Game-theoretic Robustness of Many-to-one Networks

Aron Laszka[1], Dávid Szeszlér[2], and Levente Buttyán[1]

[1] Budapest University of Technology and Economics,
Department of Telecommunications,
Laboratory of Cryptography and System Security,
http://www.crysys.hu/,
{laszka,buttyan}@crysys.hu
[2] Budapest University of Technology and Economics,
Department of Computer Science and Information Theory,
http://www.cs.bme.hu/,
szeszler@cs.bme.hu

Abstract. In this paper, we study the robustness of networks that are characterized by many-to-one communications (e.g., access networks and sensor networks) in a game-theoretic model. More specifically, we model the interactions between a network operator and an adversary as a two player zero-sum game, where the network operator chooses a spanning tree in the network, the adversary chooses an edge to be removed from the network, and the adversary's payoff is proportional to the number of nodes that can no longer reach a designated node through the spanning tree. We show that the payoff in every Nash equilibrium of the game is equal to the reciprocal of the *persistence* of the network. We describe optimal adversarial and operator strategies and give efficient, polynomial-time algorithms to compute optimal strategies. We also generalize our game model to include varying node weights, as well as attacks against nodes.

Key words: game theory, adversarial games, network robustness, directed graph strength, graph persistence, access networks, sensor networks

1 Introduction

Access networks and sensor networks are inherently vulnerable to physical attacks, such as jamming and destruction of nodes and links. From a topological point of view, the common characteristic of these networks is that the primary goal of the nodes is to communicate with a designated node; therefore, we will refer to them as many-to-one networks, as opposed to many-to-many networks, such as backbone networks. For example, in a mesh network of wireless routers that provide Internet access to mobile terminals, every router is typically interested in communicating with a designated gateway router through which the Internet is reachable, and not with other peer routers of the network (except for

the purpose of packet forwarding of course). As another example, in a sensor network, the goal of the network is to collect the sensed data at a designated central node.

In this paper, we study the robustness of many-to-one networks in a game-theoretic model. Traditionally, game-theoretic analysis of many-to-one networks has been focused on resource allocation and routing in order to ensure fairness and efficiency [1, 2, 3]. To the best of our knowledge, our work is the first that uses game-theoretic analysis of network robustness in many-to-one networks.

Our work is inspired by [4] and [5], which use game-theoretic analysis of robustness of many-to-many networks. In [4], the strategic interactions between a network manager, whose goal is to keep the network connected by choosing a spanning tree, and an attacker, whose goal is to disconnect the network by attacking a link, were modeled as a zero-sum game. It was shown that the payoff in every Nash equilibrium of the game is the reciprocal of the *(undirected) strength* of the network. Furthermore, an efficient algorithm was provided to compute an optimal attack. In [5], the game model was generalized to include link attack costs, which can vary based on the targeted links, resulting in a non-zero-sum game.

While the definition of our game resembles that of [4] and [5], it is actually fundamentally different:

- First, our game models many-to-one networks, while [4] and [5] modeled many-to-many networks. We believe that studying adversarial games in many-to-one networks is more important as these networks are usually more vulnerable to attacks.
- Second, our payoff function considers the number of separated nodes, i.e., how disconnected the network becomes as the result of an attack. This is a more realistic function for both the operator and the adversary.
- Finally, besides giving an algorithm to compute an optimal adversarial strategy, we also give an algorithm to compute an optimal operator strategy.

Since we believe that a general theory of adversarial network games and graph robustness metrics is possible, we have kept our notions as similar to that of [4] and [5] as possible, even though our model and methodology is different.

In [6], a robustness metric for directed graphs with a designated node, called *directed graph strength*, was introduced and shown to be computable in polynomial time. Unfortunately, the name "directed strength" is misleading for two reasons: Firstly, the definition works for undirected graphs as well, without any modifications. Secondly, the fundamental difference between directed strength and the similarly named (undirected) strength (which is also introduced in [6] and used in [4]) is that the former is concerned with reachability between each node and a designated node, while the latter is concerned with reachability between every pair of nodes. Therefore, to avoid ambiguity, we renamed directed graph strength to *persistence* in [7]. In this paper, we continue to to use the name persistence.

