24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Strategic Monitor Placement against Malicious Flows

Vincenzo Auletta[1] and **Giuseppe De Nittis**[2] and **Diodato Ferraioli**[3] and **Nicola Gatti**[4] and **Domenico Longo**[5]

Abstract. Security Games have been widely adopted to model scenarios in which one player, the *Defender*, has to decide how to deploy her resources to minimize the loss that can be caused by an attack performed by another player, the *Attacker*, aiming at maximizing such loss. In the present paper, we focus on scenarios in which the Defender has *lexicographic-like* preferences on the targets, being *primarily* interested in defending the integrity of a subset of the targets and, only *secondarily*, to reduce the amount of the other damaged targets. Our central motivation for studying this problem comes from the need to reduce the impact of *malicious flows* in networks, that can be either physical, like cities, or virtual, e.g., social networks.

In this work, we introduce a new class of security games to model these scenarios, characterizing it and proving the NP-hardness of computing a leader-follower equilibrium, which is the most appropriate solution concept for this setting. To compute such an equilibrium, we then provide an exact exponential-time algorithm, capable of exploiting the topological properties of the network. Finally, we show that, with opportune optimizations, this algorithm can work efficiently even on network of 10000 nodes.

1 INTRODUCTION

Due to the terrible waves of attacks that invested both US and Europe in the last decades, guaranteeing protection to people and critical infrastructures is nowadays a priority for every country. This need raised the attention of the Artificial Intelligence scientific community to *Security Games* (SGs). They are mathematical models customarily used to represent strategic interactions between opposing agents. SGs are 2-player games set in an environment that contains valuable targets: one player, the Attacker, aims at compromising such targets while the Defender must decide how to schedule her scarce resources to maximize the protection over these critical areas. Generally, it is assumed that players choose their policies *asynchronously*: the Defender announces her choice and then the Attacker, after having observed the Defender commitment, undertakes her action. This reflects natural scenarios in which, for examples, thiefs can observe the patrolling routes of guards before committing a crime. In this context, the most suitable game-theoretic solution concept is the *leader-follower Stackelberg equilibrium* [33], with the Defender acting as the leader and the Attacker as the follower. Since computing such an equilibrium in normal-form games requires time that is polynomial in the size of the problem [6], many security games have been proposed to solve real-world problems. Their development led to actual deployments in use today by major security agencies in several domains. For example, they are employed by LAX airport for checkpoint placement [24], the Federal Air Marshal Service for scheduling undercover air marshals [29] and the US Coast Guard for port patrolling [27]. More recently, they have also been tested for airport passenger screening [4], wildlife protection [10] and in the cybersecurity domain [25], where the goal is to find optimal assignments of the incoming cyber alerts to analysts in the presence of a strategic adversary.

Our Problem. In the present paper, we focus on scenarios in which the Defender has *lexicographic-like* preferences on the targets, being *primarily* to defend the integrity of a subset of the targets—said *critical*—and, only *secondarily*, to reduce the amount of the other—*non-critical*—damaged targets. Our central motivation for studying this problem comes from the need to reduce the impact of *malicious flows* in networks. Such networks can be either physical, like cities, or virtual, considering connections in social networks.

The enormous popularity of social networks and their lack of an effective control system made them the favorite channel for diffusing either inaccurate and not verified information or even fake news spread by malicious users for their illicit profit. Limiting this *misinformation* diffusion in social networks is nowadays recognized of paramount importance [28, 32]. Indeed, it can lead to undesirable and severe consequences, such as widespread panic, libelous campaigns against competitors, conspiracies, frauds. For example, a large discussion has been run in U.S. on the manipulating power of the social media related to the 2017 presidential campaign [22]. In a different context, many governments are actively working to contrast online campaigns against vaccinations, which are based on the diffusion of non-scientifically confirmed news about relations between vaccines and autism [9]. However, we are often interested in a control system that is also able to guarantee the safety of critical target nodes while limiting the spread of misinformation. For example, one may want to protect teenagers from killer games on social networks, as Blue Whale [26]. A widespread technique to contrast misinformation consists in placing *monitors* on the network, able to analyze all the information flowing through them, recognize dangerous information and eliminate it [38, 2].

The same approach can be adopted to limit the diffusion of violent actions in the cities: for example, a pacific gathering of people have become targets of criminal groups willing to generate terror or to provoke damages. In these cases, monitors could be physical checkpoints people must get through to access some areas. Moreover, in many of these scenarios, there is a *red-zone*, i.e., a forbidden area that must not be accessed, and whose damage would provoke irreparable consequences, thus characterizing it as a critical target.

Due to the large extent of the environment to control and the limited number of resources the Defender can use, she has to cope with

[1] Università degli Studi di Salerno, Italy, email: auletta@unisa.it
[2] Politecnico di Milano, Italy, email: giuseppe.denittis@polimi.it
[3] Università degli Studi di Salerno, Italy, email: dferraioli@unisa.it
[4] Politecnico di Milano, Italy, email: nicola.gatti@polimi.it
[5] Università degli Studi di Salerno, Italy, email: d.longo6@studenti.unisa.it

an allocation problem. Specifically, the question we want to answer is the following: *how should the Defender place a set of monitors, in such a way that they can guarantee the safety of one or more critical target nodes and, at the same time, limit the number of nodes exposed to the attack?*

Our Contribution. In this paper, we define a new class of security games with additional feasibility constraints over the leader's strategy space, and we apply it to settings in which an agent wants to stop the infection started by an adversary. We characterize such games, proving them not to be unilaterally competitive. Then, we study them from a computational perspective, proving that computing the leader-follower Stackelberg equilibrium is NP-hard. Actually, we prove an even stronger result: computing the best-response of both the Attacker and the Defender is an NP-hard problem, even if the Defender does not have any critical target to protect and the source the Attacker uses to begin the infection is known. While a greedy approach allows the Attacker to still compute a good approximation for her best response, we show that this is not the case for the Defender's best response function, proving that such function is not submodular so that many known approximation approaches fail to work for computing it.

Next, we present algorithms to compute the leader-follower Stackelberg equilibrium of our game. Although the well-known *double oracle approach* [12] can be applied to the case without critical targets, the same technique does not work for the case with critical targets and we need to propose a different approach to compute the equilibrium that exploits the structure of the problem. We design an algorithm that returns an exact equilibrium, but requires access to an oracle able to compute the equilibrium of a subgame with no critical targets. Our algorithm works by looking for the Defender's equilibrium strategy: this corresponds to a specific *small* cut in a directed network, and our algorithm proceeds by enumerating cuts of this network to find the one corresponding to the equilibrium. Even though it is possible to enumerate cuts with only polynomial delay between two successive outputs [36], it may be the case that the number of small cuts in a directed graph is exponential in the number of vertices, and thus our algorithm requires exponential time and exponentially many queries to the oracle.

