Quantifying Network Topology Robustness Under Budget Constraints: General Model and Computational Complexity

Aron Laszka[1] and Assane Gueye[2]

[1] Laboratory of Cryptography and System Security (CrySyS Lab),
Budapest University of Technology and Economics
laszka@crysys.hu
[2] National Institute of Standards and Technology (NIST),
assane.gueye@nist.gov

Abstract. Recently, network blocking game (NBG) models have been introduced and utilized to quantify the vulnerability of network topologies in adversarial environments. In NBG models, the payoff matrix of the game is only "*implicitly*" given. As a consequence, computing a Nash equilibrium in these games is expected to be harder than in more conventional models, where the payoff matrix is "*explicitly*" given.
In this paper, we first show that computing a Nash equilibrium of a NBG is in general NP-hard. Surprisingly, however, there are particular interesting cases for which the game can be solved in polynomial time. We revisit these cases in a framework where the network is to be operated under budget constraints, which previous models did not consider. We generalize previous blocking games by introducing a budget limit on the operator and consider two constraint formulations: the maximum and the expected cost constraints.
For practical applications, the greatest challenge posed by blocking games is their computational complexity. Therefore, we show that the maximum cost constraint leads to NP-hard problems, even for games that were shown to be efficiently solvable in the unconstrained case. On the other hand, we show that the expected cost constraint formulation leads to games that can be solved efficiently.

Keywords: network topology robustness; robustness metrics; game theory; blocking games; computational complexity.

1 Introduction

Designing network topologies that are robust and resilient to attacks has been and continues to be an important and challenging topic in the area of communication networks. One of the main difficulties resides in quantifying the robustness of a network in the presence of an intelligent attacker, who might exploit the structure of the network topology to design harmful attacks. Quantifying the robustness or, equivalently, the vulnerability of topologies has been extensively

studied [1–5]; however, the simultaneous and strategic decision making of the defender and the adversary, which is key to the security of information systems, has received only little attention.

To capture the strategic nature of the interactions between a defender and an adversary, game-theoretic models have been gaining a lot of interest in the study of the security of communication networks. In a recent line of research [6–10], *network blocking games* (NBGs) have been introduced and applied to the analysis of the robustness of network topologies. An NBG takes as input the communication model and the topology of a network, and casts the strategic interactions between an adversary and the defender, called the network operator, as a two-player game. The Nash equilibrium strategies are then used to predict the attacker's most likely actions; and the attacker's equilibrium payoff[3] serves as a quantification of the vulnerability (i.e., inverse robustness) of the network.

A particularity of NBG models is that the payoff matrix of the game is not given as an input. In other words, the strategy set of (at least) one player (and hence the payoff matrix) is only *implicitly* defined, and the actual strategy sets need to be computed from the input of the game (here, the communication model and the network topology). Furthermore, in most NBG models, checking whether a given action is a feasible strategy can be done efficiently; however, computing the complete strategy set is inherently difficult. For instance, in the game described in [6], the operator's strategy space is the set of feasible network flows. In general, checking whether a given flow is feasible can be done efficiently. However, computing the set of all feasible network flows (which is required for computing the payoff matrix) is difficult: the number of feasible flows is exponential in the number of nodes and links in the graph, so they cannot be enumerated in polynomial time.

Hence, with respect to the complexity of computing a Nash equilibrium, NBG models present two challenges: first, the game is only implicitly defined; second, the payoff matrix is potentially exponential in size. Thus, solving network blocking games can be expected to be harder than solving games for which the payoff matrix is "explicitly given". Recall that computing a NE for "explicit" two-player games has been shown to be PPAD-complete (*Polynomial Parity Arguments on Directed graphs*), a class of problems that are believed to be hard, but not necessarily NP-hard [11]. In this paper, we show that computing a Nash equilibrium of a network blocking game is NP-hard in general.

Interestingly though, in the series of NBG papers cited above, new algorithms have been developed to *efficiently* compute a Nash equilibrium in a number of communication models: All-to-All (e.g., Ethernet) networks with constant [7] and linear loss [9], All-to-One (e.g., access and sensor) networks [10], and Supply-Demand networks [6]. These algorithms are mostly based on the theory of network flows and, for some models, on the minimization of submodular functions. More precisely, the problem of finding a Nash equilibrium is cast as a network

[3] It has been shown that the attacker's payoff is the same in every equilibrium of a network blocking game; thus, it suffices to find a single equilibrium in order to characterize the robustness of a network.

flow problem (or a submodular function minimization problem), which enables bypassing the computation of the payoff matrix. In this paper, we revisit some of these models and discuss the complexity of computing their NE in scenarios where the network operator has access to only a limited budget to operate the network.

Such budget constraints were not considered in previous NBG models, which implicitly assume that the operator can use the network elements at zero cost. However, this assumption is not realistic: indeed, links in a network have positive usage costs (e.g., operation/maintenance costs, protection costs) and these costs might be non-uniform. Since network operators do not have an unlimited budget, they cannot use any combination of network element. In [6], a usage cost model as well as a budget constraint have been introduced for the particular case of Supply-Demand (S-D) networks. This budget constraint means that the network operator can use a set of network elements (links) only if its associated cost does not exceed a given budget.

In the present paper, we extend the budget constraint idea to network blocking games in general, and provide a number of complexity results with regard to the computation of the equilibrium payoff. Recall that the aim of solving these models is to derive a quantification of the network's robustness in the presence of a strategic adversary, and that the equilibrium payoff is used as the vulnerability metric. Thus, computational complexity is of central importance in these models, and analyzing it is the primary goal of this paper.

This paper builds upon the studies in [6] and [12], but considers a more general setting and presents many additional results compared to those papers. [6] is the first study to introduce the idea of a budget limit and usage costs in the context of a NBG. However, it considers only the special case of Supply-Demand networks and (what we call here) the maximum cost constraint. Furthermore, it does not provide a complexity analysis. [12] presents a complexity analysis and introduces a new constraint formulation (the expected cost constraint), but limits the discussion to the special case of the All-to-One communication model with zero attack costs. In the present paper, we consider a unifying framework and provide a thorough complexity analysis for NBGs in general. The main contributions of this paper are the following:

- In Section 3, we show that solving a blocking game is generally NP-hard (Theorem 1).
- In Section 4, we generalize the network blocking game model by introducing a budget limit for the operator. We consider two constraint formulations: the maximum cost constraint (MCC) and the expected cost constraint (ECC).
- In Section 5, we show that the problem of determining the equilibrium payoff is NP-hard under the MCC in the previously proposed models, which can be solved efficiently in the unconstrained game (Theorem 2).
- In Section 6, we show how to solve the game under the ECC in polynomial time given a linear characterization of the operator's mixed strategy space (Theorem 3).

Notational conventions We use lower case bold letters (e.g., $\boldsymbol{\alpha}$) and upper case bold letters (e.g., $\boldsymbol{S}$) to denote column vectors and matrices, respectively. We use the prime sign ($'$) to denote transpose, and subindices (e.g., $\boldsymbol{\alpha}_T$) to refer to elements of vectors.

2 Unconstrained Network Blocking Games

In this section, we summarize the previous work on network blocking games. Since these models do not consider a budget constraint, we will refer to them as *unconstrained network blocking games* when the distinction is important.

