A Graph-Theoretic Network Security Game *

Marios Mavronicolas[†], Vicky Papadopoulou[*†], Anna Philippou[†] and Paul Spirakis[‡]

Abstract

Consider a network vulnerable to viral infection. The system security software can guarantee safety only to a limited part of the network. Such limitations result from economy costs or processing costs. The problem raised is to *which* part of the network the security software should be installed, so that to secure as much as possible 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. For these two games, we are interested in the associated *Nash equilibria*, 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 [7].
- In [7] a characterization of mixed Nash equilibria for the Edge model is provided. However, that characterization only implies an exponential time algorithm for the general case. Here, combining it with clever exploration of properties of various practical families of graphs, 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* [6] 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.

Throughout the paper, some missing proofs can be found in the attached Appendix. It may be read at the discretion of the Program Committee.

*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).

†Department of Computer Science, University of Cyprus, P.O. Box 20537, Nicosia CY-1678, Cyprus. Email: {mavronic,viki,annap}@ucy.ac.cy

‡Department of Computer Engineering and Informatics, University of Patras, Rion, 26500 Patras, Greece. Email: spirakis@cti.gr

1 Introduction

Motivation. Although *Network Security* has been always considered to be a critical issue in networks, the recent huge growth of public networks (e.g. the Internet) made it even more very important [14]. This work consider a branch of this area, related to the protection of a system from harmful procedures (e.g. viruses, worms, eavedroppers [3]). Consider an information network where the nodes of the network are insecure and vulnerable to infection such as, viruses, Trojan horses, eavedroppers, 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. Such limitations result from money and system performance costs caused in order to purchase a global security software or by the reduced efficiency or usability of a protected node. At any time, a number of harmful entities is known (or an upper bound of this number) to be present in the network. 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* [10, 11] 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 3.2). This opposes to that, the simpler case of this model, i.e. the Edge model posses no pure Nash equilibrium, proved in [7].

- In [7], we also provide a graph-theoretic characterization of mixed Nash Equilibria for the Edge model, relating them to graph-theoretic notions, such as edge covers and vertex covers of the graph (Theorem 4.6, here). 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 (Theorems 4.7, 4.8, 4.9 and 4.15, respectively).

- We measure the system performance related to the scenario considered utilizing the notion of the *social cost* [6]. 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 4.17).

Related Work. This work is a step further in the development of the new born area of *Algorithmic Game Theory*. At the same time, it contributes in the subfield of *Network Security*, related to the protection of a system from harmful procedures (e.g. viruses, worms, malicious procedures, or eavedroppers [3]). This work is one of the only few works to model *network security problems* as a strategic game. The most of these work consider *Interdependent Security* games, [5],[2]. In such a game, a large number of players must make individual investment decisions related to security, in which the ultimate safety of each participant may depend in a complex way on the actions of the entire population. Another related work is that of [3], which studies the feasibility and computational complexity of two privacy tasks in distributed environments with mobile *eavesdroppers*; of distributed database maintenance and message transmission. We remark however, 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 considers inoculation strategies for victims of viruses and establishes 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 and provide an approximate solution for the simple *network design* problem. In a recent work of ours [7], 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 then focus on mixed Nash equilibria where we compute a polynomial time algorithm for their computation for bipartite graphs.
Finally, our results contribute towards answering the general question of Papadimitriou [13] 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. A graph H, is an *induced* subgraph of G, if $V(H) \subseteq V(G)$ and $(u, v) \in E(H)$, whenever $(u, v) \in E(G)$. 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 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 $parents(A) := \{u \in V : u = father(v),\ v \in A\}$. For all above properties of a graph G, when there is no confusion, we omit G.

2.1 Protector-Attacker models

Definition 2.1 *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 *$\vec{\mathbf{s}} = \langle s_1, \ldots, s_{|\mathcal{N}_{VP}|}, s_p \rangle \in \mathcal{S}$, also called a* configuration.

– *The* Individual Cost *of vertex player* vp_i *is a function* $\mathsf{IC}_i : \mathcal{S} \rightarrow \{0,1\}$ *such that* $\mathsf{IC}_i(\vec{\mathbf{s}}) = \begin{cases} 0, & s_i \in s_p \\ 1, & s_i \notin s_p \end{cases}$ *; 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* $\mathsf{IC}_p : \mathcal{S} \rightarrow \mathbb{N}$ *such that* $\mathsf{IC}_p(\vec{\mathbf{s}}) = |\{s_i : s_i \in s_p\}|$.

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 $\vec{\mathbf{s}}$ is a *pure Nash equilibrium* [10, 11] (abbreviated as *pure NE*) if for each player $i \in \mathcal{N}$, it minimizes IC_i over all configurations $\vec{\mathbf{t}}$ that differ from $\vec{\mathbf{s}}$ 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 player $i \in \mathcal{N}$ is a probability distribution over its strategy set S_i; thus, 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* $\vec{\mathbf{s}}$ is a collection of mixed strategies, one for each player. Denote $P_{\vec{\mathbf{s}}}(ep, e)$ the probability that edge player ep chooses edge $e \in E(G)$ in $\vec{\mathbf{s}}$; denote $P_{\vec{\mathbf{s}}}(vp_i, v)$ the probability that player vp_i chooses vertex $v \in V$ in $\vec{\mathbf{s}}$. Note $\sum_{v \in V} P_{\vec{\mathbf{s}}}(vp_i, v) = 1$ for each player vp_i; similarly, $\sum_{e \in E} P_{\vec{\mathbf{s}}}(ep, e) = 1$. Denote $P_{\vec{\mathbf{s}}}(vp, v) = \sum_{i \in \mathcal{N}_{VP}} P_{\vec{\mathbf{s}}}(vp_i, v)$ the probability that vertex v is chosen by some vertex player in $\vec{\mathbf{s}}$. The *support* of a player $i \in \mathcal{N}$ in the configuration $\vec{\mathbf{s}}$, denoted $D_{\vec{\mathbf{s}}}(i)$, is the set of pure strategies in its strategy set to which i assigns strictly positive probability in $\vec{\mathbf{s}}$. Denote $D_{\vec{\mathbf{s}}}(vp) = \bigcup_{i \in \mathcal{N}_{VP}} D_{\vec{\mathbf{s}}}(i)$; so, $D_{\vec{\mathbf{s}}}(vp)$ contains all pure strategies (that is, vertices) to which some vertex player assigns strictly positive probability. Let also $ENeigh_{\vec{\mathbf{s}}}(v) = \{(u,v) \in E : (u,v) \in D_{\vec{\mathbf{s}}}(ep)\}$; that is $ENeigh_{\vec{\mathbf{s}}}(v)$ contains all edges incident to v that are included in the support of the edge player in $\vec{\mathbf{s}}$. Given a mixed strategy profile $\vec{\mathbf{s}}$, we denote $(\vec{\mathbf{s}}_{-x}, [y])$ a configuration obtained by $\vec{\mathbf{s}}$, where all but player x play as in $\vec{\mathbf{s}}$ and player x plays the pure strategy y.