The main contributions of our paper are the following:

– We model the interactions between a network operator and an adversary as a two player, zero-sum game.
– We show that the payoff in every Nash equilibrium of the game is equal to the reciprocal of the *persistence* of the network.
– We describe optimal adversarial and operator strategies and give efficient, polynomial-time algorithms to compute such optimal strategies.

The organization of this paper is the following: In Section 2, we present our game model. In Section 3, we introduce the concepts and notions used by subsequent sections. In Section 4, we propose an optimal adversarial strategy and show that the expected payoff of the adversary cannot be smaller than the reciprocal of the persistence of the network if she adopts the optimal strategy. In Section 5, we propose an optimal operator strategy and show that the expected payoff of the operator cannot be smaller than minus the reciprocal of the persistence of the network when it follows the optimal strategy. In Section 6, we combine the results of the preceding sections to describe a class of Nash equilibria of the game. In Section 7, we generalize our game model to allow nodes with non-uniform weights and attacks against nodes. Finally, in Section 8, we conclude the paper.

2 The Game

The network topology is represented by a connected undirected graph $G = (V, E)$ with a designated node $r \in V(G)$. The goal of the network operator is to keep the nodes of the network connected to the designated node, while the goal of the adversary is to separate as many nodes as possible from it.

The interaction between the network operator and the adversary is modeled as a two player, one-shot, zero-sum game. The network operator chooses a spanning tree to be used for communications. The mixed strategy of the network operator is a distribution on the set of spanning trees $\mathcal{T}(G)$, i.e., $\mathcal{A} := \{\alpha \in \mathbb{R}_{\geq 0}^{|\mathcal{T}(G)|} | \sum_{T \in \mathcal{T}(G)} \alpha_T = 1\}$. The adversary chooses an edge to be attacked. The mixed strategy of the adversary is a distribution on $E(G)$, i.e., $\mathcal{B} := \{\beta \in \mathbb{R}_{\geq 0}^{|E(G)|} | \sum_{e \in E(G)} \beta_e = 1\}$. The payoff for the adversary is the number of nodes from which there is no path to r in $T \setminus \{e\}$, where T and e are the spanning tree and edge chosen by the operator and the adversary, respectively. If $e \notin T$, then the payoff is obviously zero.

Let $\lambda(T, e)$ denote the number of nodes that are disconnected from r if the operator uses T and the adversary attacks e. Then, the payoff function of the game for the adversary can be written as

$$P(\alpha, \beta) = \sum_{e \in E(G)} \sum_{T \in \mathcal{T}(G)} \alpha_T \beta_e \lambda(T, e) \ . \tag{1}$$

The adversary has to solve $\max_{\beta} \min_{\alpha} P(\alpha, \beta)$, while the operator has to solve $\min_{\alpha} \max_{\beta} P(\alpha, \beta)$. The corresponding solutions, i.e., the optimal adversarial and operator strategies, are presented in Section 4 and Section 5, respectively.

In this paper, similarly to [4] and [5], we restrict the pure strategies of the adversary to attacking single edges only. Studying generalized game models, in which the pure strategies of the adversary consist of subsets of edges, is an open problem in case of both many-to-many and many-to-one networks.

3 Preliminaries

In this section, we introduce the basic concepts and notions used by subsequent sections.

For a set of edges $A \subseteq E(G)$, let $\lambda(A)$ denote the number of nodes from which there is no path leading to r in the graph when A is removed.

In [6], the persistence of a graph was defined as:

Definition 1 (Persistence). *Given a directed graph G with a designated node $r \in V(G)$, the* persistence $\pi(G)$ *is defined as*

$$\pi(G) = \min \left\{ \frac{|A|}{\lambda(A)} : A \subseteq E(G), \lambda(A) > 0 \right\} . \tag{2}$$

Since reachability is well-defined in case of undirected graphs as well, the above definition also works for undirected graphs without any modifications.