As it was stated earlier, network blocking games are defined by the communication model and the topology of the network. The topology of the network is represented by a connected simple graph $G = (V, E)$, where V is the set of nodes and E is the set of links. The edges can be undirected or directed depending on the communication models (as we will see later). The network operator wants to guarantee *"some" connectivity* between the nodes of the network. For this, she selects a collection $T \subseteq E$ of the links as the communication infrastructure. The type of *connectivity* and the set of feasible collections (denoted by $\mathcal{T}$) are determined by the communication model (see the next subsection for examples of communication models).

Assume that the operator chooses collection T for her communication and that a given link e in the network fails. In this paper, we only consider failures that are due to the actions of a malicious and strategic adversary. If $e \notin T$, then the communication is not affected at all. If, on the other hand, $e \in T$, then e can longer be used: the operator incurs some *usage loss*, which is how much she would transmit on the link if it were intact. For a given T and e, we let $\boldsymbol{\lambda}(T, e)$ denote this usage loss (or zero if $e \notin T$). Notice that all results presented in this paper also hold if the attacker is allowed to attack nodes as well[4], but we restrict our analysis to link attacks only due to the lack of space.

2.1 Communication Models

The communication model defines the type of "connectivity" that the network operator is trying to achieve, the set of feasible collections which she can use for that, and the usage losses $\boldsymbol{\lambda}(T, e)$ for the network elements. Next, we introduce the three communication models that are of interest in this paper.

All-to-One Model In an *All-to-One* network [10], the primary goal of the network operator is to enable all nodes to communicate with a designated node r. This models sensor and access networks, where all nodes are trying to reach a gateway or data collection node (or, alternatively, a set of nodes, which can be modeled by a designated super-node).

[4] The results for both node and edge attacks can be derived using vertex splitting.

To get all nodes connected to r, the network operator chooses a collection of links T that forms a spanning tree. Hence, the set of feasible collections $\mathcal{T}$ is the set of all spanning trees. In practice, a spanning tree can be implemented, for example, as the next-hop forwarding table entries for r, which are stored at the individual nodes of the network.

Let the network be connected using a spanning tree T. Then, if a given link $e \in E$ fails, some nodes might no longer be able to communicate with r and can be considered lost for the network operator. Thus, we define the usage (loss) $\boldsymbol{\lambda}(T, e)$ as *the number of those nodes that are disconnected from* r.

All-to-All Model In an *All-to-All* network [7, 9], the goal of the network operator is to enable each node to communicate with every other node, using the minimum number of links. For example, this is the case for bridged Ethernet LANs, where every node should be able to "logically" communicate with every other node, but the topology has to be loop-free. Assuming that links are undirected, spanning trees are the subgraph structures that (looplessly) connect all nodes with the minimum number of links. Hence, the network operator selects a spanning tree as communication infrastructure. Thus, the set of feasible collections $\mathcal{T}$ corresponds to the set of all spanning trees.

Let the network be connected using a spanning tree T and assume that link e fails. If link e does not belong to T, then the network remains connected and the operator does not lose any connectivity. If, on the other hand, $e \in T$, the network is cut into two separate components that are unable to communicate. Now, if e is a link connecting a leaf to the rest of the spanning tree, only that leaf gets disconnected and all the other nodes can still reach each other. In this case, the operator loses some connectivity, but the loss can be considered *minor*. If, on the other hand, the removal of link e cuts the network into two components of comparable size, then connections between many pairs of nodes are now missing, and the loss to the operator is considerably *larger*. In general, the more fractured the network is, the more severe the loss is. To capture this phenomenon, the usage (loss) $\boldsymbol{\lambda}(T, e)$ is defined as *the size of the smaller connected component of* $G(V, T \setminus e)$, where $G(V, T \setminus e)$ is the subgraph containing only the links in $T \setminus e$.

Supply-Demand Model In a Supply-Demand (S-D) network [6], the operator wants to carry a fixed amount of goods from a nonempty set $S \subseteq \mathcal{V}$ of "source" nodes to a nonempty set $D \subseteq \mathcal{V}$ of "destination" nodes using the network links. We assume that $S \cap D = \emptyset$ and that network links are directed. With each node $u \in S$, we associate a nonnegative number $s(u)$, the "supply" at u, and with each node $u \in D$, we associate a nonnegative number $d(u)$, the "demand" at u. We consider *uncapacitated* networks, where each link can carry an unlimited amount of goods[5]. We also assume that links carry only *integer* amounts of goods and that the total amount of goods to be carried from S to D is also a given positive integer.

[5] The analysis of *capacitated* network follows from the study in this paper, but it is not considered in this paper due to space limitation.

To transport the goods, the network operator chooses a collection of links that forms a *feasible (integer) flow*. A feasible flow $T \in \mathcal{T}$ is a function that assigns to each link e the amount of goods $T(e)$ (≥ 0) it carries, such that the *conservation of flow* property is satisfied at each node. Hence, the set of collections $\mathcal{T}$ is equal to the set of all feasible flows.

The usage (loss) $\boldsymbol{\lambda}(T, e)$ is defined to be the *amount of goods $T(e)$ that flow T assigns to link e*. This is how much the operator will lose if she uses a feasible flow $T \in \mathcal{T}$ and link e fails.

2.2 Game-Theoretic Measure of Robustness

Given the communication model and the topology of the network, a two-player game is defined between the network operator and a strategic attacker. The network operator wants to guarantee "some" connectivity by choosing a *feasible* collection of links in the network (i.e., her strategy space is the set $\mathcal{T}$ of feasible collections). The type of connectivity and the set of feasible collections are defined by the communication model, as previously discussed. At the same time, a strategic and malicious adversary is trying to disrupt the communication by attacking a link (i.e., her strategy space is the set E of links in the network). We assume that to successfully attack a link e, the adversary has to spend some effort which is quantified by $\boldsymbol{\mu}_e$. The players' payoffs are defined as follows: when the operator picks collection T and the attacker targets link e, the operator loses $\boldsymbol{\lambda}(T, e)$ (as defined above), and the attacker gets a net reward of $\boldsymbol{\lambda}(T, e) - \boldsymbol{\mu}_e$. The attacker also has the option not to launch an attack, which results in zero loss for the operator and zero gain for the attacker.

We consider mixed strategy Nash equilibria, where the network operator chooses a distribution (denoted by $\boldsymbol{\alpha}$) over the set $\mathcal{T}$, and the attacker chooses a distribution (denoted by $\boldsymbol{\beta}$) over the set E or the option of not attacking. We assume that the operator tries to minimize her *expected loss*, while the attacker tries to maximize her *expected net reward*. Formally, the operator chooses $\boldsymbol{\alpha}$ to minimize $L(\boldsymbol{\alpha}, \boldsymbol{\beta})$ defined as

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{T \in \mathcal{T}} \sum_{e \in E} \boldsymbol{\alpha}_T \boldsymbol{\beta}_e \boldsymbol{\lambda}(T, e) \ , \tag{1}$$

while the attacker chooses $\boldsymbol{\beta}$ to maximize $R(\boldsymbol{\alpha}, \boldsymbol{\beta})$ defined as

$$R(\boldsymbol{\alpha}, \boldsymbol{\beta}) = L(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \sum_{e \in E} \boldsymbol{\beta}_e \boldsymbol{\mu}_e \tag{2}$$

or not attacking if the maximum is negative.

