SAND2016-4810C

Network Security Games with Probabilistic Evasion

Paper XXX

Abstract

Stackelberg or defender-attacker games have recently become one of the main tools used in modeling security decisions in adversarial settings. However, the adversarial nature of these interaction leads to a great deal of uncertainty in many areas, such as the eventual outcome when particular actions are taken or the exact utilities of these outcomes for the two sides. We propose a model of attack interdiction in network settings that takes into account outcome uncertainty for the attacker and give a double oracle formulation for solving games in this setting. Additionally, we show how to extend this model and algorithmic approach to a robust setting where there is uncertainty in attacker probabilities or attacker utilities. Finally, we evalute these models experimentally, focusing on the effect of problem paramters on runtime and the potential loss in defender efficiency if these uncertainties are ignored.

1 Introduction

In recent years, game theoretic techniques have been used to model adaptive intelligent adversaries in a wide range of critical infrastructure settings, from airports [5] to ports [3]. Network security games (NSG) focus on a subset of these problems, namely those that are most naturally modeled as a graphical network, such as placing check points on road networks [4] or choosing patrol routes in the waterways near our major cities [11]. One important element of NGSs is the asymmetry between the attacker and the defender. It is commonly assumed that the defender must *commit* to a defense strategy that the adversary can observe to some extent before launching an attack. This can be modeled by allowing the defender to commit to a mixed strategy, with the adversary only observing the overall strategy of the defender, but not the specific strategies chosen on the day of the attack. This allows the defender to calculate how the attacker might optimally respond to any potential defense configuration, and thus, the optimal defense strategies generated by these algorithms are robust to worst case choices by the attacker.

For computational reasons, much of the work in this area [6; 4] has primarily focused on settings where interdiction success is binary; once an edge or a target is defended, if the attacker then chooses a strategy that uses this edge or target they are guaranteed to be interdicted. However, an important aspect of interdiction in real world settings is the uncertainty in the success of the interdiction.

In this work we extend this line of research by allowing for defense actions that only provide probabilistic guarantees on their ability to prevent attacks. We propose a model where the attacker has a set of possible states (nodes), connected by a set of possible actions (arcs or edges) that allow them to transition between these states. Each of these actions has a baseline success rate, where we assume that if any action fails, the defender is alerted to the presence of the attacker and the entire attack fails. Furthermore, we assume that the defender has the ability to impose additional security with some limited budget. These additional security choices impact the attacker by reducing the success probability of particular actions.

2 Problem Description

The base network security game model and domain follows Tsai et. al [10] and Jain et. al [4]. The directed network is defined on a graph $G(\mathcal{N}, \mathcal{A})$. Although, the model presented here is based on a directed graph, extensions to undirected and bi-directed variants are straight forward.

A pure strategy of the defender is defined by a defense allocation vector $\boldsymbol{x}^i \in \{0,1\}^{|\mathcal{A}|}$, where each arc x^i_{uv} can be defended at a cost of c_{uv}. A total defense allocation budget of Γ_b is enforced. A pure strategy for the attacker is a path defined by a vector $\boldsymbol{y}^j \in \{0,1\}^{|\mathcal{A}|}$ that starts at a source node and terminates at a target node, where the attacker's utility is given by the product of the payoff of the target node $t(j)$ and the probability of successfully traversing the entire path prescribed by $\boldsymbol{y}^j$. For each arc in the path, we assume evasion probabilities are independent. Thus, the attacker's utility $\mathcal{U}(\boldsymbol{x}^i, \boldsymbol{y}^j)$ is a function of the independent arc evasion probabilities, which depends on the defender's allocation $\boldsymbol{x}^i$, and the payoff of the target node $t(j)$. As an example, if arc (u, v) is in attack path $\boldsymbol{y}^j$, that is to say $y^j_{uv} = 1$, the attacker's evasion probability is given by p_{uv} if the arc is undefended and given by p'_{uv} if the arc is defended, with $p_{uv} \geq p'_{uv}$.