Nevertheless, we show that, by adopting opportune optimizations, our approach allows to compute a solution to the problem for networks of medium size, at the cost of only a small loss in the utility of the Defender.

Related Works. Threats related to misinformation in online social networks recently attracted the attention of the scientific community [7]. Research on how to contrast misinformation concentrated on the problems of recognizing it [35] (and references therein), identifying its sources [19, 21, 13], and limiting its diffusion [5, 38, 2].

Our work focuses on preventing the diffusion of an infection in a network. As described above, both the infection and the network can be either physical or virtual. To limit the infection, two main approaches have been proposed in the literature. The first one, proposed by [5], relies on the idea of contrasting the spread of misinformation by injecting true information in the network. Specifically, a true information campaign is initiated from a subset of highly influential nodes, so that the diffusion of misinformation and true information proceeds in parallel, except for nodes that have received the true information, which will be immune to the misinformation and will not transmit it. The second approach considers a more difficult case in which there is no possibility of positively infecting nodes, but only to place monitors that are able of detecting and blocking misinformation over the network. Most of the literature focuses on scenarios in which the Defender has the only goal of minimizing the number of infected nodes [3, 11, 18, 39]. Recently, [38, 2] studied a novel scenario for monitor placement by requiring not only that the number of nodes exposed to misinformation should be limited, but also that critical targets are not reached. These works study the complexity of the problem and propose algorithms that could be suitable to compute Defender's strategies efficiently. However, such algorithms cannot be exploited in our setting since they assume that the misinformation sources, i.e., nodes from which the attack starts, are known. Nevertheless, it is often the case that the Defender has to set up the monitors before an attack occurs. In this case, an observer may analyze the network traffic, make an estimate on the position of the monitors, and then plan her attack as her best response to the Defender strategy. This peculiarity motivates our study and distinguishes our contribution from that one in the aforementioned works.

A couple of works already analyzed the problem of limiting flow diffusion within the context of Security Games [30, 34]. We will discuss similarities and differences of these works with ours when presenting our problem in details (see Section 2).

2 PROBLEM FORMULATION

Our problem can be modeled as a 2-player normal-form game, played by an Attacker, aiming at maximizing the number of *infected* vertices by deploying malicious sources, and a Defender, willing to block the *infection* by placing monitors on such network. Hereafter, we will use the term *infected* to denote vertices that are reached by the infection. More formally, a vertex u can be infected if there is a path in the network connecting a malicious source to vertex u.

Let $G = (V, E)$ be a (connected) graph modeling a *network*, with $|V| = n$ and $|E| = m$. We assign a weight $p_e \in [0, 1]$ to each edge $e = (u, v) \in E$, representing the probability that u and v can *spread the infection* to each other. For each vertex $u \in V$, we denote as $N(u)$ the set of neighbors of u in G and with $w(u) \in \mathbb{R}$ the value associated to each vertex.Vertices are also characterized by two quantities $c_d(u) \in \mathbb{R}$ and $c_a(u) \in \mathbb{R}$, representing the costs the Defender and the Attacker, respectively, are forced to pay to put a monitor or a source on vertex u. Furthermore, in the network there is a set $T \subseteq V$ of special vertices, called *critical targets*, that the Defender cannot allow being infected. If the network is physical, i.e., a set of streets, critical targets could correspond to nodes where people are located, while in a virtual network, e.g., a social network, such vertices could correspond to profiles of important people that must not be influenced with wrong information, e.g., about fake terrorist attacks.

The infection diffuses according to the *Independent Cascade Model with Monitors* [15]. Specifically, given a set $S \subseteq V$ of *sources*, and a set $M \subseteq V$ of *monitors*, let $S_0 = \emptyset$, $S_1 = S \setminus M$. At each time step $t \geq 1$, $S_{t+1} = S_t$ and, for each vertex $u \in S_t \setminus (S_{t-1} \cup M)$ and for each $v \in N(u) \setminus (S_t \cup M)$, v is added in S_{t+1} with probability $p_{(u,v)}$.[6] In other words, each newly infected vertex that is not a monitor tries to infect her neighbors, and the probability of a successful infection depends on the weight on the corresponding edge. Conversely, once a monitor is put on a vertex, it filters the communication towards its neighbors, thus preventing the contagion of other vertices. For each $v \in V$, we denote with $\Pr_v(S, M)$ the probability that vertex v is infected given that sources are in S and

[6] Note that this is the probability that v is infected by the single neighbor u at time step t. If v has another neighbor u' that have been infected at time step $t-1$, then the probability that v will be infected at time t would be $p_{(u,v)} + p_{(u',v)}$.

monitors are in M. We also define $\mu(S, M)$ as the expected value of vertices infected in G given that sources are in S and monitors are in M, i.e., $\mu(S, M) = \sum_{u \in V \setminus M} w(u) \cdot \Pr_u(S, M)$.

The game is structured as follows. The Attacker may spend some budget $b_a \in \mathbb{R}_+$ to deploy sources, a.k.a. seeds, of infection on the vertices, paying a cost of $c_a(u)$ to select u as a source. On the other side, the Defender, able to put monitors to contain the contagion, may choose vertex u to be a monitor, paying a cost equal to $c_d(u)$, without exceeding her budget $b_d \in \mathbb{R}_+$. Thus, pure strategies of the Defender and the Attacker are all possible sets of vertices. Specifically, we denote with $S \subseteq V$ a pure strategy of the Attacker while we employ $M \subseteq V$ to indicate a set of vertices on which the Defender has put monitors. We adopt a leader-follower paradigm, meaning that the Defender commits to a (mixed) strategy and the Attacker undertakes her action after having observed the Defender's commitment.

The goal of the Attacker is to maximize the expected value $\mu(S, M)$ of infected vertices, while the Defender must protect all critical targets while minimizing the number of infected vertices. Thus, utilities are defined as follows: for the Attacker, we have $U_a(S, M) = \mu(S, M)$, while the utility of the Defender is $U_d = -Z$, for some very large Z, if there is non-zero probability that some $t \in T$ becomes infected, and $U_d = -U_a(S, M)$ otherwise. Hence, the utility is lexicographic-like, i.e., with a primary objective, namely to protect target, and a secondary one, i.e., to reduce infected vertices.[7]

We notice that critical targets are valuable vertices only for the Defender and not for the Attacker, whose goal is to spread the infection as much as possible. We call our game *Strategic Monitor Placement against Infection (SMPI)*.