Since the attacker has the option not to attack and get a payoff of zero, it is not hard to show that there does not exist an equilibrium in which the attacker receives a negative expected payoff. We let θ^* be the attacker's equilibrium payoff, which has been shown [13] to be the same in all equilibria. As a consequence, θ^* is uniquely defined. The next subsection gives a characterization of θ^* using the theory of blocking pairs of polyhedra.

2.3 Equilibrium Characterization Based on Blocking Pairs of Polyhedra

Here, we recall the notions of polyhedra and blockers, and discuss how they can be used to characterize the Nash equilibria of the game (see [13, Chap. 4] for more details).

Let $\boldsymbol{\Lambda}$ be the operator's payoff matrix, whose rows are $(\boldsymbol{\lambda}_T,\ T \in \mathcal{T})$, where the entries of the vector $\boldsymbol{\lambda}_T \in \mathbb{R}^{|E|}_{\geq 0}$ are given by $\boldsymbol{\lambda}(T, e),\ e \in E$. We define its associated polyhedron $P_{\boldsymbol{\Lambda}}$ as the vector sum of the convex hull of the row vectors $(\boldsymbol{\lambda}_T,\ T \in \mathcal{T})$ and the nonnegative orthant. This polyhedron can be represented as

$$P_{\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|E|}_{\geq 0} \;\middle|\; \exists \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \left(\boldsymbol{\Lambda}' \boldsymbol{\alpha} \leq \boldsymbol{x} \ \wedge\ \boldsymbol{\alpha}' \mathbf{1} \geq 1 \right) \right\}. \tag{3}$$

The *blocker* of $P_{\boldsymbol{\Lambda}}$ is the polyhedron defined as

$$bl(P_{\boldsymbol{\Lambda}}) := \left\{ \boldsymbol{y} \in \mathbb{R}^{|E|}_{\geq 0} \;\middle|\; \boldsymbol{y}' \boldsymbol{x} \geq 1 \ \forall \boldsymbol{x} \in P_{\boldsymbol{\Lambda}} \right\}. \tag{4}$$

For each vertex $\boldsymbol{\omega} = (\boldsymbol{\omega}_e, e \in E)$ of the blocker, define the quantity

$$\theta(\boldsymbol{\omega}) := \frac{1}{\sum_{e \in E} \boldsymbol{\omega}_e} \left(1 - \sum_{e \in E} \boldsymbol{\omega}_e \boldsymbol{\mu}_e \right). \tag{5}$$

A vertex of the blocker is called critical if it maximizes the quantity $\theta(\boldsymbol{\omega})$, i.e., $\theta(\boldsymbol{\omega}) = \max_{\tilde{\boldsymbol{\omega}}} \theta(\tilde{\boldsymbol{\omega}})$. Finally, let $\tilde{\theta}$ denote the maximum quantity.

In [13], it has been shown that every Nash equilibrium strategy for the attacker is a critical vertex or a convex combination of critical vertices, and that the attacker's equilibrium payoff is $\theta^* = \max(0, \tilde{\theta})$. As a consequence, if this blocker can be "efficiently" characterized, then an efficient algorithm can be derived to solve the maximization problem and, hence, the game.

2.4 Vulnerability/Robustness Metric

In the analysis of the general NBG [13, Chap. 4], it has been shown that θ^* is a property of (i.e., solely determined by) the topology of the network, the communication model, and the attack costs $\boldsymbol{\mu}$. Furthermore, this unique equilibrium payoff reflects both the network operator's expected loss due to attack as well as the attacker's willingness to attack. For a given $\boldsymbol{\mu}$, a low θ^* indicates that operating the network has low expected loss due to attack, that is, the network is robust against attacks. If, on the other hand, θ^* is high, then the expected loss is also high, and the network can be considered vulnerable. As such, θ^* has been proposed [7] as a measure of network topology vulnerability (i.e., inverse robustness) in an adversarial environment. Another property of θ^* is that, when $\boldsymbol{\mu} = \mathbf{0}$ (the case of the most powerful attacker), it can be related to well-known graph-theory notions. For instance, in the All-to-One model, θ^* was shown to be the inverse of the *persistence* of the graph of the network [10], a metric that has previously been proposed in [14] to quantify graph robustness (although in a

non-game theoretic framework). In the All-to-All model with constant loss [7], θ^* can be related the *spanning tree packing number of the graph* [15]. In the All-to-All model with linear loss [9], θ^* is closely bounded by the *Cheeger constant* [16] (also called the *edge-expansion*) of the graph. In the Supply-Demand model [6], the metric is equal to the maximum average flow traversing an edge-cut, where the average is obtained by dividing the total flow by the size of the edge-cut.

As a metric for robustness, understanding the computational complexity of calculating θ^* is of primal importance. In the next section, we discuss the complexity of computing a Nash equilibrium in the unconstrained NBG model.

3 Computational Complexity of the Unconstrained Game

In this section, we show that solving a NBG is NP-hard in general. Recall that computing a Nash equilibrium in general two-player games has been shown to be PPAD-complete. Zero-sum, two-player games, on the other hand, can be cast as linear programs and, hence, can be solved in polynomial time using linear programming tools. In all these cases, the input of the computational problem is assumed to be the payoff matrix. For NBG models however, only an *implicit* description of the payoff matrix is available. In addition, the payoff matrix is potentially exponential in size, which makes NBG models even more challenging to deal with. The following theorem shows that, indeed, computing a NE for a general blocking game is NP-hard. We prove this by reducing a well-known NP-hard problem, the Knapsack Problem (KP), to the problem of computing the attacker's equilibrium payoff, which we formalize as the Equilibrium Problem (EP). The KP and the EP are formally defined as follows.

Definition 1 (Knapsack Problem [KP]). *Given N items, where item i has weight $\boldsymbol{c}_i$ and value $\boldsymbol{v}_i$, a capacity C, and a value V, is there a subset S whose sum weight is at most C, i.e., $\sum_{i\in S} \boldsymbol{c}_i \leq C$, and whose sum value is at least V, i.e., $\sum_{i\in S} \boldsymbol{v}_i \geq V$?*

Definition 2 (Equilibrium Problem [EP]). *Given a set of elements E, a polynomial-time function $I_{T\in\mathcal{T}}$ for testing $T \in \mathcal{T}$, a polynomial-time function $\boldsymbol{\lambda}(T, e)$, a vector of attack costs $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{|E|}$, and a payoff value p, is the adversary's equilibrium payoff less than or equal to p?*

The above formulation of EP allows us to easily show the computational complexity of all the problems relevant to NBGs. First, if the adversary's equilibrium payoff can be efficiently computed, then EP can also be solved efficiently. Conversely, if EP is NP-hard, then computing the adversary's equilibrium payoff is also necessarily NP-hard. Second, for similar reasons, we also have that computing the equilibrium strategies of the game is also at least as hard as EP.

The following theorem shows that EP is NP-hard.

Theorem 1. *The Knapsack Problem is polynomial-time reducible to the Equilibrium Problem.*

The proof of the theorem can be found in Appendix A.

Thus, solving a NBG is NP-hard in general. Interestingly, however, efficient algorithms have been derived to compute a NE for the models discussed in Subsection 2.1. In the following sections, we introduce a budget constraint and revisit the complexity of computing a NE of the constrained game in those models.

