine the probabilities distributions of players in $\mathbf{s}^t$ as follows:
   $ep :\ \forall\, e \in D_{\mathbf{s}^t}(ep)$, set $P_{\mathbf{s}^t}(ep, e) := 1/|EC|$. Also, $\forall\ e' \in E(T)$, $e' \notin D_{\mathbf{s}^t}(ep)$, set $P_{\mathbf{s}^t}(ep, e') := 0$.
   For any vp_i, $i \in \mathcal{N}_{VP}\ :\ \forall\ v \in D_{\mathbf{s}^t}(vp_i)$, set $P_{\mathbf{s}^t}(vp_i, v) := \frac{1}{|VC|}$. Also, $\forall\ u \notin D_{\mathbf{s}^t}(vp_i)$, set $P_{\mathbf{s}^t}(vp_i, u) := 0$. Then set $\mathbf{s}_j^t = \mathbf{s}_i^t$, $\forall\, j \neq i$, $j \in \mathcal{N}_{VP}$.

Fig. 1. Algorithm $\mathsf{Trees}(\Pi_{\mathsf{E}}(T))$.

Lemma 2. *Set VC, computed by Algorithm* $\mathsf{Trees}(\Pi_{\mathsf{E}}(T))$*, is an independent set of T.*

Lemma 3. *Set EC is an edge cover of T and VC is a vertex cover of the graph obtained by EC.*

Lemma 4. *For all $v \in D_{\mathbf{s}^t}(vp)$, $m_{\mathbf{s}^t}(v) = \frac{\nu}{|D_{\mathbf{s}^t}(vp)|}$. Also, for all $v' \notin D_{\mathbf{s}^t}(vp)$, $m_{\mathbf{s}^t}(v') = 0$.*

Lemma 5. *Each vertex of IS is incident to exactly one edge of EC.*

Proof. By Lemma 3, for each $v \in IS$ there exists at least one edge $e \in EC$ such that $e = (v, u)$. Assume by contradiction that there exists another edge, $(v, u') \in EC$. But since by step **2** of the algorithm for each vertex added in IS we add only one edge incident to it in EC, we get that it must be that $u' \in IS$. However, this contradicts to that IS is an independent set, proved in Lemma 2. □

By Lemmas 3(EC is an edge cover of G) and 5, we can show that:

Lemma 6. *For all $v \in D_{\mathbf{s}^t}(vp)$, $P_{\mathbf{s}}(Hit(v)) = \frac{1}{|D_{\mathbf{s}^t}(ep)|}$. Also, for all $v' \notin D_{\mathbf{s}^t}(vp)$, $P_{\mathbf{s}}(Hit(v')) \geq \frac{1}{|D_{\mathbf{s}^t}(ep)|}$.*

Theorem 5. *For any $\Pi_{\mathsf{E}}(T)$, where T is a tree graph, algorithm* $\mathsf{Trees}(\Pi_{\mathsf{E}}(T))$ *computes a mixed NE in polynomial time $O(n)$.*

Proof. **Correctness**: We prove the computed profile $\mathbf{s}^t$ satisfies all conditions of Theorem 2, thus it is a mixed NE. **1.**: By Lemma 3 . **2.**: By Lemma 6. **3.(a)**: Note that, $D_{\mathbf{s}^t}(vp)$ is an independent set of G and also a vertex cover of $D_{\mathbf{s}^t}(vp)$, by Lemmas 2, 3, respectively. Thus, by Lemma 4, for any $e = (u, v) \in D_{\mathbf{s}^t}(ep)$, we have $m_{\mathbf{s}^t}(e) = m_{\mathbf{s}^t}(v) + m_{\mathbf{s}^t}(u) = \frac{\nu}{|D_{\mathbf{s}^t}(vp)|} + 0$.
3.(b): Since VC is an independent set of G, for any $e = (u, v) \in E$, $e \notin D_{\mathbf{s}^t}(ep)$, $m_{\mathbf{s}^t}(e) = m_{\mathbf{s}^t}(v) + m_{\mathbf{s}^t}(u) \leq \frac{\nu}{|D_{\mathbf{s}^t}(vp)|} = m_{\mathbf{s}^t}(e')$, where $e' \in EC$.
Time Complexity: See [5]. □

4.2 The Price of Anarchy

Lemma 7. *For any $\Pi_{\mathsf{E}}(G)$ and an associated mixed NE $\mathbf{s}^*$, the social cost $\mathsf{SC}(\Pi_{\mathsf{E}}(G), \mathbf{s}^*)$ is upper and lower bounded as follows:*

$$\max\left\{\frac{\nu}{|D_{\mathbf{s}^*}(ep)|}, \frac{\nu}{|V(D_{\mathbf{s}^*}(vp))|}\right\} \leq \mathsf{SC}(\Pi_{\mathsf{E}}(G), \mathbf{s}^*) \leq \frac{\Delta(D_{\mathbf{s}^*}(ep)) \cdot \nu}{|D_{\mathbf{s}^*}(ep)|} \qquad (7)$$

These bounds are tight.

Theorem 6. *The Price of Anarchy for the Edge model is $\frac{n}{2} \leq \mathsf{r}(\mathsf{E}) \leq n$.*

References

1. Alon, N., Karp, R. M., Peleg, D., West, D.: A Graph-Theoretic Game and its Application to the k-Server Problem. SIAM Journal on Computing **24**(1) (1995) 78-100
2. Aspnes, J., Chang, K., A. Yampolskiy: Inoculation Strategies for Victims of Viruses and the Sum-of-Squares Problem. Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (2005) 43-52
3. Koutsoupias, E., Papadimitriou, C. H.: Worst-Case Equilibria. In Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, Vol. 1563. Springer-Verlag, 1999 404–413
4. Mavronicolas, M., Papadopoulou, V., Philippou, A., Spirakis, P.: A Network Game with Attacker and Protector Entities. In the Proceedings of the 16th Annual International Symposium on Algorithms and Computation (2005)
5. Mavronicolas, M., Papadopoulou, V., Philippou, A., Spirakis, P.: A Graph-Theoretic Network Security Game. *TR-09-05*, University of Cyprus (2005)
6. S. Micali and V.V. Vazirani, "An $O(\sqrt{V}E)$ Algorithm for Finding Maximum Matching in General Graphs", *Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science*, pp. 17-27, 1980.
7. Nash, J. F.: Equilibrium Points in n-Person Games. Proceedings of the National Acanemy of Sciences of the United States of America **36** (1950) 48-49
8. Nash, J. F.: Noncooperative Games. Annals of Mathematics **54**(2) (1951) 286-295
9. M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.
10. C. H. Papadimitriou: Algorithms, Games, and the Internet. Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (2001) 749-753
11. W. T. Tutte, "A Short Proof of the Factor Theorem for Finite Graphs", *Canadian Journal of Mathematics*, Vol 6, pp. 347-352, 1954.