A mixed strategic profile $\vec{\mathbf{s}}$ induces an *Expected Individual Cost* IC_i for each player $i \in \mathcal{N}$, which is the expectation, according to $\vec{\mathbf{s}}$, of its corresponding Individual Cost (defined previously for pure strategy profiles). The mixed strategy profile $\vec{\mathbf{s}}$ is a *mixed Nash equilibrium* [10, 11] (abbreviated as mixed NE) if for each player $i \in \mathcal{N}$, it minimizes IC_i over all configurations $\vec{\mathbf{t}}$ that differ from $\vec{\mathbf{s}}$ only with respect to the mixed strategy of player i. We denote such a strategy profile as $\vec{\mathbf{s}^*}$. Denote $BR_{\vec{\mathbf{s}}}(x)$ the set of *best response (pure) strategies* of player x in a mixed strategy profile $\vec{\mathbf{s}}$, that is, $\mathsf{IC}_x(\vec{\mathbf{s}}_{-x}, y) \geq \mathsf{IC}_x(\vec{\mathbf{s}}_{-x}, y')$, $\forall\, y \in BR_{\vec{\mathbf{s}}}(x)$ and $y' \notin BR_{\vec{\mathbf{s}}}(x)$, $y' \in S_x$, where S_x is the strategy set of player x (see also [12], 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 $\vec{\mathbf{s}}$. For each vertex $v \in V$, denote $Hit(v)$ the event that the edge player hits vertex v. So, the probability (according to $\vec{\mathbf{s}}$) of $Hit(v)$ is $P_{\vec{\mathbf{s}}}(Hit(v)) = \sum_{e \in ENeigh(v)} P_{\vec{\mathbf{s}}}(ep, e)$. Define the minimum hitting probability $P_{\vec{\mathbf{s}}}$ as $\min_v P_{\vec{\mathbf{s}}}(Hit(v))$. For each vertex $v \in V$, denote $m_{\vec{\mathbf{s}}}(v)$ the expected number of vertex players choosing v (according to $\vec{\mathbf{s}}$). For each edge $e = (u,v) \in E$, denote $m_{\vec{\mathbf{s}}}(e)$ the expected number of vertex players choosing either u or v; so, $m_{\vec{\mathbf{s}}}(e) = m_{\vec{\mathbf{s}}}(u) + m_{\vec{\mathbf{s}}}(v)$. It is easy to see that for each vertex $v \in V$, $m_{\vec{\mathbf{s}}}(v) = \sum_{i \in \mathcal{N}_{VP}} P_{\vec{\mathbf{s}}}(vp_i, v)$. Define the maximum expected number of vertex players choosing e in $\vec{\mathbf{s}}$ as $\max_e m_{\vec{\mathbf{s}}}(e)$.

We proceed to calculate the Expected Individual Cost. Clearly, for the vertex player $vp_i \in \mathcal{N}_{VP}$,

$$\mathsf{IC}_i(\vec{\mathbf{s}}) = \sum_{v \in V(G)} P_{\vec{\mathbf{s}}}(vp_i, v) \cdot (1 - P_{\vec{\mathbf{s}}}(Hit(v)) = \sum_{v \in V(G)} \left(P_{\vec{\mathbf{s}}}(vp_i, v) \cdot (1 - \sum_{e \in ENeigh(v)} P_{\vec{\mathbf{s}}}(ep, e) \right) \quad (1)$$

For the edge player ep,

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

Social Cost and Price of Anarchy. We utilize the notion of *social cost* [6] for evaluating the system performance. A natural such measurement for the problem studied is the number of attackers catch by the system protector; a maximization of this quantity maximizes system's performance in respect to its safety from harmful entities. We therefore define,

Definition 2.2 *For model* M, $\mathsf{M} = \{\mathsf{P}, \mathsf{E}\}$, *we define the* social cost *of configuration* $\vec{\mathbf{s}}$ *on instance* $\Pi_{\mathsf{M}}(G)$, $\mathsf{SC}(\Pi_{\mathsf{M}}, \vec{\mathbf{s}})$, *to be the sum of vertex players of* Π_{M} *arrested in* $\vec{\mathbf{s}}$. *That is,* $\mathsf{SC}(\Pi_{\mathsf{M}}, \vec{\mathbf{s}}) = \mathsf{IC}_p(\vec{\mathbf{s}})$ *(*$p = \{pp, vp\}$*, when* $\mathsf{M} = \mathsf{P}$ *and* $\mathsf{M} = \mathsf{E}$*, respectively). The system wishes to maximize the social cost.*

Definition 2.3 *For model* M, $\mathsf{M} = \{\mathsf{P}, \mathsf{E}\}$, *we define the* price of anarchy, $\mathsf{r}(\mathsf{M})$ *to be,*

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

2.2 Background from Graph Theory

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

Definition 2.4 *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, $1 \le k \le n$. *That is,* $\bigcup_{1\le i\le k} V(G_{r_i}) = V(G)$, $V(G_{r_i})\bigcap V(G_{r_j}) = \emptyset$ *and* $\Delta_{G_{r_i}}(v) = r$, $\forall\ \le i,j \le k \le n$, $\forall v \in V$. *Let* $G'_r = \{G_{r_1}\bigcup\cdots\bigcup G_{r_k}\}$ *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 $(u,v) \in E'$). Otherwise, the edge (resp., the vertex) is not *covered* by the vertex cover (resp., the edge cover). 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 $r = 2$. Thus, its vertices can be partitioned into disjoint sets $C_1, \ldots, C_k$, $k \le n$ each of which induces a cycle in G. *Two-factor* graphs can be recognized in polynomial time and decomposed into a sequence $C_1, \ldots, C_k$, $k \le n$, in polynomial time via Tutte's reduction [15] (see also [9] for a survey in relevant problems). It can be easily observed that there exists exponentially many such graph instances. Henceforth, 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 [8].

3 Nash Equilibria in the Path Model

We provide the following characterization of pure Nash Equilibria in the Path model.

Theorem 3.1 *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 $\vec{\mathbf{s}}$ 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 $\vec{\mathbf{s}}$. 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, $\mathsf{IC}_i(\vec{\mathbf{s}}_{-i}, [v]) = 0$. Thus, $\vec{\mathbf{s}}$ is a pure NE for $\Pi_{\mathsf{P}}(G)$.

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

This characterization implies the following result regarding the existence of pure NE.

Corollary 3.2 *The problem of deciding whether there exists a pure NE for any $\Pi_{\mathsf{P}}(G)$ is $\mathcal{NP}$-complete.*

Proof. *Membership in class $\mathcal{NP}$*: Guess a configuration $\vec{\mathbf{s}}$. We may decide whether $\vec{\mathbf{s}}$ is a pure NE in polynomial time as follows: By Theorem 3.1, $\vec{\mathbf{s}}$ is a pure NE if and only if $\vec{\mathbf{s}}_{pp}$ is a hamiltonian path of G. This can be verified in polynomial time.
$\mathcal{NP}$-completeness: We reduce from the problem of deciding whether a graph G contains a hamiltonian path to our problem. The first problem is known to be $\mathcal{NP}$-complete [4]. By Theorem 3.1, the proof is completed. ■

4 Nash Equilibria in the Edge Model

We proceed to study Nash equilibria in the Edge model. Beginning with pure NE, we have the following result, proved in [7].