4 Budget Contraints

In the unconstrained NBG model, the operator is only interested in minimizing her expected loss due to attacks, without taking her operating costs into account. In practice, however, network operators also have to take economic goals and constraints into consideration when choosing their strategies. These economic decisions are affected by the topology of the network as links and, hence, feasible collections of links can have varying usage costs.

4.1 Unit Usage / Protection Cost

In [6], a (per unit) usage cost model was introduced and discussed for the particular case of the S-D communication model. Here, we extend this cost model to the general NBG. Recall that $\boldsymbol{\lambda}(T, e)$ quantifies the usage (loss) associated with collection T and link e. We assume that each link e has some *unit usage cost* $\boldsymbol{w}_e$, so that using the link costs $\boldsymbol{w}_e\boldsymbol{\lambda}(T, e)$ to the operator. With this definition, the *total cost of* using *a collection* T is

$$w(T) := \sum_{e \in E} \boldsymbol{\lambda}(T, e)\boldsymbol{w}_e \ ; \tag{6}$$

and the network operator's *expected usage cost of a mixed strategy* $\boldsymbol{\alpha}$ is

$$w(\boldsymbol{\alpha}) := \sum_{T \in \mathcal{T}} \boldsymbol{\alpha}_T w(T) = \sum_{e \in E} \boldsymbol{w}_e \sum_{T \in \mathcal{T}} \boldsymbol{\alpha}_T \boldsymbol{\lambda}(T, e) \ . \tag{7}$$

We assume that, to run the network, the operator has a fixed *budget* $b \in \mathbb{R}_{\geq 0}$ to spend. Therefore, her objective is to minimize the expected loss (see Equation (1)) by choosing an optimal strategy that satisfies her budget constraint. This budget constraint can be formulated in multiple ways. In the following sections, we introduce and study two straightforward formulations, the maximum and the expected (or average) cost budget constraints.

4.2 Maximum Cost Budget Constraint

In the first budget constraint formulation, which we refer to as the *maximum cost constraint* (MCC), we require that for a given budget b, the operator only uses collections whose total costs (see Equation (6)) are less than or equal to b. Formally, the pure strategy set of the operator is restricted to

$$\mathcal{T}^{(b)} = \{T \in \mathcal{T} \mid w(T) \leq b\} \ . \tag{8}$$

The maximum cost constraint is best-suited for budget limits that are determined by the amount of preallocated resources available. In this case, the cost of a link can be the amount of resources needed (e.g., energy consumption) to operate the link and the budget limit can be the amount of resources available (e.g., amount of power available).

4.3 Expected Cost Budget Constraint

The maximum cost constraint misses to capture certain situations. For instance, when the amount of allocated resources can be modified during operation, e.g., resources can be leased, the budget limit should apply to the average or, equivalently, the expected cost of a strategy during continuous periods of operation. Thus, in our second budget constraint formulation, which we will refer to as the *expected cost constraint* (ECC), we only require the expected (or average) cost of the operator to not exceed the budget limit.

Under the *expected cost constraint* with a budget limit b, the operator can employ a mixed strategy only if its expected cost (see Equation (7)) is less than or equal to b. Formally, the set of mixed strategies available to the operator is

$$\mathcal{A}^{(b)} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|} \middle| w(\boldsymbol{\alpha}) \leq b \right\} . \tag{9}$$

Note that the above formulation generalizes the classic notion of mixed strategies in game-theory, where the set of mixed strategies is always the set of *all* distributions over the set of pure strategies. Here, a mixed strategy is chosen from a predefined subset of distributions.

4.4 Constrained Game

Having defined the set of available strategies (pure for MCC and mixed for ECC), we can now setup the constrained game in a similar way to the unconstrained game presented in Subsection 2.2. We are interested in mixed strategy Nash equilibria, where the operator picks a distribution $\boldsymbol{\alpha}$ over $\mathcal{T}^{(b)}$ (for MCC) or from the set $\mathcal{A}^{(b)}$ (for ECC), while the attacker chooses a distribution $\boldsymbol{\beta}$ over the set of links. The attacker's Nash equilibrium payoff is denoted $\theta^*(b)$ for a game with budget limit b.

Using the same interpretation as in Subsection 2.2, the attacker's NE payoff $\theta^*(b)$ can be used to quantify the vulnerability (i.e., inverse robustness) of the network when the operator's budget is b. By varying b, one can draw the Pareto frontier between the region of achievable vulnerability/budget points and the region of unachievable ones, as was done in [6] for the particular case of S-D networks with the maximum cost constraint.

Remark In the next two sections, we discuss the complexity of solving the constrained blocking game. However, since the unconstrained NBG is in general NP-hard (see Theorem 1), we readily have that solving a NBG under a budget constraint[6] is also NP-hard in general. Therefore, we focus our discussion on

[6] The unconstrained game is the special case of $b \to \infty$.

the communication models introduced in Subsection 2.1, for which there exist efficient algorithms to compute the NE payoff in the unconstrained game.

5 NP-Hardness of the Maximum Cost Constraint

In this section, we show that computing the equilibrium payoff of the network blocking game with a maximum cost budget constraint is NP-hard for the models that were previously shown to be efficiently computable without a budget constraint.

Theorem 2. *Computing the NE payoff with a maximum cost budget constraint is NP-hard for the (a) S-D communication model, the (b) All-to-All communication model, and the (c) All-to-One communication model.*

Proof. We show NP-hardness by reducing a well-known NP-hard problem, the *Partition Problem* (*PP*) [17], to the problem of deciding whether the equilibrium payoff in a given network model with a maximum cost constraint is at most a certain value. We refer to the latter problem as the *Equilibrium Problem with Maximum Cost Constraint* (EPMAX).

Definition 3 (Partition Problem [PP]). *Given a multiset of positive integers* $\{x_1, \ldots, x_n\}$*, is there a partitioning of the multiset into two disjoint subsets* A *and* B *such that* $\sum_{x \in A} x = \sum_{x \in B} x$ *?*

Definition 4 (Equilibrium Problem with Maximum Cost Constraint [EPMAX]). *Given a communication model, a network* G*, a budget limit* b*, and a payoff value* p*, is the adversary's equilibrium payoff less than or equal to* p*?*

For each communication model, we show how an instance of *EPMAX* (i.e., a network, a budget limit and a payoff value) can be constructed in polynomial time from an instance of *PP*. Since the proof techniques follow the same lines for all models, we only give a full proof for the S-D model. For the All-to-All model, we describe the main points of the proof in Appendix B without providing the details. For the All-to-One model, the proof can be found in [12].

To simplify the notations in our proofs, we also define the *expected loss* of an edge $e \in E$ in a given operator strategy $\boldsymbol{\alpha}$ as

$$L(e) = \sum_{T \in \mathcal{T}} \boldsymbol{\alpha}_T \boldsymbol{\lambda}(T, e) \ . \tag{10}$$

Proof of Theorem 2 for the S-D Communication Model

Given an instance of *PP*, we build an instance of *EPMAX* as follows.

- Let the topology of the network be the following (see Figure 1): There is one source node, denoted by s, one sink node, denoted by d, and $3n - 1$ other nodes, which are denoted by 1_a, 1_b, 1, 2_a, 2_b, 2, ..., n_a, and n_b.