Our goal is to find a leader-follower Stackelberg equilibrium (henceforth, simply called *equilibrium*) of SMPI. In other words, we aim at answering the following question: *which is the best placement for the monitors to protect the critical targets while minimizing the spread of the infection generated by malicious sources?*

Game Characterization. Before tackling our problem, we briefly discuss some connections with [30, 34], where models with features similar to ours are presented. Nevertheless, we show that such models cannot capture and solve the problem we propose.

In [34], the authors consider an attack spreading according to the Independent Cascade Model and they allow Attacker and Defender to have different evaluations on the infected nodes. Still, there are many differences with our model: first, they assume that for every node multiple security strategies are available, and different strategies may be selected for different nodes, independently from each other; instead, we can only choose monitors, and thus the way different nodes are protected are intrinsically interdependent. Second, they assume that the Defender does not have a budget, even if each security strategy has a cost: hence, the set of security strategies available to their Defender is not limited as in our setting. Moreover, the results of [34] focus on a model that is even simpler: they assume that the Attacker may infect a single node and that the probability that a node is compromised does not depend on the security strategies of other nodes except the one infected by the Attacker. Conversely, in our model, the Attacker may place more than one source and the probability that a node is infected heavily depends on a monitor being between the seeds and the node itself.

[7] Our utility functions must not be confused with lexicographic preferences on lottery outcomes. For the latter it is known that they cannot be translated in utility functions, and hence a (mixed strategy) Nash equilibrium may not exist. In our definition, we explicitly state the utility function modeling preferences of the Defender, and hence an equilibrium always exists by Nash theorem.

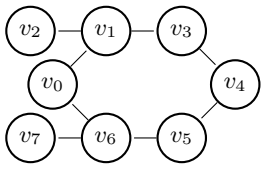

Figure 1. Network employed in Example 1.

In [30], the authors propose the following game: an Attacker chooses a set of sources from which misinformation is spread over a social network, and the Defender chooses a set of nodes from which *true information* is diffused. The Attacker' goal is to maximize the number of infected nodes, while the Defender's one is to minimize it. There are three main differences between our model and [30]: first, the diffusion model we use is the Independent Cascade Model, while the authors use a Linear Threshold Model. Then, they do not consider the presence of critical targets. Finally, and most importantly, while in [30] the Defender can directly conquer nodes of the network injecting true information, in our model we can only adopt surveillance measures, i.e., place monitors on the network to detect and block the infection. The following example shows that this difference distinguishes the two models even if we consider the same diffusion model and no critical targets.

Example 1. Consider network in Figure 1.

Let us assume that $p_e = 1$ for every edge $e \in E$, $w(u) = 1$ for every vertex u and budgets are $b_a = b_d = 1$. Costs are as follows: $c_a(v_0) = c_d(v_0) = 1$, $c_d(v_3) = c_d(v_5) = 0.5$, $c_a(v_4) = 1$, with other costs, for both Attacker and Defender, being greater than 1.

We consider that (mis)information spreads on the network according to the Independent Cascade Model, and we do not make any assumption on the tie-breaking rule used whenever a node is activated at the same time both by the Defender and by the Attacker. Specifically, our result holds for every tie-breaking rule.

Given these costs and budgets, the Defender has two possible choice for monitors, $M' = \{v_0\}$ or $M'' = \{v_3, v_5\}$ while the Attacker can put her source either in $S' = v_0$ or in $S'' = v_4$. It is easy to check that (M', S''), is an equilibrium in the model of [30]. Indeed, if the Defender put the seed in v_0, the Attacker may put hers either in v_0, with a utility of 0, or in v_4, getting a utility equal to 2, by infecting nodes v_3, v_5. Conversely, if the Defender places the seeds in M'', the Attacker may put her seed in v_4, getting 0, or in v_0, getting a utility of 4, infecting nodes v_1, v_2, v_6, v_7. Thus, the Defender will commit to v_0 and the Attacker will best respond playing v_4.

However, (M', S'') is not an equilibrium in our model. Indeed, if the Defender puts a monitor in v_0, the Attacker would choose v_4 and would infect all nodes of the network, thus getting a utility equal to 6. Conversely, if the Defender places her monitors in M'', the Attacker can put her source either in v_0 or in v_4. In the former case, she would get a utility of 4, infecting nodes v_1, v_2, v_6, v_7, while selecting v_4 she would get 0. Therefore, the best strategy for the Defender is to place monitors in M'', with the Attacker best responding by putting a source in v_0.

An interesting property about SMPIs is that they are not *unilaterally competitive* [17], as stated in the following proposition.

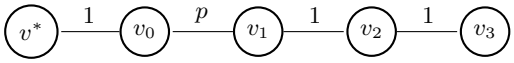

Figure 2. Network employed to prove Proposition 1.

Proposition 1. *SMPIs are not unilaterally competitive games, i.e., it is not true that whenever the utility of the Defender increases, the utility of the Attacker do not increase, too. The claim holds even if all vertices have unitary cost, both for the Attacker and the Defender, and unitary value.*

Proof. Consider the network in Figure 2, where $p < 1$.

Let $T = \{v^*\}$, $b_a = b_d = 1$. Having the Defender a unitary budget, she can place only one monitor in the network. If the monitor is in v_2, the Attacker responds by placing the source in v_0. Then, the Defender's utility is $-Z$, being the target infected with probability 1, whereas the Attacker utility is $1 + p$, since she infects v_1 and v^* with probability p and 1, respectively. Conversely, if the monitor is in v_0, then the target cannot be infected, and the cost of the Defender is the same as the Attacker's utility, i.e., 2, since v_2, v_3 are infected with probability 1 if the source is in v_1. Thus, both players increase the utility when the monitor is moved from v_2 to v_0. □

3 COMPUTATIONAL ANALYSIS

We now study our problem from a computational perspective, showing its hardness.

Theorem 1. *No polynomial-time algorithm computes an equilibrium for SMPI, unless* $\mathsf{P} = \mathsf{NP}$*, even if there are no critical targets.*

Actually, we prove a stronger result, namely that even computing the best-response of the Attacker is hard, even if the Defender has a dominant strategy. The hardness of computing an equilibrium for SMPI then follows directly: indeed, since we adopt a leader-follower paradigm, the Attacker undertakes her action after the Defender's commitment to her dominant strategy, best responding to it, and so an equilibrium for SMPI cannot be computed in polynomial time if we cannot efficiently compute the Attacker's best response in this scenario. We now formally state this result.

Lemma 1. *Finding the best response for the Attacker in SMPI is* NP*-hard, even if the Defender has a dominant strategy, can place a single monitor and there are no critical targets.*

Proof Sketch. We are given an SMPI game on a graph $G = (V, E)$, a set $M \subseteq V$ of monitors already placed by the Defender, and an integer K, and we ask whether there is a set $S \subseteq V$ of size at most b_a such that, placing the seeds on S, the Attacker infects an expected number of vertices greater than or equal to K.