Theorem 4.1 *[7] If G contains more than one edges, then $\Pi_{\mathsf{E}}(G)$ has no pure Nash Equilibrium.*

Next we present a characterization of mixed Nash equilibria, obtained via a series of claims. The characterization also appears in [7]. It is included here for completeness reasons. We begin with a characterization of the support sets of the vertex and edge players in a mixed NE.

Claim 4.2 *In any mixed NE, $\vec{\mathbf{s}^*}$, of $\Pi_{\mathsf{E}}(G)$, $D_{\vec{\mathbf{s}^*}}(ep)$ is an edge cover of G.*

Proof. Assume the contrary. Let NC be a set of vertices of G not covered by $D_{\vec{\mathbf{s}^*}}(ep)$. Then, for all $i \in \mathcal{N}_{VP}$, $D_{\vec{\mathbf{s}^*}}(i) \subseteq NC$ since these strategies maximize the individual cost of the player. This implies an individual cost of 0 for the edge player which the player can unilaterally improve by selecting any edge containing at least one vertex player. Thus, this strategy profile is not a mixed NE, a contradiction. ■

Claim 4.3 *In any mixed NE, $\vec{\mathbf{s}^*}$, of $\Pi_{\mathsf{E}}(G)$, $D_{\vec{\mathbf{s}^*}}(vp)$ is a vertex cover of the graph obtained by $D_{\vec{\mathbf{s}^*}}(ep)$.*

Proof. Assume the contrary. Let $e \in D_{\vec{\mathbf{s}^*}}(ep)$ be an edge not covered by $D_{\vec{\mathbf{s}^*}}(vp)$. Then, the gain of the edge player ep on e is zero (since there is no vertex player on it). Thus, ep can gain more by moving the probability of choosing edge e to an edge with at least one endpoint *covered* by $D_{\vec{\mathbf{s}}^*}(vp)$) (obviously such an edge exists, because the vertex players have to be somewhere on V). Thus, this strategy profile is not be a mixed NE, a contradiction. ■

The next result provides an estimation on the payoffs of the vertex players in any Nash equilibrium.

Claim 4.4 *For any* $\Pi_{\mathsf{E}}(G)$*, a mixed NE,* $\vec{\mathbf{s}}^*$*, satisfies* $\mathsf{IC}_i(\vec{\mathbf{s}}^*) = \mathsf{IC}_j(\vec{\mathbf{s}}^*)$ *and* $1 - \frac{2}{|D_{\vec{\mathbf{s}}^*}(vp)|} \leq \mathsf{IC}_i(\vec{\mathbf{s}}^*) \leq 1 - \frac{1}{|D_{\vec{\mathbf{s}}^*}(vp)|}$*,* $\forall i, j \in \mathcal{N}_{VP}$*.*

Proof. Obviously, by equation (1), in any mixed strategy profile $\vec{\mathbf{s}}$, for all $i, j \in \mathcal{N}_{VP}$ and $v \in V$

$$\mathsf{IC}_i(\vec{\mathbf{s}}_{-i}, [v]) = \mathsf{IC}_j(\vec{\mathbf{s}}_{-j}, [v]) = 1 - P_{\vec{\mathbf{s}}}(Hit(v)) \tag{3}$$

By the definition of a NE, we may see that, for all $v \in D_{\vec{\mathbf{s}}^*}(vp_i)$, $\mathsf{IC}_i(\vec{\mathbf{s}}^*) = \mathsf{IC}_i(\vec{\mathbf{s}}^*_{-i}, [v])$. So, by equation (3), $\mathsf{IC}_i(\vec{\mathbf{s}}^*) = \mathsf{IC}_i(\vec{\mathbf{s}}^*_{-i}, [v]) = \mathsf{IC}_j(\vec{\mathbf{s}}^*_{-j}, [v]) = \mathsf{IC}_j(\vec{\mathbf{s}}^*)$.

Thus, consider any $i \in \mathcal{N}_{VP}$, $\mathsf{IC}_i(\vec{\mathbf{s}}^*)$. For any $v \in D_{\vec{\mathbf{s}}^*}(vp)$,

$$\begin{aligned}
\mathsf{IC}_i(\vec{\mathbf{s}}^*) &= 1 - P_{\vec{\mathbf{s}}}(Hit(v)) = 1 - \sum_{e \in ENeigh(v)} P_{\vec{\mathbf{s}}^*}(ep, e)) \Leftrightarrow \\
\sum_{v \in D_{\vec{\mathbf{s}}^*}(vp)} \mathsf{IC}_i(\vec{\mathbf{s}}^*) &= \sum_{v \in D_{\vec{\mathbf{s}}^*}(vp)} (1) - \sum_{v \in D_{\vec{\mathbf{s}}^*}(vp)} \left(\sum_{e \in ENeigh(v)} P_{\vec{\mathbf{s}}^*}(ep, e) \right) \qquad (4)
\end{aligned}$$

By Claim 4.3, $D_{\vec{\mathbf{s}}^*}(vp)$ is a vertex cover of $D_{\vec{\mathbf{s}}^*}(ep)$. Therefore, any edge of $D_{\vec{\mathbf{s}}^*}(ep)$ appears either once or twice in the sum $\sum_{v \in D_{\vec{\mathbf{s}}^*}(vp)}$, once for each of its endpoints. Combining this with equation (4) we get the following bounds:

$$\begin{aligned}
|D_{\vec{\mathbf{s}}^*}(vp)| - 2 \sum_{e \in D_{\vec{\mathbf{s}}^*}(ep)} P_{\vec{\mathbf{s}}^*}(ep, e) \leq |D_{\vec{\mathbf{s}}^*}(vp)| \cdot \mathsf{IC}_i(\vec{\mathbf{s}}^*) &\leq |D_{\vec{\mathbf{s}}^*}(vp)| - \sum_{e \in D_{\vec{\mathbf{s}}^*}(ep)} P_{\vec{\mathbf{s}}^*}(ep, e) \\
|D_{\vec{\mathbf{s}}^*}(vp)| - 2 \leq |D_{\vec{\mathbf{s}}^*}(vp)| \cdot \mathsf{IC}_i(\vec{\mathbf{s}}^*) &\leq |D_{\vec{\mathbf{s}}^*}(vp)| - 1 \\
1 - \frac{2}{|D_{\vec{\mathbf{s}}^*}(vp)|} \leq \mathsf{IC}_i(\vec{\mathbf{s}}^*) &\leq 1 - \frac{1}{|D_{\vec{\mathbf{s}}^*}(vp)|}
\end{aligned}$$

■

By Claim 4.4 we get the following observation.