2.1 Minimax Formulation

Given the sets of defender's allocations and attacker's paths $\mathcal{X}$ and $\mathcal{Y}$, respectively; the optimal mixed strategies for both

can be found by solving the following linear program (LP), which we refer to as the minimax (MM) formulation.

$$\max_{\mathcal{U}^*, \boldsymbol{d}} \quad \mathcal{U}^* \tag{1a}$$

$$\text{s.t.} \quad \mathcal{U}^* \leq -\mathcal{U}(\boldsymbol{d}, \boldsymbol{y}^j) \quad \forall j = 1, \cdots, m \tag{1b}$$

$$\mathbf{1}^\top \boldsymbol{d} = 1 \tag{1c}$$

$$\boldsymbol{d} \in [0,1]^{|\mathcal{A}|} \tag{1d}$$

The optimal utility of the defender is given by $\mathcal{U}^*$ and $\boldsymbol{d}$ represents the defender's mixed strategy over allocations $\mathcal{X}$. The utility function $\mathcal{U}(\boldsymbol{d}, \boldsymbol{y}^j)$ defined by (2) is the weighted sum of the utility prescribed by playing defender's mixed strategy $\boldsymbol{d}$ against the attacker's pure strategy $\boldsymbol{y}^j$.

$$\mathcal{U}(\boldsymbol{d}, \boldsymbol{y}^j) = \sum_{i=1}^{n} \mathcal{U}(\boldsymbol{x}^i, \boldsymbol{y}^j) d_i \tag{2}$$

where $\mathcal{U}(\boldsymbol{x}^i, \boldsymbol{y}^j)$ is the utility function evaluated by playing the defender's pure strategy $\boldsymbol{x}^i$ against attacker's pure strategy $\boldsymbol{y}^j$.

$$\mathcal{U}(\boldsymbol{x}^i, \boldsymbol{y}^j) = \omega(t(j)) \cdot \Big[\prod_{(u,v) \in \mathcal{A} | y^j_{uv} = 1} \max \Big\{ p_{uv}(1 - x^i_{uv}), p'_{uv} \Big\} \Big] \tag{3}$$

In (3), $\omega(t(j))$ is the payoff associated with the target $t(j)$ at the end of path $\boldsymbol{y}^j$. Since success probabilities are independent, the probability of successfully traversing the path is the product of the independent evasion probability of each arc in $\boldsymbol{y}^j$. For each arc (u,v) in the path $\boldsymbol{y}^j$, $\max\{p_{uv}(1 - x_{uv}), p'_{uv}\}$ provides the success probability of traversing arc (u,v) given defender allocation $\boldsymbol{x}^i$. If the arc (u,v) is defended, that is $x_{uv} = 1$, then $p_{uv}(1 - x_{uv}) = 0$ and the probability of successfully traversing that arc is given by p'_{uv}. On the other hand, if the arc is undefended $x_{uv} = 0$, then the success probability is p_{uv}.

2.2 Defender Oracle (DO)

Given the attacker's mixed strategy $\boldsymbol{a}$ over $\mathcal{Y}$ as input, the defender computes the best pure strategy defense allocation $\boldsymbol{x}$ by solving the following bilevel program.

$$\min_{\boldsymbol{x} \in \{0,1\}^{|\mathcal{A}|}} \max_{\boldsymbol{f} \geq \mathbf{0}, \boldsymbol{f'} \geq \mathbf{0}} \sum_{j=1}^{m} a_j \Big(\sum_{(u,t) \in \mathcal{A}} \omega_u f^j_{ut} \Big) \tag{4a}$$

$$\text{s.t.} \sum_{(u,v) \in \mathcal{A}} (f^j_{uv} + f'^j_{uv}) - \sum_{(v,u) \in \mathcal{A}} (p_{vu} f^j_{vu} + p'_{vu} f'^j_{vu}) = b_u \; \forall u \in \mathcal{N} \setminus t, \forall j \tag{4b}$$

$$f^j_{uv} + f'^j_{uv} \leq y^j_{uv} \quad \forall (u,v) \in \mathcal{A}, \forall j \tag{4c}$$

$$f^j_{uv} \leq 1 - x_{uv} \quad \forall (u,v) \in \mathcal{A}, \forall j \tag{4d}$$