Our proof extends the one in [16] and reduces this problem from the problem of finding a vertex cover in a *cubic graph* (VC^3), i.e., a graph in which every vertex has degree exactly equal to 3. In this problem, we are given a graph $G' = (V', E')$, with $|V'| = n'$ and $|E'| = m'$, s.t. for every vertex $u \in G'$ there are exactly 3 edges incident on it, and an integer $K' = \frac{m'(1+\alpha)}{3}$ for some $\alpha \in [0, 1]$, and we ask whether there is a set $T \subseteq V'$ of at most K' vertices, s.t. for every $u \in V' \setminus T$ there is at least one neighbor of u in T. It is known that for VC^3 there is $\varepsilon > 0$ s.t. it is NP-hard even distinguishing if a vertex cover of size at most K' exists or every set $T \subseteq V'$ of at most K' vertices covers a fraction at most $(1 - \varepsilon)$ of the edges [1].

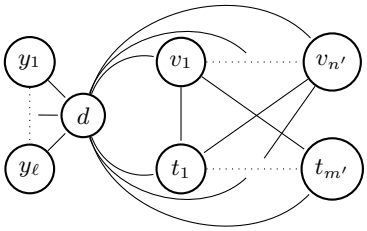

Figure 3. Network employed to prove Lemma 1.

For every instance of VC^3, we build an instance of our problem as follows. Graph G consists of the following $2\ell + 1$ vertices, where $\ell = n' + m'$; a vertex v_u is added for each vertex $u \in V'$; a vertex t_e is added for each element $e \in E'$; vertices $y_1, \ldots, y_\ell$; vertex d. Edges among these vertices are as follows: there is an edge (v_u, t_e), with $u \in V'$ and $e \in E'$ only if e is incident on u; and there is an edge between d and each other vertex in the network. The resulting graph is depicted in Fig. 3. Moreover, we assume that $p_e = p \in \left[\frac{1}{2}, 1\right)$ for every edge $e = (v_u, t_e)$, and $p_e = 1$ for every remaining edge. We set $b_a = K'$, $c_a(v_u) = 1$ and $w(v_u) = 0$ for every $u \in V$, $c_a(t_e) = K' + 1$ and $w(t_e) = 1$ for every $e \in E$, $c_a(d) = 1$ and $w(d) = 1$, $c_a(y_i) = K' + 1$ and $w(y_i) = 1$ for $i = 1, \ldots, \ell$, and $M = \{d\}$. Let us also set $\Phi_0 = p(2p^2 - p^4)(2 - p(2p^2 - p^4))$, $\Phi_1 = p + (1 - p)p(2p^2 - p^4)$, and $\Phi_2 = p + (1 - p)p$. Observe that Φ_2 is exactly the probability that a vertex t_e is infected if both neighbors are infected, Φ_1 is exactly the probability that a vertex t_e is infected if only one neighbor is infected and there is at least one neighbor infected for each vertex $t_{e'}$ with $e' \in E$; finally, Φ_0 is the probability that a vertex t_e is infected if no neighbor is infected and there is at least one neighbor infected for each vertex $t_{e'}$, with $e' \neq e$. Moreover, observe that, by our choice of p, it holds $\Phi_2 - \Phi_1 < \Phi_1 - \Phi_0$. Finally, we set $K = (1 - \alpha)m'\Phi_1 + \alpha m'\Phi_2$.

It is not hard then to check that if there is a vertex cover of size K', then there are sources from which the infection reach K vertices in expectation. On the other side, if for every set $T \subseteq V'$ of at most K' vertices there is a fraction $1 - \varepsilon_T$, with $\varepsilon_T \geq \varepsilon$, of edges e such that no vertex incident on e is in T, then there is no choice of sources such that the number of expected infected vertices is at least K.

We finally observe that M is the Defender's dominant strategy. □

The above proof can be adapted to show that there is $\varepsilon > 0$ s.t. computing a $(1 - \varepsilon)$-approximation to the Attacker's best-response is NP-hard. However, a good approximation for this problem still exists. To this aim, recall that a function f defined on sets–as the Attacker's best-response—is *monotone* if, for every A and every $x \notin A$, it holds that $f(A \cup \{x\}) \geq f(A)$. f is *submodular* if, for every $A \subset B$ and every $x \notin B$, it holds that $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$. Conversely, it is *supermodular* if $f(A \cup \{x\}) - f(A) \leq f(B \cup \{x\}) - f(B)$. If f is both monotone and submodular, then there is a greedy algorithm that, for every b, returns a set B s.t. $|B| \leq b$ and $f(B) \geq \alpha \max_{A:\, |A| \leq b} f(A)$, with $\alpha = 1 - \frac{1}{e}$ [20]. It is not hard then to see that the problem of finding the best-response of the Attacker is *monotone* and *submodular* with respect to the number of sources. Hence, despite the hardness result, one can easily compute an α-approximation.

We now turn to the Defender, proving that computing her best re-

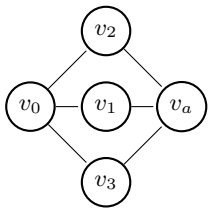

Figure 4. Network employed to prove Lemma 2. Here $p = 1$ for all edges.

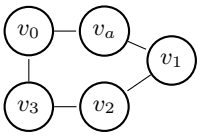

Figure 5. Network employed to prove Lemma 2. Here $p = 1$ for all edges.

sponse is hard, even if there is no critical target. Indeed, (the decision version of) this problem is equivalent to the Vaccination Problem, which has been proved to be NP-hard [8]: here we are given a graph $G' = (V', E')$, a set of infected vertices $I \subseteq V'$, a cost $c(v)$ for each $v \in V'$, a budget B, and an integer K', and we ask whether there is a set of vertices $P \subseteq V' \setminus I$ such that $\sum_{v \in P} c(p) \leq B$, and if vertices in P are removed from the graph, then the number of vertices that have a path from some $w \in I$ in G' is smaller than or equal to K'.

However, the Defender's best-response problem turns out to be even more difficult than the Attacker's one. In fact, we next show that the former is neither submodular nor supermodular with respect to the number of monitors, so that none of the approximation techniques known for these classes of functions can be exploited. Interestingly, the result holds even if the Defender has no critical targets to protect and, for all the vertices u, $c_a(u) = c_d(u) = w(u) = 1$.

Lemma 2. *The Defender's utility function is neither submodular nor supermodular, even if there are no critical targets, and all the vertices have unitary cost, for both players, and unitary value.*

Proof. Consider the network given in Figure 4.