Observation 4.1 *For any* $\Pi_{\mathsf{E}}(G)$ *and* $i \in \mathcal{N}_{VP}$*,* $IC_i(\vec{\mathbf{s}}^*)$ *is* maximized *in the mixed Nash equilibrium,* $\vec{\mathbf{s}}^*$*, which minimizes* $D_{\vec{\mathbf{s}}^*}(vp)$ *over all mixed NE of* $\Pi_{\mathsf{E}}(G)$*.*

Claim 4.5 *In any mixed NE of* $\Pi_{\mathsf{E}}(G)$*,* $\vec{\mathbf{s}}^*$*,* $\sum_{v \in V(D_{\vec{\mathbf{s}}^*}(ep))} m_{\vec{\mathbf{s}}^*}(v) = \nu$*.*

Proof.

$$\begin{aligned}
\sum_{v \in V(D_{\vec{\mathbf{s}}^*}(ep))} m_{\vec{\mathbf{s}}^*}(v) &= \sum_{v \in V(D_{\vec{\mathbf{s}}^*}(ep))} \sum_{i \in \mathcal{N}_{VP}} P_{\vec{\mathbf{s}}^*}(vp_i, v) = \sum_{i \in \mathcal{N}_{VP}} \sum_{v \in V(D_{\vec{\mathbf{s}}^*}(ep))} P_{\vec{\mathbf{s}}^*}(vp_i, v) \\
&\overset{(Claim\ 4.2)}{=} \sum_{i \in \mathcal{N}_{VP}} \sum_{v \in V} P_{\vec{\mathbf{s}}^*}(vp_i, v) = \sum_{i \in \mathcal{N}_{VP}} 1 = \nu
\end{aligned}$$

■

The characterization of mixed NE follows:

Theorem 4.6 *(***Characterization of Mixed NE***) A mixed strategy profile* $\vec{\mathbf{s}}$ *is a Nash equilibrium for any* $\Pi_{\mathsf{E}}(G)$ *if and only if:*

1. $D_{\vec{s}}(ep)$ *is an* edge cover *of* G *and* $D_{\vec{s}}(vp)$ *is a* vertex cover *of the graph induced by* $D_{\vec{s}}(ep)$.

2. *The probability distribution of the edge player over* E, *is such that, (a)* $P_{\vec{s}}(Hit(v)) = P_{\vec{s}}(Hit(u)) = \min_v P_{\vec{s}}(Hit(v))$, $\forall\, u, v \in D_{\vec{s}}(vp)$, *(b)* $P_{\vec{s}}(Hit(v)) \leq P_{\vec{s}}(Hit(u))$, *for any* $v, u \in V$, $v \in D_{\vec{s}}(vp)$, $u \notin D_{\vec{s}}(vp)$ *and (c)* $\sum_{e \in D_{\vec{s}}(ep)} P_{\vec{s}}(ep, e) = 1$.

3. *The probability distributions of the vertex players over* V *are such that, (a)* $m_{\vec{s}}(e_1) = m_{\vec{s}}(e_2) = \max_e m_{\vec{s}}(e)$, $\forall\, e_1, e_2 \in D_{\vec{s}}(ep)$, *(b)* $m_{\vec{s}}(e_1) \geq m_{\vec{s}}(e_2)$, $\forall\, e_1,\, e_2 \in E$, $e_1 \in D_{\vec{s}}(ep)$, $e_2 \notin D_{\vec{s}}(ep)$ *and (c)* $\sum_{v \in V(D_{\vec{s}}(ep))} m_{\vec{s}}(v) = \nu$.

Proof. We first prove that if $\vec{\mathbf{s}^*}$ is a mixed NE for $\Pi_{\mathsf{E}}(G)$ then conditions 1-3 hold. **1.**: By Claims 4.2 and 4.3. **2.(a)**: By Claim 4.4, for any $u, v \in D_{\vec{\mathbf{s}^*}}(vp)$, $u, v \in BR_{\vec{\mathbf{s}^*}}(vp_i)$. Moreover, by the same claim and since $\vec{\mathbf{s}^*}$ is a Nash equilibrium, for any $i, j \in \mathcal{N}_{VP}$, we get $\mathsf{IC}_i(\vec{\mathbf{s}^*}) = \mathsf{IC}_j(\vec{\mathbf{s}}) = \mathsf{IC}_i(\vec{\mathbf{s}}^*_{-i}, [v]) = 1 - P_{\vec{\mathbf{s}^*}}(Hit(v)) = 1 - P_{\vec{\mathbf{s}^*}}(Hit(u))$. Thus, $P_{\vec{\mathbf{s}^*}}(Hit(v)) = P_{\vec{\mathbf{s}^*}}(Hit(u)) = \min_v P_{\vec{\mathbf{s}}}(Hit(v))$. **2.(b)**: Using the same arguments as in **2.(a)**, for any $i \in \mathcal{N}_{VP}$, $v \in D_{\vec{\mathbf{s}^*}}(vp) \subseteq BR_{\vec{\mathbf{s}^*}}(vp_i)$, $u \in V$, $u \notin D_{\vec{\mathbf{s}^*}}(vp)$, we get $IC_i(\vec{\mathbf{s}^*}, [v]) \geq IC_i(\vec{\mathbf{s}^*}, [u])$. Thus, by equation (1), $1 - P_{\vec{\mathbf{s}^*}}(Hit(v)) \geq 1 - P_{\vec{\mathbf{s}^*}}(Hit(u))$ and $P_{\vec{\mathbf{s}^*}}(Hit(v)) \leq P_{\vec{\mathbf{s}^*}}(Hit(u))$. **2.(c)**: Obvious, since $P_{\vec{\mathbf{s}^*}}(ep)$ is a probability distribution over E. **3.(a) and (b)**: by equation (2) and the fact that $\vec{\mathbf{s}^*}$ is a Nash equilibrium. **3.(c)**: by Claim 4.5.

The proof that any mixed strategy profile $\vec{\mathbf{s}}$ of $\Pi_{\mathsf{E}}(G)$ satisfying conditions 1-3 is a Nash equilibrium, is obvious. ■

Remark. Note that the characterization implies no polynomial time algorithm for computing a mixed Nash equilibrium (unless $\mathcal{P} = \mathcal{NP}$), since it involves solving a mixed integer programming problem.

4.1 Mixed Nash Equilibria in Various Graphs

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

Theorem 4.7 *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 $\vec{\mathbf{s}^{\mathbf{r}}}$ on $\Pi_{\mathsf{E}}(G)$:

$$\left.\begin{array}{l} \text{For any } i \in \mathcal{N}_{VP},\ P_{\vec{\mathbf{s}^{\mathbf{r}}}}(vp_i, v) = \frac{1}{n},\ \forall v \in V(G). \text{ Then set, } \vec{\mathbf{s}^{\mathbf{r}}}_j = \vec{\mathbf{s}^{\mathbf{r}}}_i,\ \forall j \neq i, j \in \mathcal{N}_{VP} \text{ and} \\ \qquad P_{\vec{\mathbf{s}^{\mathbf{r}}}}(ep, e) = \frac{1}{m},\ \forall e \in E. \end{array}\right\} \quad (5)$$

Obviously, $\vec{\mathbf{s}^{\mathbf{r}}}$ is a valid (*fully*) mixed strategy profile of Π_{E}. Observe that the assignment can be accomplished in time $O(1)$; we take constant time to compute $P_{\vec{\mathbf{s}^{\mathbf{r}}}}$ for a vertex player i and the edge player. We prove that $\vec{\mathbf{s}^{\mathbf{r}}}$ is a mixed NE for Π_{E}.