$$\sum_{(u,v) \in \mathcal{A}} c_{uv} x_{uv} \leq \Gamma_b \tag{4e}$$

The defender's objective (4a) is to minimize the (maximum) utility of the attacker, which is the sum of the attacker's utility over the attack paths in $\mathcal{Y}$ weighted by the mixed strategy prescribed by $\boldsymbol{a}$. Constraints (4b) are *probabilitically* adjusted nodal balanced constraints. Observe that the amount of inflow on arc (v,u) is multiplied (adjusted) by the probability that the attacker successfully traverses arc p_{vu} (or p'_{vu}). Constraints (4c) stipulate that non-zero flows can only occur on arcs that are in the attacher's path. Constraints (4d) are flow shut-off constraints. If (u,v) is defended, then $f^j_{uv} = 0$ and the only option is to flow through the "defended" arc f'^j_{uv} with lower evasion probability. Finally, (4e) is the constraint on the overall mitigation budget.

Theorem 2.1 *Bilevel program (4) has an equivalent mixed-integer linear programming reformulation.*

Given a fixed upper-level decision $\boldsymbol{x}$, the lower-level problem of (4) is a linear program. We can thus replace the maximization problem with an equivalent dual minimization formulation. This results in a bilinear program, where the objective function contains bilinear terms, which can be subsequently linearized using a set of non-negative variables and disjunctive constraints.

2.3 Attacker Oracle (AO)

Given the defender's mixed strategy $\boldsymbol{d}$ over $\mathcal{X}$ as input, the attacker computes the best pure strategy attack path $\boldsymbol{y}$ by solving the following mixed-integer linear program (MILP).

$$\max_{\boldsymbol{y} \in \{0,1\}^{|\mathcal{A}|}, \boldsymbol{f} \geq \mathbf{0}, \boldsymbol{f'} \geq \mathbf{0}} \sum_{i=1}^{n} d_i \Big(\sum_{(u,t) \in \mathcal{A}} \omega_u f^i_{ut} \Big) \tag{5a}$$

$$\text{s.t.} \sum_{(u,v) \in \mathcal{A}} y_{uv} - \sum_{(v,u) \in \mathcal{A}} y_{vu} = b_u \; \forall u \in \mathcal{N} \setminus t \tag{5b}$$

$$\sum_{(u,v) \in \mathcal{A}} (f^i_{uv} + f'^i_{uv}) - \sum_{(v,u) \in \mathcal{A}} (p_{vu} f^i_{vu} + p'_{vu} f'^i_{vu}) = b_u \; \forall u \in \mathcal{N} \setminus t, \forall i \tag{5c}$$

$$f^i_{uv} + f'^i_{uv} \leq y_{uv} \quad \forall (u,v) \in \mathcal{A}, \forall i \tag{5d}$$

$$f^i_{uv} \leq 1 - x^i_{uv} \quad \forall (u,v) \in \mathcal{A}, \forall i \tag{5e}$$

The objective (5a) is to maximize the attacker's utility, which is the sum of the utility of the chosen path $\boldsymbol{y}$ weighted against the defender's mixed strategy $\boldsymbol{d}$. Constraints (5b) are standard nodal balance constraints for *path selection*. (5c) are probabilistically adjusted nodal balanced constraints for each node and each defender allocation $\boldsymbol{x}^i$. Constraints (5d) and (5e) are constraints restricting flows to arcs in the path and not "shut-off", respectively.