Let v_a be the Attacker's seed. Let $A = \{v_3\}$ and $B = \{v_2, v_3\}$. The Attacker's utility when the monitor is in A is 3, since she can infect v_0, v_1, and v_2. If monitors are in $A \cup \{v_1\}$ or in $B = A \cup \{v_2\}$, the Attacker's utility is 2, since she can infect v_0 and the non-monitored vertex between v_1 and v_2. Finally, the Attacker's utility when monitor are in $B \cup \{v_1\}$ is 0, since she cannot infect any vertex. Recalling that, without critical targets, the game is zero sum, we observe that: $f(A \cup \{v_1\}) - f(A) = -2 - (-3) = 1 < 2 = f(B \cup \{v_1\}) - f(B)$.

Unfortunately, the Defender's utility function turns out to be also non supermodular. Consider the network given in Fig. 5.

As above, let v_a be the Attacker's seed and let $A = \{v_1\}$ and $B = \{v_0, v_1\}$. The Attacker's utility when the monitor is in A is 3, since she can infect v_0, v_2, and v_3. The Attacker's utility when the monitors are in $A \cup \{v_3\}$ is 1, since she can infect only v_0. The Attacker's utility when the monitors are in B or in $B \cup \{v_3\}$ is 0, since she cannot infect any vertex. Since the game is zero sum in this case, we observe that $f(A \cup \{v_3\}) - f(A) = -1 - (-3) = 2 > 0 = f(B \cup \{v_3\}) - f(B)$. □

Whereas these negative results imply that even designing a good approximation algorithm may be an hard task, still recent literature introduced several heuristics with experimentally acceptable performances [11, 37, 39, 2]. Moreover, almost all of these works show that centrality measures, e.g., variants of the well-known PageRank algorithm [23], also are experimentally a good, and often more-efficient, alternative to more complex algorithms.

We note that both the Defender's best-response hardness result and the proof of Lemma 2 assume that the source of infection (i.e., the choice of the Attacker) is known. However, it is not hard to see that these instances may be modified in order to make the Attacker's strategy dominant: e.g., in Figure 4 one may connect v_a to a large number of expensive (for both the Attacker and the Defender) nodes. Hence, our results imply hardness (and absence of submodularity) even if the Defender commits to a strategy before than the Attacker decides the sources of infection.

4 AN EXACT ALGORITHM

In this section we present some approaches able to compute an equilibrium for SMPI that scale well in practice, even if in the worst case they require exponential time.

Let us first consider the case where no critical target exists. Observe that in this case, the game is zero-sum. Several approaches have been introduced in literature for computing the leader-follower Stackelberg equilibrium of a zero-sum game. The most successful is the *double oracle* approach [12], that is based on the iterative computation of the best response of one player w.r.t. a known strategy of the other. Unfortunately, results given in previous section, prove that even computing these best-response oracles is NP-hard. Nevertheless, it would still possible to adopt approximation algorithms or heuristic as oracles, just as the ones introduced in the previous section. Experimental evidence exists that an outcome close to an equilibrium can be quickly returned even in this case (see, e.g., [30]).

Unfortunately, heuristic oracles do not work when we introduce critical targets. Indeed, it is not hard to design instances in which heuristics lead the double oracle approach to outcomes in which there is a chance for the Attacker to infect the targets, even if there is an equilibrium in which this never occurs: for example, if PageRank is adopted as Defender's oracle, then one can think of a ring and a center node connected to every ring node, infection probabilities 1 on every edge, $w(u) = c_d(u) = c_a(u) = 1$ for every u, budget equal to 1 for both the Attacker and the Defender, and the target being a ring node. It is easy to see that the oracle chooses the center node, allowing the Attacker to reach the target, regardless of the selected source, whereas it would be possible for the Defender to protect the target by placing the monitor directly on it. Similar (more complex) constructions can be given for different heuristics.

For this reason we propose a different approach that exploits the structure of the problem. We assume that $Z = \infty$, and present an approach that computes an exact equilibrium whenever an oracle returning an equilibrium for a SMPI with no critical targets is provided. It will be immediate to see that our approach still works even if one considers finite large Z, by returning an approximate equilibrium, with the approximation factor that depends on how large Z is with respect to the infection probabilities and the number of nodes in the network. We stress that very large values of Z are required by the nature of the problems considered in this work, namely to have lexicographic-like preferences.

Consider an SMPI with network $G = (V, E)$, critical targets T, Defender's and Attacker's costs $c_d(u)$ and $c_a(u)$, and budgets b_d and b_a. Consider the network $G' = (V', E')$ where critical targets $v \in T$ are substituted by a unique super-vertex t such that the edge (u, t),

for $u \in V' = V \setminus T$, exists if and only if $(u, v) \in E$ for some $v \in T$. We also write $|G'|$ to denote the size of G'. A *vertex cut* of G' is a separator, that is set $C \subseteq V$ s.t. the removal of vertices of C from G' disconnects the graph in two components (L_C, R_C), where we always identify R_C with the component containing t. The size $|C|$ of a vertex cut is the sum of the costs of vertices in C, i.e., $|C| = \sum_{u \in C} c_d(u)$.

Given a vertex cut C, the *game induced by* C is a SMPI game with network $H = (L_C \cup R_C, \mathcal{E})$, where, for every $u, v \in L_C \cup R_C$, $(u, v) \in \mathcal{E}$ if and only if $(u, v) \in E'$, no critical target, Defender's and Attacker's costs $c_d(u)$ and $c_a(u)$, and budgets $b'_d = b_d - |C|$ and b_a. In what follows, we assume that we are provided a *subgame equilibrium oracle* that, for every vertex cut C, computes the equilibrium of the game induced by C. Actually, since in these games there is no critical target for the Defender, then, as suggested above, an outcome close to the equilibrium can be easily computed through a double oracle approach. Now we can state the following theorem.

Theorem 2. *There exists an algorithm that, having access to a subgame equilibrium oracle, computes an equilibrium of the SMPI game. Moreover, if N is the number of vertex cuts in G' of size at most b_d, the algorithm runs in time $O(N \cdot \texttt{poly}(|G'|))$, and requires at most $O(N)$ accesses to the oracle.*

We say that an equilibrium *protects the Defender's critical targets* if there is zero probability that these targets are infected whenever both the Defender and the Attacker follow the equilibrium. In Lemma 3, we give a characterization of these equilibria.