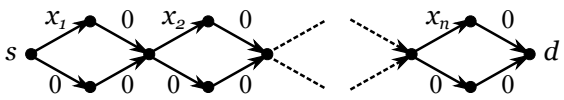

Fig. 1: Illustration for the proof of Theorem 2 for the S-D model. Numbers along the edges indicate unit costs.

Node s is connected to nodes 1_a and 1_b with edges having unit costs of x_1 and 0, respectively. Nodes i_a and i_b, $i < n$, are connected to node i with edges having zero unit cost. Node i is connected to nodes $(i+1)_a$ and $(i+1)_b$ with edges having unit costs of x_{i+1} and 0, respectively. Finally, nodes n_a and n_b are connected to node d with edges having zero unit cost.

- Let the capacity of the links and the amount of goods to be moved from s to d be 1.
- Let the operator's budget be $b = \frac{1}{2}\sum_{i=1}^{n} x_i$.
- Let the equilibrium payoff value be $p = \frac{1}{2}$.

We claim that the equilibrium payoff in the above network is greater than p iff *PP* does not have a solution.

First, we assume that the set can be partitioned into two subsets A and B of equal sum, that is, *PP* has a solution. In this case, we have to show that the equilibrium payoff is at most $\frac{1}{2}$. First, notice that since the total amount of goods to be moved from s to d is 1, the set of feasible integer flows is equal to the set of s-d paths as the amount of flow on each edge is either 0 or 1. Now, we show that there exist two disjoint paths (or flows) that satisfy the budget constraint. The first path (i.e., set of links with positive flow values) consists of the edges $(i-1, i_a)$ and (i_a, i) for each $x_i \in A$ and $(i-1, i_b)$ and (i_b, i) for each $x_i \notin A$. The second path consists of the remaining edges. In other words, the first flow takes the "path above" whenever $x_i \in A$ and the "path below" whenever $x_i \notin A$, while the second flow does the contrary. It is easy to see that the cost of both flows is $\sum_{x_i \in A} x_i = \sum_{x_i \in B} x_i = \frac{1}{2}\sum_i x_i$; thus, they satisfy the maximum budget constraint. By assigning $\frac{1}{2}$ probability to each flow, we obtain an operator strategy in which the expected loss of every edge is at most $\frac{1}{2}$. If the operator employs this strategy, the payoff of every pure and, consequently, every mixed adversarial strategy is at most $\frac{1}{2}$. Therefore, the equilibrium payoff has to be at most $\frac{1}{2}$.

Second, we assume that the set cannot be partitioned into two subsets of equal sum, that is, *PP* does not have a solution. If the equilibrium payoff of the game were at most $\frac{1}{2}$, then there would exist an operator strategy $\boldsymbol{\alpha}$ in which the expected loss of every edge is at most $\frac{1}{2}$. We show that no such strategy can exist.

Because of the maximum cost budget constraint, the cost of every pure strategy is less than or equal to $b = \frac{1}{2}\sum_i x_i$. Moreover, this inequality is *strict* as every pure strategy is an s-d path and, if its cost is equal to b, there must exist a subset of links $I \subsetneq \{1, 2, \ldots, n\}$ such that $\sum_{i \in I} x_i = b$. By letting $A = \{x_i \mid i \in I\}$ and $B = \{x_i \mid i \notin I\}$ we get a solution for *PP*, which would contradict the assumption that the set cannot be partitioned. Thus, the cost of every pure strategy

is strictly less than b and, as a consequence, the expected cost of every mixed strategy is also strictly less than b; formally,

$$\sum_{e \in E} L(e)\boldsymbol{w}_e < b = \frac{1}{2}\sum_{i=1}^{n} x_i = \sum_{e \in E} \frac{1}{2}\boldsymbol{w}_e \ . \tag{11}$$

Now, recall that the expected loss $L(e)$ of an edge e in the S-D model is equal to the expected amount of flow on that edge. Since the total amount of goods to be moved is equal to 1 and since each pair of "above" and "below" edges (e.g., e_a and e_b) is an s-d cut, the sum of the flows on any pair of "above" and "below" edges is at least 1. Thus, for every pair of edges e_a and e_b, $L(e_a) + L(e_b) \geq 1 = \frac{1}{2} + \frac{1}{2}$. Combined with our initial assumption that the expected loss of each edge is at most $\frac{1}{2}$, we have that

$$\forall e \in E : \ L(e) = \frac{1}{2} \tag{12}$$

and

$$\sum_{e \in E} L(e)\boldsymbol{w}_e = \sum_{e \in E} \frac{1}{2}\boldsymbol{w}_e \ . \tag{13}$$

But this leads to a contradiction with Equation 11, showing that if PP does not have a solution, then the equilibrium payoff is greater than $\frac{1}{2}$, which concludes our proof. □

6 Efficient Algorithms for the Expected Cost Constraint

In this section, we show how the expected cost constrained game can be solved efficiently for the models introduced in Subsection 2.1. In Subsection 2.3, we gave a derivation of the attacker's Nash equilibrium payoff in the unconstrained game model using the theory of blocking pairs of polyhedra. In this section, we use a similar derivation to show how polynomial-time algorithms can be derived to solve the game with the expected cost constraint. The same detailed analytical steps presented in [13, Chap. 4] (for the unconstrained game) can be followed to show the same results for the constrained game. In this case, the definition of the polyhedron $P_{\boldsymbol{\Lambda}}$ in Equation (3) includes an additional linear inequality (given by Equation (9)) that corresponds the budget constraint. Since the expected cost $w(\boldsymbol{\alpha})$ in Equation (9) can also be formulated as $w(\boldsymbol{\alpha}) = \boldsymbol{w}'\boldsymbol{\Lambda}'\boldsymbol{\alpha}$, the constrained polyhedron can be written as

$$P_{\boldsymbol{\Lambda}} := \left\{ \boldsymbol{x} \in \mathbb{R}^{|E|}_{\geq 0} \ \middle| \ \exists \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \left(\boldsymbol{\Lambda}'\boldsymbol{\alpha} \leq \boldsymbol{x} \ \wedge \ \boldsymbol{\alpha}'\mathbf{1} \geq 1 \ \wedge \ \boldsymbol{w}'\boldsymbol{\Lambda}'\boldsymbol{\alpha} \leq b \right) \right\}. \tag{14}$$

Notice that the definition of $P_{\boldsymbol{\Lambda}}$ above involves the matrix $\boldsymbol{\Lambda}$, which is generally exponential in size. As a consequence, this definition of $P_{\boldsymbol{\Lambda}}$ cannot be directly used to efficiently solve the game.

To derive a polynomial-time solution for the ECC model, we first characterize the blocker $bl(P_{\boldsymbol{\Lambda}})$ of $P_{\boldsymbol{\Lambda}}$ using a set of linear equations whose cardinality is

polynomial in the size of the network. We do so by showing that if a polynomial-size characterization exists for the unconstrained polyhedron, then there also exists one for the blocker of the constrained game. We then show how one can use linear programming tools to efficiently compute the equilibrium payoff based on a polynomial-size characterization of the blocker. Finally, we provide a characterization for each of the models discussed in Section 2.1.