By equation (1) and the construction of $\vec{\mathbf{s}^{\mathbf{r}}}$, we get, for any $v, u \in V (= D_{\vec{\mathbf{s}^{\mathbf{r}}}}(vp_i))$, $i \in \mathcal{N}_{VP}$

$$IC_i(\vec{\mathbf{s}^{\mathbf{r}}}_{-i}, [v]) \;=\; 1 - P_{\vec{\mathbf{s}}}(Hit(v)) = 1 - \sum_{e \in ENeigh(v)} P_{\vec{\mathbf{s}^{\mathbf{r}}}}(ep, e) = 1 - \frac{r}{|E|} = 1 - \frac{2}{n} = IC_i(\vec{\mathbf{s}^{\mathbf{r}}}_{-i}, [u]).$$

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

$$IC_{ep}(\vec{\mathbf{s}^{\mathbf{r}}}_{-ep}, [e]) \;=\; m_e(\vec{\mathbf{s}^{\mathbf{r}}}) = \sum_{i \in \mathcal{N}_{VP}} \left(P_{\vec{\mathbf{s}^{\mathbf{r}}}}(vp_i, v) + P_{\vec{\mathbf{s}^{\mathbf{r}}}}(vp_i, u)\right) = \sum_{i \in \mathcal{N}_{VP}} 2 \cdot \frac{1}{n} = \frac{2\nu}{n} = IC_{ep}(\vec{\mathbf{s}^{\mathbf{r}}}_{-ep}, [e'])$$

The above result combined with the fact that $D_{\vec{\mathbf{s}^{\mathbf{r}}}}(ep) = E = S_{ep}$, concludes that ep is also satisfied in $\vec{\mathbf{s}^{\mathbf{r}}}$ and henceforth $\vec{\mathbf{s}^{\mathbf{r}}}$ is a mixed NE of Π_{E}. ■

The above result can be extended to *polynomially computable* r*-factor* and *two-factor* graphs.

Corollary 4.8 *For any* $\Pi_\mathsf{E}(G)$ *for which* G *is a* polynomially computable r-factor *graph, a mixed NE can be computed in polynomial time,* $O(1 + T(G))$, *where* $O(T(G))$ *is the (polynomial) time needed for the decomposition of* G *into vertex disjoint* r*-regular subgraphs.*

Proof. Compute a sequence of k vertex disjoint r-regular subgraphs $G_{r_1}, \ldots, G_{r_k}$ of G, for an integer k, $1 \leq k \leq n$ covering all vertices of G. This can be performed in polynomial time, by the definition of *polynomially computable* r*-factor* graphs, denoted as $O(T(G))$. Consider the subgraph of G induced by the sequence $G'_r = \{\bigcup_{1 \leq i \leq k} G_{r_i}\}$. Note that G'_r is a (none necessarily connected) r-regular graph. We then apply the mixed strategy profile $\vec{\mathbf{s^r}}$ described in Theorem 4.7 on the graph G'_r. Now, noting that $D_{\vec{\mathbf{s^r}}} = V$, we can easily show that all equations of Theorem 4.7 for configuration $\vec{\mathbf{s^r}}$ also hold for graph G. ■

Observation 4.2 *For any* $\Pi_\mathsf{E}(G)$ *for which* G *is a* two-factor *graph, a mixed NE can be computed in polynomial time,* $O(m + T(G))$, *where* $O(T(G))$ *is the (polynomial) time needed for the decomposition of* G *into vertex disjoint cycles.*

4.1.2 Perfect Graphs

Theorem 4.9 *For any* $\Pi_\mathsf{E}(G)$ *for which* G *has a perfect matching, a mixed NE can be computed in linear time,* $O(\sqrt{n} \cdot m)$.

Proof. Compute a perfect matching M of G. Construct the following configuration $\vec{\mathbf{s^f}}$ on $\Pi_\mathsf{E}(G)$:

$$\left.\begin{array}{r}\text{For } any\ i \in \mathcal{N}_{VP},\ P_{\vec{\mathbf{s^f}}}(vp_i, v) = \frac{1}{n},\ \forall v \in V(G). \text{ Then set, } \vec{\mathbf{s^f}}_j = \vec{\mathbf{s^f}}_i,\ \forall j \neq i, j \in \mathcal{N}_{VP} \text{ and} \\ P_{\vec{\mathbf{s^f}}}(ep, e) = \frac{1}{|M|},\ \forall e \in E.\end{array}\right\} \quad (6)$$

Obviously, $\vec{\mathbf{s^f}}$ is a valid mixed strategy profile of Π_E. Recall that $|M| = n/2$. For the time complexity of the assignment, note first that a maximum matching of a graph can be computed in time $O(\sqrt{n} \cdot m)$ [8]. Furthermore, the assignment of (6) can be accomplished in time $O(1)$; we take constant time to compute $P_{\vec{\mathbf{s^f}}}$ for a vertex player i and the edge player. Thus, in total we used $O(\sqrt{n} \cdot m)$ time. Note also that $\vec{\mathbf{s^f}}$ is a *fully* mixed strategy profile as it concerns the vertex players but not the edge player, i.e. $D_{\vec{\mathbf{s^f}}}(vp_i) = V$, $\forall i \in \mathcal{N}_{VP}$ and $D_{\vec{\mathbf{s^f}}}(ep) = M$. We first prove that any $i \in \mathcal{N}_{VP}$ is satisfied in $\vec{\mathbf{s^f}}$. Note that each vertex of G is hit by exactly one edge of $D_{\vec{\mathbf{s^f}}}(ep)$. By equation (1) we get that, for any $i \in \mathcal{N}_{VP}$, $v, u \in V$,

$$IC_i(\vec{\mathbf{s^f}}_{-i}, [v]) \;=\; 1 - P_{\vec{\mathbf{s^f}}}(Hit(v)) = 1 - \frac{1}{|M|} = 1 - \frac{2}{|n|} = IC_i(\vec{\mathbf{s^f}}_{-i}, [u])$$

The above result combined with the fact that $D_{\vec{\mathbf{s^f}}}(vp_i) = V = S_i$ concludes that any vp_i is satisfied in $\vec{\mathbf{s^f}}$. Now, as it concerns the edge player, recall that by equation (2),

$$IC_{ep}(\vec{\mathbf{s^f}}_{-ep}, [e]) = \sum_{i \in \mathcal{N}_{VP}} \left(P_{\vec{\mathbf{s^f}}}(vp_i, v) + P_{\vec{\mathbf{s^f}}}(vp_i, u) \right), \forall e = (u, v) \in E,$$

thus the function depends only on the strategies of the vertex players in $\vec{\mathbf{s^f}}$. Furthermore, these strategies are the same as the strategies of the vertex players on configuration $\vec{\mathbf{s^r}}$ of Theorem 4.7. Henceforth, using the same arguments as in the theorem we conclude that the edge player is satisfied in $\vec{\mathbf{s^f}}$. Since both kinds of players are satisfied in $\vec{\mathbf{s^f}}$, the profile is a mixed NE for Π_E. ■

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 $\vec{\mathbf{s^t}}$ with the following support:
   For any $i \in \mathcal{N}_{VP}$, set $D_{\vec{\mathbf{s^t}}}(vp_i) := VC$ and $D_{\vec{\mathbf{s^t}}}(ep) := EC$. Then set $D_{\vec{\mathbf{s^t}}}(vp_j) := D_{\vec{\mathbf{s^t}}}(vp_i)$, $\forall\, j \neq i$, $j \in \mathcal{N}_{VP}$.
4. Determine the probabilities distributions of players in $\vec{\mathbf{s^t}}$ as follows:
   ep : $\forall\, e \in D_{\vec{\mathbf{s^t}}}(ep)$, set $P_{\vec{\mathbf{s^t}}}(ep, e) := 1/|EC|$. Also, $\forall\ e' \in E(T)$, $e' \notin D_{\vec{\mathbf{s^t}}}(ep)$, set $P_{\vec{\mathbf{s^t}}}(ep, e') := 0$.