Lemma 3. *An equilibrium that protects the Defender's critical targets exists if and only if there is vertex cut C in G', of size at most b_d, such that the equilibrium (M, S) of the game induced by C has $S \subseteq L_C$.*

Proof. As for the *If* direction, let $\mathcal{C}$ be the set of vertex cuts C of G' of size at most b_d such that the equilibrium (M, S) of the game induced by C has $S \subseteq L_C$. By hypothesis, $\mathcal{C}$ contains at least one element. For every $C \in \mathcal{C}$, consider the profile $(C \cup M, S)$, where (M, S) is the equilibrium of the game induced by C[8]. Then, among these profiles, there must be at least one profile $(C^\star \cup M^\star, S^\star)$ that maximizes the utility of the Defender. We claim that $(C^\star \cup M^\star, S^\star)$ is an equilibrium. Clearly, given the cut $C^\star$, neither the Defender nor the Attacker may improve her utility, since $(M^\star, S^\star)$ is an equilibrium of the game induced by $C^\star$ and $S^\star \subseteq L_{C^\star}$. Moreover, according our choice of $C^\star$, the Defender cannot improve her utility by choosing another vertex cut in $\mathcal{C}$. Finally, every $C \notin \mathcal{C}$ either is not a vertex cut of G', or it is a vertex cut but S is not a subset of L_C, and thus it is a subset of R_C. Observe that in both cases, either M is a vertex cut that, as observed above, cannot generate an equilibrium preferred by the Defender to $(C^\star \cup M^\star, S^\star)$, or the Attacker infects t with non-zero probability from every vertex in S, meaning that the Defender does not protect her critical targets, and thus her utility is lower than the one provided by $(C^\star, S_{C^\star})$.

The *Only if* direction follows by similar arguments. □

The above characterization is useful to design an algorithm to verify if there exists an equilibrium (M, S) that protects the Defender's critical targets, and, if so, to compute it.

[8] For readability, we assume that M is a pure strategy. However, our argument can be immediately applied even if the oracle outcome consists of mixed strategies.

Lemma 4. *There exists an algorithm that, given access to a subgame equilibrium oracle, verifies if there exists an equilibrium that protects the Defender's critical targets. Moreover, if N is the number of vertex cuts in G' of size at most b_d, such algorithm runs in time $O(N \cdot \texttt{poly}(|G'|))$, and requires at most $O(N)$ accesses to the oracle.*

Proof. According to Lemma 3, it is sufficient to show that there is a vertex cut C in G' of size at most b_d s.t. the equilibrium (M, S) of the game induced by C is such that S is a subset of L_C.

Computing a vertex cut in G' is equivalent to find a cut in a directed weighted graph G'' achieved from G' by substituting every vertex v in G' with vertices v_{in} and v_{out} connected by a directed edge $e = (v_{\text{in}}, v_{\text{out}})$ with weight $c_d(v)$, and every edge $e = (u, v)$ with directed edges $(u_{\text{out}}, v_{\text{in}})$ and $(v_{\text{out}}, u_{\text{in}})$ with weight $W > n \max_{v \in V} c_d(v)$. Observe that all the edge cuts of G'' of total weight less than W only involve edges $(v_{\text{in}}, v_{\text{out}})$ for some $v \in G'$, and the removal of each of these edges in G'' is equivalent to the removal of v in G'. Moreover, we can also enumerate all the vertex cuts of G' of size at most b_d. As described above, this is equivalent to enumerate all the cuts of G'' in a non-decreasing order of weights until a cut of size larger than b_d is found.

To verify if there exists an equilibrium that protects the Defender's critical targets, we can simply enumerate all the vertex cuts C of G' of size at most b_d, and check, for each of them, if the equilibrium (M, S) of the game induced by C is such that S is a subset of L_C. As discussed above, this can be done if the access to a subgame equilibrium oracle is given.

The second part of the claim holds since enumerating the cuts of G'' only requires $O(n'm' \log(n'^2/m'))$ steps between two successive cuts [36], where n' is the number of vertices in G'' (that is, $n' = 2n$) and m' the number of edges in G'' (that is, $m' = n + 2m$). □

The above algorithm also provides a way to compute an equilibrium that protects the Defender's critical targets if any.

Lemma 5. *There exists an algorithm that, given access to a subgame equilibrium oracle, computes an equilibrium that protects the Defender's critical targets, if any. Moreover, if N is the number of vertex cuts in G' of size at most b_d, such algorithm runs in time $O(N \cdot \texttt{poly}(|G'|))$, and requires at most $O(N)$ accesses to the oracle.*

Proof. According to Lemma 3, if there is a vertex cut C in G' of size at most b_d such that the equilibrium (M, S) of the game induced by C is such that S is a subset of L_C, then an equilibrium that allows the Defender to protect the critical target exists. However, this does not mean that $(C \cup M, S)$ is necessarily an equilibrium.

In fact, there may be different cuts satisfying the above property, leading to different profiles with different costs for Defender. Still, the strategy that the Defender may announce in an equilibrium that protects the Defender's target must contain a vertex cut C' of G' of size at most b_d such that the equilibrium (M', S') of the game induced by C' is such that S' is a subset of $L_{C'}$, otherwise, there is a target that is infected with non-zero probability. Hence, it is possible to find the equilibrium by enumerating all vertex-cuts C of G', and returning the pair $(C \cup M, S)$ that minimizes the Attacker's utility, when (M, S) is the equilibrium of the game induced by C.

The second part of the claim follows from the complexity of enumerating the vertex-cuts of G', discussed above. □

We can now prove Theorem 2.

Proof of Theorem 2. According to Lemma 4, there is an algorithm to verify if there exists an equilibrium that protects the defender's targets. If such an equilibrium exists, then, according to Lemma 5, there is an algorithm to compute it.

Conversely, suppose that such an equilibrium does not exist. This means that for every strategy the Defender can commit to, the best-response of the Attacker allows infecting the targets with non-zero probability. Hence, for every announced strategy of the Defender, there is no alternative action that she can commit to that gives her a lower cost. Thus, an equilibrium consists of choosing an arbitrary strategy for the Defender, and computing the corresponding Attacker best-response.

The complexity follows from Lemma 4 and 5. □

By Theorem 1, the algorithm described in Theorem 2 is not polynomial, unless $\mathsf{P} = \mathsf{NP}$, since it requires to compute subgame equilibria that is an NP-complete problem. Moreover, in the worst case, the algorithm has to enumerate all the cuts in G'' of size at most b_d. While this number is known to be small for undirected graphs if b_d is not too *large* [14], for directed graphs (as G'' is) it may happen that the number of minimum cuts is exponential in the size of the graph [31].

However, we remark that on tree topologies, this algorithm turns out to be polynomial since the best responses of both the Attacker and the Defender can be computed in polynomial time, and the number of vertex cuts is polynomial, too.