Let the polynomial-size linear characterization of the polyhedron $P_{\boldsymbol{\Lambda}}$ be

$$P_{\boldsymbol{\Lambda}} = \{\boldsymbol{x} \mid \exists \boldsymbol{f}\, (\boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \wedge \boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{d})\} \tag{15}$$

for the unconstrained game, where $\boldsymbol{f} \in \mathbb{R}^k_{\geq 0}$ is a vector of polynomial length (i.e., k is a polynomial function of the network size), $\boldsymbol{S}$ is a mapping to the mixed strategy space, and $\boldsymbol{C}$, $\boldsymbol{d}$ are linear constraints. Then, the expected cost constrained polyhedron is characterized by

$$P_{\boldsymbol{\Lambda}} = \{\boldsymbol{x} \mid \exists \boldsymbol{f}\, (\boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \wedge \boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{d} \wedge \boldsymbol{w}'\boldsymbol{S}\boldsymbol{f} \leq b)\}\ . \tag{16}$$

The following theorem gives a polynomial-size characterization of the blocker in the expected cost constrained game.

Theorem 3. *The blocker of the polyhedron defined as*

$$P_{\boldsymbol{\Lambda}} = \{\boldsymbol{x} \mid \exists \boldsymbol{f}\, (\boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \wedge \boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{d} \wedge \boldsymbol{w}'\boldsymbol{S}\boldsymbol{f} \leq b)\} \tag{17}$$

is

$$bl(P_{\boldsymbol{\Lambda}}) = \left\{\boldsymbol{y} \mid \exists K, \boldsymbol{g}, \boldsymbol{h}\, \left(\boldsymbol{g} \leq \boldsymbol{y} \wedge \boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{S}'\boldsymbol{w}K + \boldsymbol{S}'\boldsymbol{g} \wedge \boldsymbol{d}'\boldsymbol{h} - bK \geq 1\right)\right\}\ , \tag{18}$$

where $K \in \mathbb{R}_{\geq 0}$, $\boldsymbol{g} \in \mathbb{R}^{|E|}_{\geq 0}$, *and* $\boldsymbol{h} \in \mathbb{R}^l_{\geq 0}$ *(*l *is the length of* $\boldsymbol{d}$*).*

Proof. We prove Equation (18) in two steps:

– RHS of Equation (18) $\subseteq$ $bl(P_{\boldsymbol{\Lambda}})$: We have to show that, for any $\tilde{\boldsymbol{y}}$ that satisfies the constraints of the RHS with some $\tilde{\boldsymbol{g}}$, $\tilde{\boldsymbol{h}}$ and $\tilde{K}$, it holds that $\tilde{\boldsymbol{y}}'\boldsymbol{x} \geq 1$ for every $\boldsymbol{x} \in P_{\boldsymbol{\Lambda}}$. We can formulate this as a linear programming problem as follows:

$$\text{Minimize } \tilde{\boldsymbol{y}}'\boldsymbol{x} \tag{19}$$

subject to

$$\boldsymbol{w}'\boldsymbol{S}\boldsymbol{f} \leq b \tag{20}$$

$$\boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \tag{21}$$

$$\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{d}\ , \tag{22}$$

where $\boldsymbol{f} \in \mathbb{R}^k_{\geq 0}$.
Observe that the constraints of the above LP correspond to the characterization of $P_{\boldsymbol{\Lambda}}$; consequently, the above linear program's set of feasible solutions projected to $\boldsymbol{x}$ is $P_{\boldsymbol{\Lambda}}$. Thus, we have to show that the value of the above linear program is at least 1. To see this, consider the dual linear program:

$$\text{Maximize } \boldsymbol{d}'\boldsymbol{h} - bK \tag{23}$$

subject to

$$\boldsymbol{g} \leq \tilde{\boldsymbol{y}} \tag{24}$$

$$\boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{S}'\boldsymbol{w}K + \boldsymbol{S}'\boldsymbol{g}\ , \tag{25}$$

where $K \in \mathbb{R}_{\geq 0}$, $\boldsymbol{g} \in \mathbb{R}_{\geq 0}^{|E|}$, and $\boldsymbol{h} \in \mathbb{R}_{\geq 0}^{l}$.

Since $\tilde{\boldsymbol{y}}$ satisfies the constraints of the RHS of Equation (18) with $\tilde{K}, \tilde{g}, \tilde{h}$, we have that $(\tilde{K}, \tilde{g}, \tilde{h})$ is a feasible solution. Furthermore, we also have that the objective function for this solution is at least 1. Thus, the value of the linear program has to be at least 1.

– $bl(P_{\boldsymbol{\Lambda}}) \subseteq$ RHS of Equation (18): We have to show that every $\tilde{\boldsymbol{y}} \in bl(P_{\boldsymbol{\Lambda}})$ satisfies the constraints of the RHS. To see this, first consider the linear program from the first part of the proof. Since $\tilde{\boldsymbol{y}}$ blocks every $\boldsymbol{x} \in P_{\boldsymbol{\Lambda}}$, we have that the value of the linear program is at least 1. Now, consider an optimal solution $\tilde{K}, \tilde{g}, \tilde{h}$ of the dual linear program. Since the value of the linear program is at least 1, we have that $1 \leq \boldsymbol{d}'\tilde{\boldsymbol{h}} - b\tilde{K}$. We also have $\tilde{\boldsymbol{g}} \leq \tilde{\boldsymbol{y}}$ and $\boldsymbol{C}'\tilde{\boldsymbol{h}} \leq \boldsymbol{S}'\boldsymbol{w}\tilde{K} + \boldsymbol{S}'\tilde{\boldsymbol{g}}$ from the constraints. Thus, $\tilde{\boldsymbol{y}}$ satisfies the constraints of the RHS of Equation (18) with $\tilde{K}, \tilde{g}, \tilde{h}$. □

Recall that our goal is to compute $\theta^* = \max\{\tilde{\theta}, 0\}$ in polynomial time. The most straightforward solution is to formulate this as an optimization problem subject to the set of linear constraints given by the above characterization. Unfortunately, the objective function θ cannot be expressed as a linear function because of the division with $\mathbf{1}'\boldsymbol{y}$. Thus, to formulate the problem as a linear program, we introduce a variable ϕ which measures $\frac{1}{\mathbf{1}'\boldsymbol{y}}$ and scale the original variables. The resulting linear program is

$$\text{Maximize } \phi - \boldsymbol{\mu}'\boldsymbol{\beta} \tag{26}$$

subject to

$$\mathbf{1}'\boldsymbol{\beta} = 1 \tag{27}$$

$$\boldsymbol{g} \leq \boldsymbol{\beta} \tag{28}$$

$$\boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{S}'\boldsymbol{w}K + \boldsymbol{S}'\boldsymbol{g} \tag{29}$$

$$\boldsymbol{d}'\boldsymbol{h} - bK \geq \phi\ , \tag{30}$$

where $K, \phi \in \mathbb{R}_{\geq 0}$, $\boldsymbol{\beta}, \boldsymbol{g} \in \mathbb{R}_{\geq 0}^{|E|}$, and $\boldsymbol{h} \in \mathbb{R}_{\geq 0}^{l}$.

All-to-All Communication Model In [9], it was shown that the mixed strategy space of the operator in the All-to-All model be characterized using multi-commodity flows. In this characterization, there exists a commodity for each node. For each commodity, the corresponding node is a sink, while all the other nodes are a sources with a uniform supply. It was shown that, if the total amount of flow transported is at least 1, the vector representing the sum flows on each edge is an element of the polyhedron, and vice versa.