Definition 2 (Critical set). *A set of edges $A \subseteq E(G)$ is* critical*, if $\frac{|A|}{\lambda(A)} = \pi(G)$, i.e., if the minimum in Definition 1 is attained.*

Definition 3 (Expected loss). *The* expected loss *of an edge $e \in E(G)$ in a given operator strategy α is the expected payoff of the pure adversarial strategy targeting exclusively e, i.e., $\sum_{T \in \mathcal{T}} \alpha_T \cdot \lambda(T, e)$.*

3.1 Computing persistence

It is shown in [6] that the computation of persistence can be performed using a maximum flow algorithm [1].

Assume that the task is to decide if $\pi(G) \geq \pi_0$ holds, where π_0 is a given constant. For any set $X \subseteq V(G)$, denote by $\delta(X)$ the set of edges leaving X. It is easy to see that the minimum in Definition 1 is attained at a set $A = \delta(X)$ for a suitable $X \subseteq V(G) \setminus \{r\}$. (Indeed, "spare" edges could be deleted from A without increasing the ratio $|A|/\lambda(A)$.) Of course, $A = \delta(X)$ implies $\lambda(A) = |X|$. Therefore, $\pi(G) \geq \pi_0$ is equivalent to saying that $|\delta(X)| - \pi_0 \cdot |X| \geq 0$ holds for

[1] In this subsection we build on the basics of network flow theory; the required background can be found in most introductory graph theory textbooks.

all $X \subseteq V(G) \setminus \{r\}$. Adding $\pi_0 \cdot (|V(G)| - 1)$ to both sides we get that $\pi(G) \geq \pi_0$ is equivalent to

$$|\delta(X)| + \pi_0 \cdot (|\overline{X}| - 1) \geq \pi_0 \cdot (|V(G)| - 1) \ , \tag{3}$$

for all $X \subseteq V(G) \setminus \{r\}$ (where $\overline{X} = V(G) \setminus X$).

Consider the following maximum network flow problem. Add a new node s to G; for each $v \in V(G) \setminus \{r\}$ add a new arc from s to v and set its capacity to π_0; finally, set the capacity of each original arc of G to 1. Denote the obtained network by G^*. According to the well-known "max-flow-min-cut" theorem of Ford and Fulkerson, the maximum flow in the obtained network from s to r is equal to the minimum cut capacity, that is, the minimum of the sum of capacities on arcs leaving a set X, where minimum is taken over all subsets $X \subseteq V(G^*)$ for which $s \in X$ and $r \notin X$. Obviously, the capacity of the cut X is $|\delta(X)| + \pi_0 \cdot (|\overline{X}| - 1)$. Comparing this with Equation 3 above, we get that $\pi(G) \geq \pi_0$ is equivalent to the existence of a flow of value $\pi_0 \cdot (|V(G)| - 1)$ from s to r in the above constructed network; or, in other words, a flow that satures all arcs leaving s.

Consequently, the question of $\pi(G) \geq \pi_0$ can be answered by a maximum flow algorithm. From this, the actual value of $\pi(G)$ (that is, the maximum π_0 for which the above described flow exists) can be determined by binary search (which yields a polynomial time algorithm if all input numerical data is assumed to be integer). In [6] a refinement of this approach is also given: it is shown that $\pi(G)$ can be determined by at most $|V(G)|$ maximum flow computations (even for arbitrary input data).

Furthermore, if $\pi(G)$ is known, the above described reduction to maximum flow can be also used to find a critical set: Construct a G^* in the above manner with $\pi_0 = \pi(G)$. A minimum cut in G^* is a critical set in G.