   For any vp_i, $i \in \mathcal{N}_{VP}$: $\forall\, v \in D_{\vec{\mathbf{s^t}}}(vp_i)$, set $P_{\vec{\mathbf{s^t}}}(vp_i, v) := \frac{1}{|VC|}$. Also, $\forall\, u \notin D_{\vec{\mathbf{s^t}}}(vp_i)$, set $P_{\vec{\mathbf{s^t}}}(vp_i, u) := 0$. Then set $\vec{\mathbf{s^t}}_j = \vec{\mathbf{s^t}}_i$, $\forall\, j \neq i$, $j \in \mathcal{N}_{VP}$.

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

4.1.3 Trees

In Figure 1 we present in pseudocode an algorithm for computing mixed NE for trees graph instances. In the analysis following it is proved to be of polynomial time.

Claim 4.10 *Set VC, computed by Algorithm* $\mathsf{Trees}(\Pi_{\mathsf{E}}(T))$, *is an independent set of T.*

Proof. We prove that at any time in the execution of the algorithm, VC is an independent set of G. We prove this by induction. In the first iteration of step **2** of the algorithm the statement is true since $leaves(T_r) = leaves(T)$ and these vertices are the first to be added to VC. Assume by induction, that until the rth iteration of step **2**, the statement is true. We prove that after the r-th iteration VC remains an independent set of G. During iteration r, VC is expanded by the leaves of tree T_r, $leaves(T_r)$. We consider the neighbours of any $v \in leaves(T_r)$ in T and show that none of them belongs to VC, as required. The neighbours of v in T are either (i) its children or (ii) its parent. For case (i), consider a child of v, u. We may see that vertex u cannot belong to VC, for, if it was added to VC during the ith iteration of the algorithm, $i < r$, then the parent v of u would have been excluded from tree T_{i+1}. Since no vertices are ever added to this tree, this would imply that $v \notin T_r$ which is a contradiction. For case (ii), suppose v has a parent, u. By step **2**, we note that in every iteration only the leaves of T_r are added to VC. Vertex u is not a leaf neither in T_r nor in any T'_r, $r' \leq r$, thus it could not have been added to VC, in previous iterations of the step. ■

Claim 4.11 *Set EC is an edge cover of T.*

Proof. Note that any vertex v of V, considered in step **2** of the algorithm, is either as a leaf of some T_r or as a parent of such a leaf. In either case, an edge (v, u) is added to VE, where (i) if v

is a leaf of T_r then u is its parent or one of its children, depending on whether v has a parent, and (ii) if v is a parent of a leaf of T_r then u is that leaf. Thus, in any case, vertex v is covered by EC.

■

Claim 4.12 *Set VC is a vertex cover of the graph induced by set EC.*

Proof. By step **2** of the algorithm, we add an edge (u, v) in EC, for each vertex v added in VC, such that either $u = parent(v)$ or $u \in children_T(v)$. Thus, set VC covers all edges of EC. ■

Claim 4.13 *For all $v \in D_{\vec{\mathbf{st}}}(vp)$, $m_{\vec{\mathbf{st}}}(v) = \frac{\nu}{|D_{\vec{\mathbf{st}}}(vp)|}$. Also, for all $v' \notin D_{\vec{\mathbf{st}}}(vp)$, $m_{\vec{\mathbf{st}}}(v') = 0$.*

Proof. By step **4** of the algorithm, $\forall\, v \in D_{\vec{\mathbf{st}}}(vp)$, $m_{\vec{\mathbf{st}}}(v) = \sum_{i \in \mathcal{N}_{VP}} P_{\vec{\mathbf{st}}}(vp_i, v) = \sum_{i \in \mathcal{N}_{VP}} \frac{1}{|D_{\vec{\mathbf{st}}}(vp)|}$ $= \frac{\nu}{|D_{\vec{\mathbf{st}}}(vp)|}$. In contrast, for any other vertex v' not in $D_{\vec{\mathbf{st}}}(vp)$, by the same step, $P_{\vec{\mathbf{st}}}(vp_i, v') = 0$, $\forall\, i \in \mathcal{N}_{VP}$ and hence $m_{\vec{\mathbf{st}}}(v') = 0$. ■

Claim 4.14 *For all $v \in D_{\vec{\mathbf{st}}}(vp)$, $P_{\vec{\mathbf{s}}}(Hit(v)) = \frac{1}{|D_{\vec{\mathbf{st}}}(ep)|}$. Also, for all $v' \notin D_{\vec{\mathbf{st}}}(vp)$, $P_{\vec{\mathbf{s}}}(Hit(v')) \geq \frac{1}{|D_{\vec{\mathbf{st}}}(ep)|}$.*

Proof. We first consider any $v \in D_{\vec{\mathbf{st}}}(vp)$. If $v \in leaves(T)$, the only edge incident to v is its parent and by step **2** of the algorithm, edge $e = (v, parent(v))$ is added in EC. Thus $P_{\vec{\mathbf{st}}}(Hit(v)) = P_{\vec{\mathbf{st}}}(ep, e) = \frac{1}{|D_{\vec{\mathbf{st}}}(ep)|}$ for any such vertex. Now consider any non-leaf vertex $v \in T$, $v \in D_{\vec{\mathbf{st}}}(vp)$, and let r be the iteration in which v is added in VC. In the same iteration, an edge $e = (u, v)$, where u is either (i) a parent or (ii) a child of v in T, is added in EC. For case (i), we may see that vertex u cannot belong to VC, for, if it was added to VC during the ith iteration of the algorithm, $i < r$, then the parent v of u would have been excluded from tree T_{i+1}. Since no vertices are ever added to this tree, this would imply that $v \notin T_r$ which is a contradiction. Henceforth, any such edge (u, v) could not have been been added in EC before. For case (ii), by step **2**, we note that in every iteration only the leaves of T_r are added to VC. Vertex u is not a leaf neither in T_r nor in any T'_r, $r' \leq r$, thus it could not have been added to VC, in previous iterations of the step. Henceforth, again edge (u, v) could not have been been added in EC before. We conclude that in both cases, the only edge incident to v in EC is the one added in the set in current iteration of step **2**. Thus, $P_{\vec{\mathbf{st}}}(Hit(v)) = P_{\vec{\mathbf{st}}}(ep, e) = \frac{1}{|D_{\vec{\mathbf{st}}}(ep)|}$ for any such vertex v and its corresponding edge $e = (u, v)$ added in EC in current iteration.