This can formulated as a set of linear constraints with $|V|+|V|\cdot|E|$ variables (the uniform supply value for each commodity and the flow along each edge for each commodity) and $|V|\cdot|V|+1$ constraints (flow conservation at each node for each commodity and the constraint on the total amount of flow transported). Then, by applying Theorem 3, we have a polynomial-size characterization of the constrained blocker:

$$bl(P_{\boldsymbol{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{|E|}_{\geq 0} \;\middle|\; \exists \pi : V \times V \mapsto \mathbb{R}_{\geq 0}, K \in \mathbb{R}_{\geq 0} \Big(\right.$$
$$\forall r \in V : \sum_{v \in V} \pi_r(v) - bK \geq 1 \;\wedge$$
$$\left. \forall r \in V, e = (u,v) \in E : |\pi_r(u) - \pi_r(v)| \leq \boldsymbol{y}_e + \boldsymbol{w}_e K \Big) \right\}, \quad (31)$$

where $\pi_r(r) \equiv 0$ by definition to simplify the notation.

S-D Communication Model Based on [6], we can characterize the polyhedron for the S-D model using network flows. Then, from Theorem 3, we have that the constrained blocker has the following polynomial-size characterization:

$$bl(P_{\boldsymbol{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{|E|}_{\geq 0} \;\middle|\; \exists \pi : V \mapsto \mathbb{R}, K \in \mathbb{R}_{\geq 0} \Big(\right.$$
$$\sum_{v \in V} \pi(v)(s(v) - d(v)) - bK \geq 1 \;\wedge$$
$$\left. \forall e = (u,v) \in E : \pi(u) - \pi(v) \leq \boldsymbol{y}_e + \boldsymbol{w}_e K \Big) \right\}. \quad (32)$$

All-to-One Communication Model In [10], it was shown that the mixed strategy space of the operator in the All-to-One model can be characterized using special multi-source flows. By combining this result with Theorem 3, we can show that the constrained blocker has the following polynomial-size characterization:

$$bl(P_{\boldsymbol{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{|E|}_{\geq 0} \;\middle|\; \exists \pi : V \setminus \{r\} \mapsto \mathbb{R}_{\geq 0}, K \in \mathbb{R}_{\geq 0} \Big(\right.$$
$$\sum_{v \in V} \pi(v) - bK \geq 1 \;\wedge$$
$$\left. \forall e = (u,v) \in E : \pi(u) - \pi(v) \leq \boldsymbol{y}_e + \boldsymbol{w}_e K \Big) \right\}, \quad (33)$$

where $\pi(r) \equiv 0$ by definition to simplify the notation.

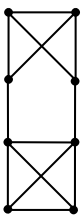

Fig. 2: All-to-All network

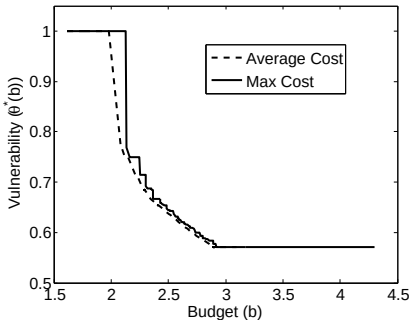

Fig. 3: Vulnerability/budget tradeoff

7 Application Example: Vulnerability/Budget Tradeoff

As it was mentioned earlier, by varying the budget limit b, one can draw the Pareto frontier between the region of achievable vulnerability/budget points and the region of unachievable ones. Here, we illustrate this using the All-to-All communication model on the topology depicted in Figure 2. The link costs $\boldsymbol{w}_e$ are randomly chosen between 0 and 0.6, which makes the average cost of a spanning tree equal to 2.1. For each value of b, a game is played with the defender's strategy set given by Equation (8) for the maximum cost constraint (MCC) and by Equation (9) for the expected (or average) cost constraint (ECC). In all games, the attacker's strategy set is the set of all links and the cost of attack is $\boldsymbol{\mu} = \mathbf{0}$. Figure 3 shows the vulnerability $\theta^*(b)$ as a function of the budget b for both the MCC and the ECC. Observe that the two curves are very close to each other, but vulnerability for the MCC is always at least as high as for the ECC.

8 Conclusions & Future Work

In this paper, we have generalized *network blocking games* by introducing budget constraints on the operator. This generalization allows the application of network blocking games in scenarios where the budget of the network operator is limited. We have studied two budget constraint formulations: the *maximum cost* and the *expected* (or *average*) *cost constraints*.

Network blocking games are used to quantify the robustness of topologies in the presence of a strategic adversary, and the equilibrium payoffs of the games are used as such quantifications. As the greatest challenge to computing the equilibrium in practice is the exponential size of the implicitly given payoff matrix, we have focused our work on computational complexity: we have shown that the maximum cost formulation leads to NP-hard problems, and proposed efficient solutions for the expected cost formulation.

Proving that the maximum cost formulation leads to NP-hard problems was a very important first step. Since we now know that no polynomial-time algorithm can solve the game under the MCC (for the discussed models), an interesting

future work is finding polynomial-time approximation algorithms or efficient heuristics. Another interesting future direction is the study of the cost-security tradeoff problem, where the operator has to maximize security and minimize budget at the same time.

Acknowledgement

This paper has been supported by HSN Lab, Budapest University of Technology and Economics [7], NIST-ARRA Program award 70NANB10H026, and NIST grant award 70NANB13H012, through the Univ. of Maryland, College Park.

A Proof of Theorem 1

Proof. Given an instance $(\boldsymbol{c}, \boldsymbol{v}, C, V)$ of the Knapsack Problem, we construct an instance $(E, I_{T\in\mathcal{T}}, \boldsymbol{\lambda}(T,e), p)$ of the Equilibrium Problem as follows:

- Let $E = \{1, \ldots, N\}$,
- $I_{T\in\mathcal{T}} = \begin{cases} \text{true} & \text{if } \sum_{i\in T} \boldsymbol{c}_i \leq C \\ \text{false} & \text{otherwise,} \end{cases}$
- $\boldsymbol{\lambda}(T,e) = \frac{1}{\sum_{i\in T} \boldsymbol{v}_i}$,
- $\boldsymbol{\mu} = \mathbf{0}$,
- $p = \frac{1}{V}$.

Observe that we define $\boldsymbol{\lambda}(T,e)$ such that its value does not depend on e. Consequently, the payoff of the game does not depend on the adversary's strategy, it only depends on the operator's strategy. To simplify our proof, we will let $\boldsymbol{\lambda}(T)$ denote $\boldsymbol{\lambda}(T,e)$ for any e.

It is easy to see that both $I_{T\in\mathcal{T}}$ and $\boldsymbol{\lambda}(T)$ can be computed in polynomial time as they only require summing over a given set (and comparing the sum with a constant or computing a reciprocal). Furthermore, every step of the reduction can be carried out in time and space that is polynomial in the size of the Knapsack Problem instance.

We claim that there exists a subset $S \subseteq \{1, \ldots, N\}$ whose sum weight is at most W and whose sum value is at least V if and only if the adversary's equilibrium payoff is less than or equal to p (since $\boldsymbol{\mu} = \mathbf{0}$).