4 Adversary strategy

In this section, we describe an adversarial strategy, which achieves an expected payoff of $\frac{1}{\pi(G)}$, regardless of the strategy of the operator. Later, in Section 5, we show that this strategy is optimal by proving that this is the highest attainable expected payoff for the attacker if the operator is rational.

Theorem 1. *If an adversary targets exclusively the edges of a critical set A with uniform probability, then her expected payoff is at least $\frac{1}{\pi(G)}$.*

Proof. For any given spanning tree $T \in \mathcal{T}$ and set of edges $B \subseteq E(G)$, $\sum_{e \in B} \lambda(T, e) \geq \lambda(B)$, since every node cut off by removing B has to increase $\lambda(T, e)$ by one for at least one $e \in B$. Therefore, the expected payoff for the adversary is

$$\begin{aligned}\sum_{e\in E(G)}\sum_{T\in\mathcal{T}}\alpha_T\beta_e\lambda(T,e) &= \sum_{e\in A}\sum_{T\in\mathcal{T}}\alpha_T\frac{1}{|A|}\lambda(T,e)\\ &= \frac{1}{|A|}\sum_{T\in\mathcal{T}}\alpha_T\sum_{e\in A}\lambda(T,e)\\ &\geq \frac{1}{|A|}\sum_{T\in\mathcal{T}}\alpha_T\lambda(A)\\ &= \frac{\lambda(A)}{|A|}\sum_{T\in\mathcal{T}}\alpha_T\\ &= \frac{\lambda(A)}{|A|} = \frac{1}{\pi(G)}\ .\end{aligned}$$

□

As seen before in Subsection 3.1, a critical set can be computed in polynomial time, which implies that the same holds for the adversary strategy described in Theorem 1.

5 Operator strategy

In this section, we propose an efficient algorithm that computes an optimal operator strategy, which achieves $\frac{1}{\pi(G)}$ expected payoff, regardless of the strategy of the operator. We have already shown in Section 4 that this is the best attainable expected payoff for the operator if the adversary is rational.

The following lemma is required by the proof of our main theorem:

Lemma 1. *Let G be a graph with a designated sink node r. Let G' denote the graph obtained from G in the following way: Add a source node s to the graph. For each $v \in V(G)\setminus\{r\}$, add an arc from s to v and set its capacity to 1. Finally, set the capacity of every original edge of the graph to $\frac{1}{\pi(G)}$. The maximum flow in G' from s to r is $|V(G)|-1$.*

Proof. This readily follows from Subsection 3.1 by scaling the capacity of each edge with $\frac{1}{\pi(G)}$.

The proof of the following theorem is a constructive proof, where we describe our efficient algorithm for obtaining optimal operator strategies:

Theorem 2. *Let G be a graph with a designated node r. There is an operator strategy in which the expected loss of every edge is at most $\frac{1}{\pi(G)}$.*

Proof. Our proof is constructive and it is based on the following algorithm:

1. Let G' be the graph obtained from G in the way described in Lemma 1 with the designated node used as the sink node. Find a maximum flow f in G' from the source node s to the designated node r.

2. Find a spanning reverse arborescence [2] T rooted at r in G such that
   – T only includes edges to which f assigns a positive flow amount and
   – every edge is directed in the same way as the flow.
3. Calculate $\lambda(T, e)$ for every $e \in T$.
4. Let $\alpha_T := \min_{e \in T} \frac{f(e)}{\lambda(T,e)}$.
5. For every $e \in E(G)$, let $f(e) := f(e) - \alpha_T \cdot \lambda(T, e)$.
6. For every $v \in V(G) \setminus \{r\}$, let $f((s, v)) := f((s, v)) - \alpha_T$.
7. If the flow assigned by f from s to r is greater than zero, then continue from Step 2.
8. Let $\alpha_T := 0$ for every other spanning tree.