Moreover, in the next section we propose several optimizations to the algorithm presented in this section and we will show that practical performances allow it to be efficiently implemented in every network of medium size.

5 OPTIMIZATION AND EXPERIMENTS

Previous section presented an algorithm, based on the double oracle approach and using a cut enumeration procedure, to find an equilibrium of the SMPI game. As highlighted above, the algorithm is exponential in the worst case, and it may be unpractical due to the large number of cuts to be enumerated. Still, we show that through simple optimizations, the algorithm can be made practical for networks of medium size at the cost of a very small loss in the quality of the solution, defined as the Defender's utility.

First, observe that in order to compute the utility of both the Attacker and the Defender, it would be necessary to consider all paths rooted in seeds. This operation may be quite expensive on large networks. For this reason, we optimize our algorithm by filtering out paths that bring the infection with very low probability.

Whereas above optimization turns out to be very useful, it still does not address the main bottleneck of our algorithm, namely the large number of cuts that need to be enumerated. Moreover, cut enumeration algorithms do not care about the size of the generated partitions: they may for example return a cut C such that L_C is very small (e.g., contains less then b_a) nodes, and hence cause that the attacker best response would be to place at least one seed in R_C, and thus potentially infecting the target. Thus, most of the enumerated cut will be discarded. To address this issue, we apply a reverse approach: we first compute a subgame equilibrium (S, M') oracle on the full network, with Attacker's budget b_a and Defender's budget $b'_d \leq b_d$; next we place the remaining $\delta = b_d - b'_d$ Defender's monitors on a vertex cut C of size δ, chosen only among the ones in which $S \subseteq L_C$. The cut enumerating algorithm in [31] allows to implement this constraint efficiently, resulting in a very large reduction

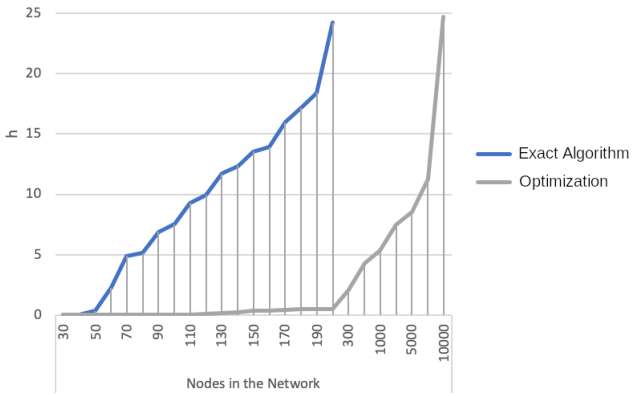

Figure 6. Performances of the exact algorithm and its optimization.

of the number of enumerated cuts. The equilibrium is then computed by repeating above procedure for multiple values of b'_d, and to choose the best one. Note that there may be exponentially many values of b'_d, that would be necessary to check, in order to consider all possible alternatives. However, we run a further optimization, by only varying b'_d by discrete steps within the interval $[0, b_d]$.

In order to evaluate the performance of this algorithm, we run it on synthetic Small World undirected graphs with increasing number of nodes and the 1% of them, selected at random, marked as critical targets. Values and costs of nodes are assigned randomly from a range $[1, \log |V|]$. Edge probabilities also are assigned uniformly at random. We set the Defender's budget to be the 2% of the total value of the network, and the Attacker's budget to be the 6%. Moreover, in order to choose the parameters suggested required by the optimizations described above, we run extensive experiments. It turns out that better results[9] are achieved when we filter out all paths whose infection probability is at most 0.07 and we will consider a step size for the enumeration of b'_d equivalent to the 0.1% of b_d.

Figure 6 reports a comparison of the size of graphs for which a solution to the problem is computable within a time limit of 25 hours[10]. It turns out that our optimizations improve over the original exact algorithm by a factor of at least 30, allowing to compute a solution to the problem even for network of 10000 nodes. This solution differs from the optimal solution of at most 10% (clearly, more precise results can be achieved by opportunely tuning parameters). Moreover, it turns out that monitors placed by the optimized algorithm never allows the attacker to infect the critical targets.

6 CONCLUSIONS

In this paper, we defined a new class of security games to study network-flows attack settings, introducing additional feasibility constraints over the Defender's strategy space. Our model could be applied in the general context of controlling flow of traffic through a (physical or computer) network to guarantee that specific vertices are not reached. For example, whenever security agencies have to

[9] Here, we mean that these parameters are the largest ones that assure the results to be close to the one achieved by the original exact algorithm.

[10] Simulations have been run on a CPU Intel Core i7 860 2.8 GHz, 4 core with 8MB cache and 4GB RAM. For each choice of the number of vertices going from 30 to 10000, the figure shows an average of 10 different runs.

primarily guarantee the safety of specific infrastructures (red zones) and then limit the area exposed to the attacks of an enemy. In these scenarios, monitors can represent squads placed in a checkpoint or alarm systems. We demonstrated that our game model is not unilaterally competitive, and we proved that computing the leader-follower equilibrium is a hard problem even if the Defender does not have any critical target to protect and then showed that the Defender's best response is neither submodular nor supermodular. To compute the leader-follower equilibrium, we designed an exact algorithm, and we showed that, by applying opportune optimizations, it has good experimental performances. In our experiments, we decided to choose critical targets at random to provide a robust test case to evaluate the performance of the proposed algorithm. However, it would be very interesting to evaluate how the behavior of the algorithm is affected from how far critical targets are from each other.

In the future, we will enrich our model considering new real-world features, e.g., a time constraint on the misinformation or a non-constant health status to each vertex, such that a non-infected vertex may change its status faster or slower than an infected one. Finally, it would be interesting to give the Defender the possibility to both inject true information in the network and place monitors on the vertices, and to find the best trade-off among these actions in terms of both protection and cost.

ACKNOWLEDGEMENTS

VA and DF were supported by "GNCS-INdAM". VA, DF, and NG were supported by the Italian MIUR PRIN 2017 Project ALGADIMAR "Algorithms, Games, and Digital Markets".