2.4 Solution Approach

We initialize the nominal (NOM) algorithm with a single pure strategy corresponding to no-defenses (i.e. $\mathcal{X} = \{\mathbf{0}\}$) and $\mathcal{Y} = \emptyset$. The algorithm starts by solving the attacker oracle (AO) against pure strategy $\boldsymbol{x} = \mathbf{0}$ to find the optimal attack path given no defenses (step 1). The optimal attack path generated will then be used to initialize $\mathcal{Y}$ (step 2). Then given $\mathcal{X}$ and $\mathcal{Y}$, the following steps are iterated until convergence is achieved. The minimax formulation (1) is solved to find the optimal mixed strategy for both players based on current sets $\mathcal{X}$ and $\mathcal{Y}$ (step 4). Next, the defender's best response is computed by solving the MILP reformulation of (4) (steps 5-6) and finally, the best response of the attacker is computed

by solving (5) (steps 6-7). The algorithm terminates if in a given iteration the optimal solutions $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ are already in $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Algorithm 1: NOM

Input: Initial $\mathcal{X} = \{\mathbf{0}\}, \mathcal{Y} = \emptyset, \boldsymbol{d} \leftarrow 1$

1 $\mathcal{Y}^* \leftarrow \text{AO}(\boldsymbol{d})$;
2 $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{\mathcal{Y}^*\}$;
3 **repeat**
4 $\quad (\boldsymbol{d}, \boldsymbol{a}) \leftarrow \text{MM}(\mathcal{X}, \mathcal{Y})$;
5 $\quad \mathcal{X}^* \leftarrow \text{DO}(\boldsymbol{a})$;
6 $\quad \mathcal{X} \leftarrow \mathcal{X} \cup \{\mathcal{X}^*\}$;
7 $\quad \mathcal{Y}^* \leftarrow \text{AO}(\boldsymbol{d})$;
8 $\quad \mathcal{Y} \leftarrow \mathcal{Y} \cup \{\mathcal{Y}^*\}$;
9 **until** *convergence*;
10 **return** $(\boldsymbol{d}, \boldsymbol{a})$

3 Uncertain Evasion Probabilities

Practically speaking, the probability that an attacker successfully traverses an arc and the effectiveness of mitigation options are not known with certainty. As examples, the effectiveness of sensors may be hindered by nuisance sources and background noise and the effectiveness of roadblocks may depend on adversary types and search durations. In this section, we consider a robust variant of NSG that accounts for uncertainties in arc evasion probabilities. In the context of NSGs, a robust optimization approach that implicitly accounts for worst-case outcomes is appealing, since such risk-averse strategies are consistent with how critical infrastructure and network security decision makers operate. The robust optimization framework presented here provides a natural way to model uncertainties associated with attacker and defender interactions and allows for control of the conservativeness of the solution.

Our robust optimization framework is built upon the work of Soyster [9] and Bertsimas and Sim [1], where we seek an optimized mixed-strategy for the defender that performs well agains all uncertainty realizations in a given uncertainty set. [9] proposed an LP model that ensures feasibility of the solution against data uncertainty in a convex set. However, a drawback of [9] is that such solutions may be too conservative; especially, in settings where uncertainties are assumed to be independent, since the probability that all uncertain parameters take on worst-case outcomes is not only highly improbable but may also be prohibitively expensive to defend. This is certainly the case in the context of NSGs. Subsequently, [1] proposed an LP approach where the uncertain parameters fall within intervals compose of a nominal (mean) value and a deviation. Conservativeness of the solution is then controlled by a budget of uncertainty Γ_u, which constraints the number of uncertain parameters that can deviate from nominal values.

Our proposed robust NSG framework is a variant of the robust optimization framework presented in [1]. In our RO model, the evasion probabilities for each arc $(u, v) \in \mathcal{A}$, depends on (1) whether the arc is defended or not ($x_{uv} \in \{0, 1\}$) and (2) whether the arc takes on the nominal or worst-case values ($z_{uv} \in \{0, 1\}$). Thus, the adversary's probability of successfully traversing an arc $(u, v) \in \mathcal{A}$ is as follows.

Table 1: Evasion probabilities over arc $(u, v) \in \mathcal{A}$ prescribed by x_{uv} and z_{uv}.