Before proving the correctness of the algorithm, we have to prove that Step 2 can be executed in each iteration, otherwise the algorithm would terminate incorrectly. Obviously, if f is a network flow and the amount of flow along every $(s, v), v \in V(G) \setminus \{r\}$ edge is positive, there has to be a directed path from every $v \in V(G) \setminus \{r\}$ to r consisting of edges with positive flow amounts. Thus, we have to show that if f is a network flow carrying γ from s to r before Step 5, then it is a network flow carrying $\gamma - \alpha_T(|V(G)| - 1)$ from s to r after Step 6.

For a $v \in V(G) \setminus \{r\}$, let λ_v denote $\lambda(T, e_{out})$, where e_{out} is the outgoing edge of v in T. Clearly, the sum of $\lambda(T, e_{in})$ over all incoming edges $e_{in} \in E(G)$ of v is $\lambda_v - 1$. Since the flow along every edge e is decreased by $\alpha_T \cdot \lambda(T, e)$, the sum of outgoing flows is decreased by $\alpha_T \cdot \lambda_v$. Similarly, the sum of incoming flows is decreased by $\alpha_T \cdot (\lambda_v - 1) + \alpha_T = \alpha_T \cdot \lambda_v$, which takes the α_T decrease on (s, v) into account as well. Clearly, the net flow at v remains zero. Since this is true for every node, except s and r, f remains a network flow. The flow from s to r is decreased by $\alpha_T(|V(G)| - 1)$, since the flow on every (s, v), $v \in V(G) \setminus \{r\}$, edge is decreased by α_T.

Now, we can prove the correctness of the algorithm. First, we have to prove that α is indeed a distribution, i.e., $\sum_{T \in \mathcal{T}} \alpha_T = 1$ and $\alpha_T \geq 0, \forall T \in \mathcal{T}$. This is evident, as the amount of flow from s to r is decreased by $\alpha_T(|V(G)| - 1)$ at every assignment, and the amount is $|V(G)| - 1$ after Step 1 and zero after the algorithm has finished.

Second, we have to prove that the expected loss of every edge in $E(G)$ is at most $\frac{1}{\pi(G)}$. After Step 1, the amount of flow along every edge is at most $\frac{1}{\pi(G)}$. At every α_T assignment, the flow along every edge is decreased by $\alpha_T \cdot \lambda(T, e)$ and it is never decreased to a negative value. Therefore $\sum_{T \in \mathcal{T}} \alpha_T \cdot \lambda(T, e) \leq \frac{1}{\pi(G)}$.

Finally, we have to prove that the algorithm terminates after a finite number of iterations. In every iteration, the flow along at least on edge (i.e., along every edge for which $\frac{f(e)}{\lambda(T,e)}$ is minimal) is decreased from a positive amount to zero. Since there are a finite number of edges, the algorithm terminates after a finite number of iterations. □

Theorem 3. *The above algorithm runs in polynomial time.*

[2] A directed, rooted spanning tree in which all edges point to the root.

Proof. In Step 8, the assignment does not have to be actually performed for every spanning tree, since it is enough to output the probabilities of only the trees in the support of the distribution. Therefore, every step of the algorithm can be performed in polynomial time. Furthermore, the number of iterations is less than or equal to the number of edges $|E(G)|$, since the flow along at least one edge is decreased from a positive amount to zero in every iteration. □

Corollary 1. *An operator strategy that achieves at least* $-\frac{1}{\pi(G)}$ *expected payoff for the operator can be found in polynomial time.*

Proof. The claim of this corollary follows from Theorem 2 and 3. Suppose that the strategy of the operator is constructed using the proposed algorithm. Then, the expected payoff of every pure adversarial strategy is at most $\frac{1}{\pi(G)}$, since $\forall e \in E(G) : \sum_{T \in \mathcal{T}} \alpha_T \cdot \lambda(T, e) \leq \frac{1}{\pi(G)}$. Therefore, the expected payoff of every mixed adversarial strategy is at most $\frac{1}{\pi(G)}$ as well. □

6 Nash-equilibrium

Based on the above results, we can describe a class of Nash equilibria:

Corollary 2. *The adversarial strategies presented in Section 4 and the operator strategies presented in Section 5 form Nash equilibria of the game. The expected payoffs for the adversary and the operator are* $\frac{1}{\pi(G)}$ *and* $-\frac{1}{\pi(G)}$*, respectively.*

Since the game is zero-sum, all Nash equilibria have the same expected payoff. Consequently, graph persistence is a sensible measure of network robustness.