For any $v' \in V$, $v' \notin D_{\vec{\mathbf{st}}}(vp)$, since $D_{\vec{\mathbf{st}}}(ep)$ is an edge cover of G (Claim 4.11), there exists $e_{v'} = (u, v') \in D_{\vec{\mathbf{st}}}(ep)$ and thus, $P_{\vec{\mathbf{st}}}(Hit(v')) = \sum_{e \in Eneigh(v')} P_{\vec{\mathbf{st}}}(ep, e) \geq P_{\vec{\mathbf{st}}}(ep, e_{v'}) = \frac{1}{|D_{\vec{\mathbf{st}}}(ep)|} = P_{\vec{\mathbf{st}}}(Hit(v))$, where $v \in D_{\vec{\mathbf{st}}}(vp)$. ■

Theorem 4.15 *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 $O(n^2)$.*

Proof. **Correctness**: We prove the computed profile $\vec{\mathbf{st}}$ satisfies all conditions of Theorem 4.6, thus it is a mixed NE. **1.**: By Claims 4.11 and 4.12. **2.**: By Claim 4.14. **3.(a)**: Note that, $D_{\vec{\mathbf{st}}}(vp)$ is an independent set of G and also a vertex cover of $D_{\vec{\mathbf{st}}}(vp)$, by Claims 4.10, 4.12, respectively. Thus, by Claim 4.13, for any $e = (u, v) \in D_{\vec{\mathbf{st}}}(ep)$, we have $m_{\vec{\mathbf{st}}}(e) = m_{\vec{\mathbf{st}}}(v) + m_{\vec{\mathbf{st}}}(u) = \frac{\nu}{|D_{\vec{\mathbf{st}}}(vp)|} + 0$.
3.(b): Since VC is an independent set of G, for any $e = (u, v) \in E$, $e \notin D_{\vec{\mathbf{st}}}(ep)$, $m_{\vec{\mathbf{st}}}(e) = m_{\vec{\mathbf{st}}}(v) + m_{\vec{\mathbf{st}}}(u) \leq \frac{\nu}{|D_{\vec{\mathbf{st}}}(vp)|} = m_{\vec{\mathbf{st}}}(e')$, where $e' \in EC$.

Time Complexity: Step **2** iterates n times and each iteration takes $O(n)$ time, thus step **2** needs $O(n^2)$ time. Step **3** needs $O(n)$ time. Step **4** can is accomplished in $O(1)$ time; we take constant time to compute $P_{\vec{\mathbf{st}}}$ for a vertex player i and the edge player. Thus, the algorithm terminates in time $O(n^2)$. ■

4.2 The Price of Anarchy

Lemma 4.16 *For any* $\Pi_{\mathsf{E}}(G)$ *and an associated mixed NE* $\vec{\mathbf{s}^*}$*, the social cost* $\mathsf{SC}(\Pi_{\mathsf{E}}(G), \vec{\mathbf{s}^*})$ *is upper and lower bounded as follows:*

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

These bounds are tight.

Proof. First recall that $\mathsf{SC}(\Pi_{\mathsf{E}}(G), \vec{\mathbf{s}^*}) = \mathsf{IC}_{ep}(\vec{\mathbf{s}^*})$, henceforth we proceed with the latter notion.

First term of the Lower bound :

$$\begin{aligned}
\sum_{e \in D_{\vec{\mathbf{s}^*}}(ep)} m_{\vec{\mathbf{s}^*}}(e) \stackrel{(*1)}{=} |D_{\vec{\mathbf{s}^*}}(ep)| \cdot \max_e m_{\vec{\mathbf{s}}}(e) &= \sum_{v \in V} (|ENeigh(v)|) \cdot m_{\vec{\mathbf{s}^*}}(v) \text{ (by Claim 4.2)} \\
&\geq \sum_{v \in V(D_{\vec{\mathbf{s}^*}}(ep))} m_{\vec{\mathbf{s}^*}}(v) = \nu \text{ (by Claim 4.5)} \Leftrightarrow \\
\max_e m_{\vec{\mathbf{s}^*}}(e) \stackrel{(*2)}{=} \mathsf{IC}_{ep}(\vec{\mathbf{s}^*}) &\geq \frac{\nu}{|D_{\vec{\mathbf{s}^*}}(ep)|}
\end{aligned}$$

Equality (*1) is obtained by Theorem 4.6, condition **3.(a)**. For equality (*2), by the previous observation, equation (2) and since $\vec{\mathbf{s}^*}$ is a mixed NE we have, $\mathsf{IC}_{ep}(\vec{\mathbf{s}^*}) = \mathsf{IC}_{ep}(\vec{\mathbf{s}^*_{-ep}}, [e]) = m_{\vec{\mathbf{s}^*}}(e) = \max_e m_{\vec{\mathbf{s}}}(e)$, for any $e \in D_{\vec{\mathbf{s}^*}}(ep)$.

Second term of the lower bound: Observe that in any mixed NE, $\vec{\mathbf{s}^*}$, there exists a $v_0 \in D_{\vec{\mathbf{s}^*}}(vp)$ such that $m_{\vec{\mathbf{s}^*}}(v) \geq \nu/|V(D_{\vec{\mathbf{s}^*}}(vp)|$. This is true because otherwise, $\sum_{v \in V(D_{\vec{\mathbf{s}^*}}(vp))} m_{\vec{\mathbf{s}^*}}(v) < |V(D_{\vec{\mathbf{s}^*}}(vp))| \cdot \nu/|V(D_{\vec{\mathbf{s}^*}}(vp))| = \nu$, a contradiction to Claim 4.5. Thus, $\mathsf{IC}_{ep}(\vec{\mathbf{s}^*}) = \max_e m_{\vec{\mathbf{s}}}(e) \geq m(v_0) = \frac{\nu}{|V(D_{\vec{\mathbf{s}^*}}(vp))|}$.

Upper bound:

$$\begin{aligned}
\sum_{e \in D_{\vec{\mathbf{s}^*}}(ep)} m_{\vec{\mathbf{s}^*}}(e) = |D_{\vec{\mathbf{s}^*}}(ep)| \cdot \max_e m_{\vec{\mathbf{s}}}(e) &= \sum_{v \in V} (|ENeigh(v)|) \cdot m_{\vec{\mathbf{s}^*}}(v) \\
&\leq \sum_{v \in V} \Delta(D_{\vec{\mathbf{s}^*}}(ep)) \cdot m_{\vec{\mathbf{s}^*}}(v) \leq \Delta(D_{\vec{\mathbf{s}^*}}(ep)) \cdot \nu \text{ (by Claim 4.5)} \Leftrightarrow \\
\max_e m_{\vec{\mathbf{s}^*}}(e) = \mathsf{IC}_{ep}(\vec{\mathbf{s}^*}) &\leq \frac{\Delta(D_{\vec{\mathbf{s}^*}}(ep)) \cdot \nu}{|D_{\vec{\mathbf{s}^*}}(ep)|}
\end{aligned}$$