$x_{uv} \backslash z_{uv}$	0	1
0	p_{uv}	$\hat{p}_{uv}$
1	p'_{uv}	$\hat{p}'_{uv}$

We model this uncertainty by multiple replications of the original arc. For each arc (u, v), we create two arcs for the robust counterpart: one pertaining to robust flows when the arc is undefended $\hat{p}_{uv}$ and the other corresponding to robust flows when the arc is defended $\hat{p}'_{uv}$. Similar to nominal arc flows, if the robust flows are defended then the undefended robust flows are restricted to zero (in a sense shut off). If the robust arc is undefended, objective pressure will ensure that the undefended arc with higher evasion probability is used. Thus for each original arc in $(u, v) \in \mathcal{A}$, we create four arcs to capture all evasion probabilities under the different combinations of x_{uv} and z_{uv}.

The number of arcs for which the evasion probabilities may deviate from nominal values is controlled by a budget of uncertainty Γ_u. If $\Gamma_u = 0$, all evasion probabilities take on nominal values and we essentially get back the deterministic NSAG described in Section 2. At the other extreme, if $\Gamma = |\mathcal{A}|$, then all evasion probabilities take on extreme values, which may result in extremely conservative and prohibitively expensive defense allocations. Thus, from a risk informed security perspective, the region of interest lies in looking at security and cost tradeoffs when $0 < \Gamma_u \ll |\mathcal{A}|$.

3.1 Robust Minimiax Formulation

The optimal mixed strategies for the defender and attacker can be found by solving (1) using updated utilities $\mathcal{U}(\boldsymbol{x}^i, \boldsymbol{y}^j)$. Among the arcs in attack path $\boldsymbol{y}^j$, the optimal subset of Γ_u arcs to select to take on worst-case values can be found by a simple ratio test defined as follows.

$$\phi_{uv} = \begin{cases} \dfrac{\hat{p}'_{uv}}{p'_{uv}} & \text{if } x^i_{uv} = 1 \\ \dfrac{\hat{p}_{uv}}{p_{uv}} & \text{if } x^i_{uv} = 0 \end{cases} \tag{6}$$

For each arc (u, v) in the attack path $\boldsymbol{y}^j$, if $x^i_{uv} = 1$ we compute the ratio between the robust defended probability $\hat{p}'_{uv}$ and the nominal defended probability p'_{uv}, else we compute the analogous ratio for the undefended case. The optimal Γ_u arcs to select can be determined by sorting $\boldsymbol{\phi}$ in descending order and then selecting the arcs associated with the first Γ_u elements in the sorted list. Then $z_{uv} = 1$ if arc (u, v) is among the first Γ_u elements in the sorted list and $z_{uv} = 0$ otherwise. The utility function $\mathcal{U}(\boldsymbol{x}^i, \boldsymbol{y}^j)$ is thus defined as

follows.

$$\mathcal{U}(\boldsymbol{x}^i, \boldsymbol{y}^j) = \omega(t(j)) \cdot \Bigg[\prod_{(u,v)\in\mathcal{A}|y^j_{uv}=1} \max\Big\{p_{uv}(1-x^i_{uv}), p'_{uv}, \hat{p}_{uv}(1-x^i_{uv})z_{uv}, \hat{p}'_{uv}z_{uv}\Big\}\Bigg] \tag{7}$$

Given arc (u,v), if $z_{uv} = 0$ the max term in (7) simplifies to $\max\{p_{uv}(1-x^i_{uv}), p'_{uv}\}$, otherwise $z_{uv} = 1$ and the max term in (7) simplifies to $\max\{\hat{p}_{uv}(1-x^i_{uv})z_{uv}, \hat{p}'_{uv}z_{uv}\}$ since $p_{uv} \le \hat{p}_{uv}$ and $p'_{uv} \le \hat{p}'_{uv}$. Finally, we note that the robust minimax (RMM) problem retains the same computational complexity as the nominal minimax problem.