7 Generalizations

In this section, we present various generalizations to our basic game model introduced in Section 2, which make our model more realistic and practical. We show that all of these generalized models can be traced back to the basic game model, i.e., with minor modifications, the previously presented theorems and algorithms apply to these generalized models as well.

7.1 Directed graphs

Recall that, in Section 3, graph persistence was defined for directed graphs, even though it was applied only to undirected graphs so far. We have restricted the topologies of the studied networks to undirected graphs only to simplify our basic model. Now, we relax this restriction, and use directed graphs to represent network topologies. This is clearly a generalization, since undirected networks can be also represented in this model by replacing each undirected edge with two arcs facing opposite directions. The generalization is very straightforward, since all steps and arguments of the previously presented algorithms and proofs work with directed graphs as well, without any modifications.

7.2 Non-uniform node weights

It is possible to generalize our results to the case where nodes have non-uniform weight or importance. Let d_v be the weight of node v: by disconnecting each node v from r, the adversary gains and the operator loses d_v (instead of 1, as in the original model). Let $\lambda(T, e)$ denote the total weight of the nodes that are disconnected from r when the operator uses T and the adversary attacks e. Similarly, let $\lambda(A)$ denote the total weight of the nodes that are disconnected when A is removed. It is easy to see that the definition of graph persistence and the proposed adversarial strategy do not have to be modified to accommodate the changes in the definitions of $\lambda(T, e)$ and $\lambda(A)$.

In case of the operator strategy, the following modifications have to be made to the proposed algorithm and the proof:

- In Step 1, the capacity of each $(s, v),\ v \in V(G) \setminus \{r\}$ arc has to be d_v, instead of 1.
- In Step 6, the capacity of each $(s, v),\ v \in V(G) \setminus \{r\}$ arc has to be decreased by $d_v \cdot \alpha_T$, instead of α_T.
- Consequently,
  - the sum of $\lambda(T, e_{in})$ over all incoming edges $e_{in} \in E(G)$ of v is $\lambda_v - d_v$, instead of $\lambda_v - 1$,
  - the flow from s to r is decreased by $\alpha_T \sum_{v \in V(G) \setminus \{r\}} d_v$, instead of $\alpha_T(|V(G)| - 1)$.

7.3 Node attacks

Based on the generalization presented in the previous subsection, our results can be further generalized to the case where the adversary is not only able to target edges, but it is able to target nodes as well. In this case, the mixed strategy of the adversary is a distribution on $(V(G) \cup E(G))$, i.e., $\mathcal{B} := \{\beta \in \mathbb{R}_{\geq 0}^{|V(G)|+|E(G)|} | \sum_{e \in (V(G) \cup E(G))} \beta_e = 1\}$.

For an arbitrary subset $A \subseteq (V(G) \cup E(G))$, let $\lambda(A)$ denote total weight of the nodes which are either elements of A or from which there is no path leading to r in the graph when A is removed.

The definition of persistence has to be generalized to allow targeting nodes:

Definition 4 (Edge-node-persistence). *Given a directed graph G with a designated node $r \in V(G)$, the* edge-node-persistence $\pi_n(G)$ *is defined as*

$$\pi_n(G) = \min \left\{ \frac{|A|}{\lambda(A)} : A \subseteq (V(G) \cup E(G)), \lambda(A) > 0 \right\} . \tag{4}$$

In [7], we have shown that computing edge-node-persistence can easily be reduced to computing persistence by *vertex splitting*, a well-known trick in graph theory: replace each node v by two nodes v_1 and v_2, add the arc (v_1, v_2) to G, let $d(v_1) = d(v)$, $d(v_2) = 0$; finally, replace each original arc (u, v) by (u_2, v_1). It

is fairly easy to see that the persistence of the obtained graph is the same as the edge-vertex-persistence of the original one.