Tightness of the first term of the lower bound: Consider an instance $\Pi_{\mathsf{E}}(G)$, where G has a perfect matching M. Then the mixed NE, $\vec{\mathbf{s}^{\mathbf{f}}} = \vec{\mathbf{s}^*}$, described in Theorem 4.9 has,

$$\begin{aligned}
\sum_{e \in D_{\vec{\mathbf{s}^*}}(ep)} m_{\vec{\mathbf{s}^*}}(e) &= |D_{\vec{\mathbf{s}^*}}(ep)| \cdot \max_e m_{\vec{\mathbf{s}}}(e) \\
&= \sum_{v \in V} (|ENeigh(v)|) \cdot m_{\vec{\mathbf{s}^*}}(v) = \sum_{v \in V} (1) \cdot m_{\vec{\mathbf{s}^*}}(v) = \nu \text{ (by \textit{Claim} 4.5)} \Leftrightarrow \\
\max_e m_{\vec{\mathbf{s}^*}}(e) = \mathsf{IC}_{ep}(\vec{\mathbf{s}^*}) &= \frac{\nu}{|D_{\vec{\mathbf{s}^*}}(ep)|}
\end{aligned}$$

Tightness of the upper bound: Consider an instance $\Pi_{\mathsf{E}}(G)$, where G is an r-regular graph. Then the mixed NE, $\vec{\mathbf{s}^{\mathbf{r}}} = \vec{\mathbf{s}^*}$, described in Theorem 4.7 has

$$\begin{aligned}
\sum_{e \in D_{\vec{\mathbf{s}^*}}(ep)} m_{\vec{\mathbf{s}^*}}(e) = |D_{\vec{\mathbf{s}^*}}(ep)| \cdot \max_e m_{\vec{\mathbf{s}}}(e) &= \sum_{v \in V} (|ENeigh(v)|) \cdot m_{\vec{\mathbf{s}^*}}(v) \\
&= \sum_{v \in V} \Delta_{D_{\vec{\mathbf{s}^*}}(ep)}(v) \cdot m_{\vec{\mathbf{s}^*}}(v) = r \cdot \nu \text{ (by Claim 4.5)} \Leftrightarrow \\
\max_e m_{\vec{\mathbf{s}^*}}(e) = \mathsf{IC}_{ep}(\vec{\mathbf{s}^*}) &= \frac{r \cdot \nu}{|D_{\vec{\mathbf{s}^*}}(ep)|}
\end{aligned}$$

■

Theorem 4.17 *The Price of Anarchy for the Edge model is* $\frac{n}{2} \leq \mathsf{r}(\mathsf{E}) \leq n$.

Proof. We first prove the upper bound. Observe that $\max_{\vec{\mathbf{s}}} \mathsf{SC}(\Pi_\mathsf{E}, \vec{\mathbf{s}}) = \nu$, since in the best scenario for the edge player all vertex players are caught. By this observation and Lemma 4.16, we get,

$$\begin{aligned}\mathsf{r}(\mathsf{E}) &= \frac{\nu}{\min_{\Pi_\mathsf{E},\vec{\mathbf{s}^*}} \mathsf{SC}(\Pi_\mathsf{E}, \vec{\mathbf{s}^*})} \leq \frac{\nu}{\min_{\Pi_\mathsf{E},\vec{\mathbf{s}^*}}\{\max\{\frac{\nu}{|D_{\vec{\mathbf{s}^*}}(ep)|}, \frac{\nu}{|V(D_{\vec{\mathbf{s}^*}}(vp))|}\}\}} \\ &\leq \max_{\Pi_\mathsf{E},\vec{\mathbf{s}^*}}\{\min\{|D_{\vec{\mathbf{s}^*}}(ep)|, |V(D_{\vec{\mathbf{s}^*}}(vp))|\}\} \leq \max_{\Pi_\mathsf{E},\vec{\mathbf{s}^*}}\{|V(D_{\vec{\mathbf{s}^*}}(vp))|\} \leq n\end{aligned}$$

The last inequality holds because $\max\{\min\{f_1, f_2\}\} \leq \max\{f_1\}$, where f_1, f_2 are any two functions and by recalling that $|D_{\vec{\mathbf{s}^*}}(vp)| \leq n$ for any $\vec{\mathbf{s}} \in S$.

For the lower bound, recall the mixed NE $\vec{\mathbf{s}^\mathbf{r}}$ of $\Pi^r_\mathsf{E}(G)$, where G is an r-regular graph, described in Theorem 4.7. We get

$$\mathsf{r}(\mathsf{E}) = \frac{\nu}{\min_{\Pi_\mathsf{E},\vec{\mathbf{s}^*}} \mathsf{SC}(\Pi_\mathsf{E}, \vec{\mathbf{s}^*})} \geq \frac{\nu}{\mathsf{SC}(\Pi^r_\mathsf{E}, \vec{\mathbf{s}^\mathbf{r}})} = \frac{\nu}{\mathsf{IC}_{ep}(\vec{\mathbf{s}^\mathbf{r}})} = \frac{\nu}{2\nu/n} = \frac{n}{2}$$

■

References

[1] N. Alon, R. M. Karp, D. Peleg and D. West, "A Graph-Theoretic Game and its Application to the k-Server Problem", *SIAM Journal on Computing*, Vol 24, No 1, pp. 78-100, February 1995.

[2] J. Aspnes, K. Chang and 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*, pp. 43-52, January 2005.

[3] M. Franklin, P. Alto, Z. Galil and Moti Yung, "Eavesdropping Games: a Graph-Theoretic Approach to Privacy in Distributed Systems", *Journal of the ACM*, Vol 47, No 2, pp. 225-243, March 2000.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to Theory of NP-Completeness*, W. H./ Freeman and Company, 1979.

[5] M. Kearns and L. Ortiz, "Algorithms for Interdependent Security Games", *Proceedings of the 17th Annual Conference on Neural Information Processing Systems*, December 2003.

[6] E. Koutsoupias and C. H. Papadimitriou, "Worst-Case Equilibria", *In Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pp. 404–413, Springer-Verlag, March 1999.

[7] M. Mavronicolas, V. Papadopoulou, A. Philippou and P. Spirakis, "A Network Game with Attacker and Protector Entities", *TR-05-13*, University of Cyprus, July 2005.

[8] 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.

[9] B. Manthey, "On Approximating Restricted Cycle Covers", Technical Report *arXiv:cs.CC/0504038 v2* , June 10, 2005.

[10] J. F. Nash, "Equilibrium Points in n-Person Games", *Proceedings of the National Acanemy of Sciences of the United States of America*, Vol 36, pp 48-49, 1950.

[11] J. F. Nash, "Non cooperative Games", *Annals of Mathematics*, Vol 54, No 2, pp. 286-295, 1951.

[12] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[13] C. H. Papadimitriou, "Algorithms, Games, and the Internet", *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 749-753, June 2001.

[14] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Third Edition, Prentice Hall, 2003.

[15] W. T. Tutte, "A Short Proof of the Factor Theorem for Finite Graphs", *Canadian Journal of Mathematics*, Vol 6, pp. 347-352, 1954.

[16] D. B. West, *Introduction to Graph Theory*, Prentice Hall, 2nd edition, 2001.