3.2 Robust Defender Oracle

We first present the *bilevel integer programing* (BIP) formulation for the robust defender's oracle. In this model, the evasion probability over an arc can either take on the nominal value or a worst-case value, subject to a budget of uncertainty Γ_u. The BIP formulation for the robust defender oracle (RDO) is given as follows.

$$\min_{\boldsymbol{x}\in\{0,1\}^{|\mathcal{A}|}} \max_{\substack{\boldsymbol{f}\ge\boldsymbol{0},\boldsymbol{f'}\ge\boldsymbol{0},\\ \boldsymbol{g}\ge\boldsymbol{0},\boldsymbol{g'}\ge\boldsymbol{0},\boldsymbol{z}\in\{0,1\}^{|\mathcal{A}|}}} \sum_{j=1}^{m} a_j \Big(\sum_{(u,t)\in\mathcal{A}} \omega_u f^j_{ut} \Big) \tag{8a}$$

$$\text{s.t.} \sum_{(u,v)\in\mathcal{A}} (f^j_{uv} + f'^j_{uv}) - \sum_{(v,u)\in\mathcal{A}} (p_{vu} f^j_{vu} + p'_{vu} f'^j_{vu}) \tag{8b}$$
$$\sum_{(u,v)\in\mathcal{A}} (g^j_{uv} + g'^j_{uv}) - \sum_{(v,u)\in\mathcal{A}} (\hat{p}_{vu} g^j_{vu} + \hat{p}'_{vu} g'^j_{vu}) = b_u \;\forall u \in \mathcal{N}\setminus t, \forall j$$

$$f^j_{uv} + g^j_{uv} \le 1 - x_{uv} \quad \forall (u,v)\in\mathcal{A}, \forall j \tag{8c}$$

$$f^j_{uv} + f'^j_{uv} + g^j_{uv} + g'^j_{uv} \le y^j_{uv} \quad \forall (u,v)\in\mathcal{A}, \forall j \tag{8d}$$

$$g^j_{uv} + g'^j_{uv} \le z^j_{uv} \quad \forall (u,v)\in\mathcal{A}, \forall j \tag{8e}$$

$$\sum_{(u,v)\in\mathcal{A}} c_{uv} x_{uv} \le \Gamma_b \tag{8f}$$

$$\sum_{(u,v)\in\mathcal{A}} z_{uv} \le \Gamma_u \tag{8g}$$

For each arc $(u,v) \in \mathcal{A}$, there are *four* flow choices depending on whether the arc is defended ($x_{uv} = 1$) or not ($x_{uv} = 0$) and whether the evasion probability takes on the worst-case value ($z_{uv} = 1$) or the nominal value ($z_{uv} = 0$). Constraints (8b) are probability adjusted nodal balance constraints with the addition of the new robust flow variables $\boldsymbol{g}$ (undefended) and $\boldsymbol{g'}$ (defended). Constraints (8c) are flow shut-off constraints. Constraints (14e) restrict flows to only arcs that are in the attack path. Constraints (14f) permit flows on a robust arc only if the arc is selected to take on worst-case probabilities, that is $z^j_{uv} = 1$. Finally, (8f) and (14g) enforce budgets on mitigation selection and uncertainty, respectively.

BIPs, with integer variables in both levels, such as RDO (8), are among the most computationally challenging optimization problems, since enumerative formulations are exponentially large, removing integrality requirements do not necessarily provide a valid relaxation, and standard branch-and-bound fathoming rules cannot be applied fully, see [2] and [8]. In order to overcome these challenges, [7], [2], and [12] proposed implicit enumeration schemes, heuristics, and various decomposition algorithms, attempting to struck a tradeoff between solution quality and runtime. However, the computational tractability of BIPs are still limited, as only small- and moderate-scale instances are solvable under practical runtime limitations.

In the following, we present a new cutting plane algorithm for solving RDO, exploiting the simple path structure of the lower-level problem. Our approach relies on first reformulating (8) into an equivalent, but exponentially large, two-stage stochastic program with a convex second-stage problem. We then employ a cutting plane algorithm, built upon Benders decomposition and an MILP separation oracle.