This trick can be also used to obtain adversarial and operator strategies that achieve $\frac{1}{\pi_n(G)}$ payoff in the generalized model on any given graph G. Let G' be the graph obtained from G in the above manner. Find an optimal adversarial strategy on G' as it has been described in Section 4, which achieves $\frac{1}{\pi(G')} = \frac{1}{\pi_n(G)}$ payoff on G'. The support of the resulting distribution consists of edges in $E(G)$ and edges corresponding to nodes in $V(G)$. It is easy to see that if we replace edges corresponding to nodes with the nodes in the support of the distribution, the resulting strategy achieves $\frac{1}{\pi_n(G)}$ payoff on G. An optimal operator strategy, which achieves $\frac{1}{\pi_n(G)}$ payoff on G, can be obtained in a similar manner.

Please note that we could define a model in which an adversary is only able to target nodes, but this is unnecessary. For every optimal adversarial strategy targeting both nodes and edges, we can construct a corresponding optimal adversarial strategy that targets only nodes: simply replace each arc in the strategy with its source node. It is easy to see, that the payoff of the resulting strategy is at least as large as the payoff of the original strategy.

8 Conclusions

In this paper, we introduced a game-theoretic model of the interactions between the operator of a many-to-one network and an adversary. We showed that the payoff in every Nash equilibrium of the game is equal to the reciprocal of the persistence of the network. One of our main contributions is to link the graph-theoretic robustness of a network, measured in persistence, to game theory, which gives a better understanding of robustness and an argument for the soundness of the notion of graph persistence. We also gave efficient, polynomial-time algorithms to compute optimal strategies for the adversary and the operator. The optimal operator strategy gives a baseline for the design of robust many-to-one routing algorithms.

Acknowledgements

This paper has been supported by HSN Lab, Budapest University of Technology and Economics, `http://www.hsnlab.hu`. Dávid Szeszlér is supported by grant Nr. OTKA 103985 of the Hungarian National Science Fund. The work is also related to the internal project of the authors' hosting institution on "Talent care and cultivation in the scientific workshops of BME", which is supported by the grant TÁMOP - 4.2.2.B-10/1–2010-0009.

References

1. E. Altman, T. Boulogne, R. El-Azouzi, T. Jimenez, and L. Wynter. A survey on networking games in telecommunications. *Computers & Operations Research*, 33(2):286–311, 2006.
2. M. Felegyhazi and J.P. Hubaux. Game theory in wireless networks: A tutorial. Technical Report LCA-REPORT-2006-002, EPFL, Lausanne, Switzerland, June 2007.
3. D.E. Charilas and A.D. Panagopoulos. A survey on game theory applications in wireless networks. *Computer Networks*, 54(18):3421–3430, 2010.
4. Assane Gueye, Jean C. Walrand, and Venkat Anantharam. Design of network topology in an adversarial environment. In *Proc. of the 1st International Conference on Decision and Game Theory for Security*, GameSec'10, pages 1–20, Berlin, Germany, November 2010.
5. Assane Gueye, Jean C. Walrand, and Venkat Anantharam. A network topology design game: How to choose communication links in an adversarial environment? In *Proc. of the 2nd International ICST Conference on Game Theory for Networks*, GameNets'11, Shanghai, China, April 2011.
6. William H. Cunningham. Optimal attack and reinforcement of a network. *Journal of the ACM*, 32(3):549–561, 1985.
7. A. Laszka, L. Buttyán, and D. Szeszlér. Optimal selection of sink nodes in wireless sensor networks in adversarial environments. In *Proc. of the 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia*, WoWMoM'11, pages 1–6, Lucca, Italy, June 2011.