First, assume that there exists a subset S satisfying the constraints of the Knapsack Problem. Then, consider the operator strategy $\boldsymbol{\alpha}^*_S = 1$ (i.e., the strategy that uses only subset S). If the operator uses this strategy, her loss is

$$\boldsymbol{\lambda}(S) = \frac{1}{\sum_{i\in S} \boldsymbol{v}_i} = \frac{1}{V} = p \ . \tag{34}$$

Therefore, the operator's equilibrium loss and, hence, the adversary's equilibrium payoff is at most p.

Second, assume that there does not exist a subset satisfying the constraints of the Knapsack Problem. This implies that, for every $T \in \mathcal{T}$,

$$\boldsymbol{\lambda}(T) = \frac{1}{\sum_{i\in T} \boldsymbol{v}_i} > \frac{1}{V} = p \ . \tag{35}$$

[7] http://www.hsnlab.hu

Consequently, the expected loss for any operator strategy $\boldsymbol{\alpha}^*$ is

$$\sum_{T \in \mathcal{T}} \boldsymbol{\alpha}^*_T \underbrace{\boldsymbol{\lambda}(T)}_{>p} > p \ . \tag{36}$$

Thus, the adversary's equilibrium payoff has to be greater than p. □

B Proof of Theorem 2 for the All-to-All model

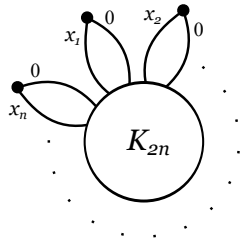

Fig. 4: Illustration for the proof of Theorem 2 for the All-to-All model.

For the All-to-All communication model, we construct an instance of *EPMAX* from an instance of *PP* as follows:

- Let the network topology be the following (see Figure 4): There is a large clique that consists of $2n$ nodes, and there are n "outer" nodes, to which we refer as node 1, node 2, ..., node n. Each node i, $i = 1, \ldots, n$, is connected to two distinct nodes of the clique with edges having unit costs of x_i and 0, such that every node in the clique is connected to exactly one outer node. Finally, edges between two nodes in the clique have zero unit cost.
- Let the operator's budget be $b = \frac{1}{2} \sum_{i=1}^{n} x_i$.
- Let the equilibrium payoff value be $p = \frac{1}{2}$.

We claim that the equilibrium payoff in the above network is greater than $\frac{1}{2}$ iff *PP* does not have a solution.

As in the previous proof, we first assume that *PP* has a solution (A, B) and use it to derive an operator strategy in which the expected loss of every edge is at most $\frac{1}{2}$. According to this strategy, the operator chooses a spanning tree as follows. First, she chooses either A or B with equal probability $(\frac{1}{2}, \frac{1}{2})$. Second, she connects each outer node i to the clique with exactly one edge: if x_i belongs to the chosen set, she uses the edge that has cost x_i; otherwise, she uses the other edge. Third, she completes the spanning tree by choosing a star subgraph of the clique uniformly at random. We show that the expected loss of every link is at most $\frac{1}{2}$: First, each outer edge e is used with probability $\frac{1}{2}$ and its removal cuts off at most 1 node; thus, $L(e) \leq \frac{1}{2}$. Second, each link e inside the clique is used with probability $\frac{1}{n}$ and its removal cuts off at most 2 nodes; thus, $L(e) \leq \frac{2}{n}$.

Next, we assume that *PP* does not have a solution and use the same argument as before to show that the cost of every pure strategy and, hence, the expected cost of every mixed strategy is strictly less than b, i.e.,

$$\sum_{e \in E_{\text{outer}}} \boldsymbol{w}_e L(e) < b = \frac{1}{2} \sum_i x_i = \sum_{e \in E_{\text{outer}}} \frac{1}{2} \boldsymbol{w}_e \ , \tag{37}$$

where E_{outer} is the set of outer links. Now, consider an arbitrary pair of edges e_a and e_b that connect an outer node to the clique. It can be shown that $L(e_a) + L(e_b) \geq 1$. If there were an operator strategy in which the expected loss of every edge is at most $\frac{1}{2}$, then it would follow that $\forall e \in E_{\text{outer}} : \ L(e) = \frac{1}{2}$. This would lead to a contradiction with Equation 37; thus, no such strategy can exist. □

References

1. Holme, P., Kim, B., Yoon, C., Han, S.: Attack vulnerability of complex networks. Physical Review E **65**(5) (2002) 056109
2. Schneider, C., Moreira, A., Andrade Jr, J., Havlin, S., Herrmann, H.: Mitigation of malicious attacks on networks. Proceedings of the National Academy of Sciences **108**(10) (2011) 3838–3841
3. Grubesic, T., Matisziw, T., Murray, A., Snediker, D.: Comparative approaches for assessing network vulnerability. International Regional Science Review **31**(1) (2008) 88–112
4. Estrada, E.: Network robustness to targeted attacks. the interplay of expansibility and degree distribution. Eur. Phys. Journal B **52**(4) (2006) 563–574
5. DallAsta, L., Barrat, A., Barthélemy, M., Vespignani, A.: Vulnerability of weighted networks. J. of Stat. Mech. **2006**(04) (2006) P04006
6. Gueye, A., Marbukh, V.: A game-theoretic framework for network security vulnerability assessment and mitigation. In: Proc. of 3rd Conference on Decision and Game Theory for Security, Springer (November 2012)
7. Gueye, A., Walrand, J.C., Anantharam, V.: Design of network topology in an adversarial environment. In: Proc. of 1st Int. Conf. on Decision and Game Theory for Security. (2010)
8. Gueye, A., Walrand, J.C., Anantharam, V.: A network topology design game: How to choose communication links in an adversarial environment? In: Proc. of 2nd Int. ICST Conf. on Game Theory for Networks. (2011)
9. Laszka, A., Szeszlér, D., Buttyán, L.: Linear loss function for the network blocking game: An efficient model for measuring network robustness and link criticality. In: Proc. of 3rd Conf. on Decision and Game Theory for Security, Springer (November 2012)
10. Laszka, A., Szeszlér, D., Buttyán, L.: Game-theoretic robustness of many-to-one networks. In: Proc. of 3rd Int. ICST Conf. on Game Theory for Networks. (2012)
11. Daskalakis, C., Goldberg, P., Papadimitriou, C.: The complexity of computing a Nash equilibrium. In: Proc. of 38th Annu. ACM Symp. on Theory of Computing. (2006) 71–78
12. Laszka, A., Gueye, A.: Quantifying All-to-One network topology robustness under budget constraints. In: Proc. of Workshop on Pricing and Incentives in Networks and Systems, ACM (June 2013)
13. Gueye, A.: A Game Theoretical Approach to Communication Security. PhD thesis, EECS Department, University of California, Berkeley (Mar 2011)
14. Cunningham, W.: Optimal attack and reinforcement of a network. Journal of the ACM **32**(3) (1985) 549–561
15. Palmer, E.M.: On the spanning tree packing number of a graph: a survey. Discrete Mathematics **230**(1) (2001) 13–21
16. Chung, F.: Laplacians and the cheeger inequality for directed graphs. Annals of Combinatorics **9**(1) (2005) 1–19

17. Mertens, S.: The easiest hard problem: Number partitioning. Comput. Complex. and Stat. Phys. **125**(2) (2006) 125–139