REFERENCES

[1] P. Alimonti and V. Kann, 'Some APX-completeness Results for Cubic Graphs', *THEOR COMPUT SCI*, **237**(1-2), 123–134, (2000).
[2] M. Amoruso, D. Anello, V. Auletta, and D. Ferraioli, 'Contrasting the Spread of Misinformation in Online Social Networks', in *AAMAS*, pp. 1323–1331, (2017).
[3] J. Aspnes, K. Chang, and A. Yampolskiy, 'Inoculation Strategies for Victims of Viruses and the Sum-of-squares Partition Problem', in *SODA*, pp. 43–52, (2005).
[4] M. Brown, A. Sinha, A. Schlenker, and M. Tambe, 'One Size Does Not Fit All: A Game-Theoretic Approach for Dynamically and Effectively Screening for Threats', in *AAAI*, pp. 425–431, (2016).
[5] C. Budak, D. Agrawal, and A. El Abbadi, 'Limiting the spread of misinformation in social networks', in *WWW*, pp. 665–674, (2011).
[6] V. Conitzer and T. Sandholm, 'Computing the Optimal Strategy to Commit to', in *EC*, pp. 82–90, (2006).
[7] R. Ehrenberg, 'Social Media Sway: Worries over Political Misinformation on Twitter Attract Scientists' Attention', *Science News*, **182**, 22–25, (2012).
[8] S. Eubank, A. Kumar, M. Marathe, A. Srinivasan, and N. Wang, 'Structure of Social Contact Networks and Their Impact on Epidemics', *DIMACS SER DISCRET M*, **70**, 181, (2006).
[9] European Center for Disease Prevention and Control. Communication on Immunisation - Building Trust, 2012.
[10] F. Fang, T. H. Nguyen, R. Pickles, W. Y Lam, G. R Clements, B. An, A. Singh, M. Tambe, and A. Lemieux, 'Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security', in *AAAI*, pp. 3966–3973, (2016).
[11] Habiba, Y. Yu, T. Y. Berger-Wolf, and J. Saia, 'Finding spread blockers in dynamic networks', in *ASONAM*, pp. 55–76, (2010).
[12] M. Jain, D. Korzhyk, O. Vaněk, V. Conitzer, M. Pěchouček, and M. Tambe, 'A Double Oracle Algorithm for Zero-sum Security Games on Graphs', in *AAMAS*, pp. 327–334, (2011).
[13] J. Jiang, S. Wen, S. Yu, Y. Xiang, W. Zhou, and E. Hossain, 'Identifying Propagation Sources in Networks: State-of-the-art and Comparative Studies', *IEEE COMMUN SURV TUT*, **17**(9), (2014).
[14] D. R Karger, 'Minimum Cuts in Near-linear Time', *JACM*, **47**(1), 46–76, (2000).
[15] D. Kempe, J. Kleinberg, and É. Tardos, 'Maximizing the Spread of Influence through a Social Network', in *KDD*, pp. 137–146, (2003).
[16] S. Khanna and B. Lucier, 'Influence Maximization in Undirected Networks', in *SODA*, pp. 1482–1496, (2014).
[17] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe, 'Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness', *J ARTIF INTELL RES*, (2011).
[18] C. J Kuhlman, A. Kumar, M. V Marathe, SS Ravi, and D. J Rosenkrantz, 'Finding Critical Nodes for Inhibiting Diffusion of Complex Contagions in Social Networks', in *ECML PKDD*, pp. 111–127, (2010).
[19] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, 'Finding Effectors in Social Networks', in *KDD*, pp. 1059–1068, (2010).
[20] G. L Nemhauser, L. A Wolsey, and M. L Fisher, 'An Analysis of Approximations for Maximizing Submodular Set Functions', *MATH PROGRAM*, **14**(1), 265–294, (1978).
[21] D. T Nguyen, N. P Nguyen, and M. T Thai, 'Sources of Misinformation in Online Social Networks: Who to Suspect?', in *MILCOM*, pp. 1–6, (2012).
[22] Office of The Director of National Intelligence. Background to 'Assessing Russian Activities and Intentions in Recent Elections': The Analytic Process and Cyber Incident Attribution, 2017.
[23] L. Page, S. Brin, R. Motwani, and T. Winograd, 'The PageRank Citation Ranking: Bringing Order to the Web', Technical report, Stanford InfoLab, (1999).
[24] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, 'Deployed ARMOR Protection: The Application of a Game Theoretic Model for Security at the Los Angeles International Airport', in *AAMAS*, pp. 125–132, (2008).
[25] A. Schlenker, H. Xu, M. Guirguis, C. Kiekintveld, A. Sinha, M. Tambe, S. Sonya, D. Balderas, and N. Dunstatter, 'Don't Bury your Head in Warnings: A Game-Theoretic Approach for Intelligent Allocation of Cyber-security Alerts', in *IJCAI*, (2017).
[26] K. Sharma, 'The Reality behind the Theory of Killer Game Blue Whale', *Times of India*, (2017).
[27] E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer, 'PROTECT: A Deployed Game Theoretic System to Protect the Ports of the United States', in *AAMAS*, pp. 13–20, (2012).
[28] D. Talbot, 'Preventing Misinformation from Spreading through Social Media', *MIT Technology Review*, (2013).
[29] J. Tsai, C. Kiekintveld, F. Ordonez, M. Tambe, and S. Rathi, 'IRIS - A Tool for Strategic Security Allocation in Transportation Networks', in *AAMAS*, pp. 13–20, (2009).
[30] J. Tsai, T. H. Nguyen, and M. Tambe, 'Security Games for Controlling Contagion', in *AAAI*, pp. 1464–1470, (2012).
[31] V. V Vazirani and M. Yannakakis, 'Suboptimal Cuts: Their Enumeration, Weight and Number', in *ICALP*, pp. 366–377, (1992).
[32] F. Vis, 'To Tackle the Spread of Misinformation Online We Must First Understand It', *Guardian Comment Network*, (2014).
[33] B. Von Stengel and S. Zamir, 'Leadership Games with Convex Strategy Sets', *GAME ECON BEHAV*, **69**(2), 446–457, (2010).
[34] Y. Vorobeychik and J. Letchford, 'Securing Interdependent Assets', *JAAMAS*, **29**(2), 305–333, (2015).
[35] L. Wu, F. Morstatter, X. Hu, and H. Liu, 'Mining Misinformation in Social Media', *Big Data in Complex and Social Networks*, 123–152, (2016).
[36] L. Yeh, B. Wang, and H. Su, 'Efficient Algorithms for the Problems of Enumerating Cuts by Non-decreasing Weights', *ALGORITHMICA*, **56**(3), 297–312, (2010).
[37] H. Zhang, M. A. Alim, X. Li, M. T Thai, and H. T Nguyen, 'Misinformation in Online Social Networks: Detect Them All with a Limited Budget', *ACM TOIS*, **34**(3), 18, (2016).
[38] H. Zhang, M. A. Alim, M. T Thai, and H. T Nguyen, 'Monitor Placement to Timely Detect Misinformation in Online Social Networks', in *ICC*, pp. 1152–1157, (2015).
[39] H. Zhang, A. Kuhnle, H. Zhang, and M. T Thai, 'Detecting Misinformation in Online Social Networks before it is too Late', in *ASONAM*, pp. 541–548, (2016).