Let the vector $\boldsymbol{\rho} \ge \boldsymbol{0}$ trictly bound the dual variables associated with constraints (8c) over all possible defense allocation $\boldsymbol{x}$. Since the attacker always has the option to traverse "defended" arcs with lower evasion probabilities, flow feasibility is always ensured. The goal is thus to penalized the attacker for the use of the unavailable arcs. If the penalty is chosen to be sufficiently large, then it is uneconomical for the attacker to used these unavailable arcs in lieu of the "defended" arcs. Thus, (8) is equivalent to the following.

$$\min_{\boldsymbol{x}\in\{0,1\}^{|\mathcal{A}|}} \max_{\substack{\boldsymbol{f}\ge\boldsymbol{0},\boldsymbol{f'}\ge\boldsymbol{0},\\ \boldsymbol{g}\ge\boldsymbol{0},\boldsymbol{g'}\ge\boldsymbol{0},\boldsymbol{z}\in\{0,1\}^{|\mathcal{A}|}}} \sum_{j=1}^{m} a_j \Big(\sum_{(u,t)\in\mathcal{A}\setminus(\cdot,t)} \omega_u f^j_{ut} - \sum_{(u,v)\in\mathcal{A}} x_{uv}\rho_{uv} f^j_{uv} \Big) \tag{9a}$$

$$\text{s.t. Constraints (8b) and (14e)} - \text{(14g)} \tag{9b}$$

$$f^j_{uv} + g^j_{uv} \le 1 \quad \forall (u,v)\in\mathcal{A}, \forall j \tag{9c}$$

The differences between (8) and (9) are the additional penalty terms $x_{uv}\rho_{uv}f^j_{uv}$ in the objective (9a) and the removal of the x_{uv} variables in (9c). With this reformulation, the feasible region of the lower-level problem is invariant to the upper-level decision $\boldsymbol{x}$, thus given $\boldsymbol{x}$ the optimal lower-level solution is an extreme point of polytope defined by the lower-level constraints and the choice of $\boldsymbol{z}$. We now employ an enumeration scheme to remove the integer variables $\boldsymbol{z}$ in the lower-level. Let the set of all valid robust arc deviations be defined as follows.

$$\mathcal{K} = \{\boldsymbol{z} \text{ binary} \mid \boldsymbol{1}^\top \boldsymbol{z} \le \Gamma_u\} \tag{10}$$

Treating elements of $\mathcal{K}$ as a scenario, (9) can be stated as a two-stage stochastic program where each scenario is defined by $\boldsymbol{z} \in \mathcal{K}$. Thus, (9) is equivalent to the following MILP.

$$\min_{\boldsymbol{x}\in X} \sum_{j=1}^{m} a_j \alpha_j \tag{11}$$
$$\text{s.t. } \mathcal{R}(\boldsymbol{x}, \boldsymbol{y}^j, \boldsymbol{z}^k) \le \alpha_j \;\forall j = 1,\cdots,m, \forall k = 1,\cdots,|\mathcal{K}|$$

where $\mathcal{R}(\boldsymbol{x}, \boldsymbol{y}^j, \boldsymbol{z}^k)$ is the lower-level maximization problem of (9) parameterized by $\boldsymbol{x}, \boldsymbol{y}^j$ and $\boldsymbol{z}^k$.

Although, now a single-level problem, (11) is an extremely large-scale MILP, with a set of constraints for each $(\boldsymbol{y}^j, \boldsymbol{z}^k)$ pair. We next describe a cutting plane algorithm for solving (11) without the need to explicitly consider each robust arc deviations $\boldsymbol{z} \in \mathcal{K}$.

Let the master problem (MP) be defined as follows

$$\min_{\boldsymbol{x}\in\{0,1\}^{|\mathcal{A}|},\boldsymbol{\alpha}\geq\mathbf{0}} \sum_{j=1}^{m} a_j \alpha_j \tag{12a}$$

$$\text{s.t.} \sum_{(u,v)\in\mathcal{A}} c_{uv} x_{uv} \leq \Gamma_b \tag{12b}$$

and the separation oracle $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}^j)$ be defined as follows.

$$\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}^j) = \max_{\boldsymbol{f},\boldsymbol{f'},\boldsymbol{g},\boldsymbol{g'},\boldsymbol{z}} \sum_{(u,t)\in\mathcal{A}} \omega_u f_{ut} - \sum_{(u,v)\in\mathcal{A}} x_{uv}\rho_{uv} f_{uv} \tag{13a}$$

$$\text{s.t.} \sum_{(u,v)\in\mathcal{A}} (f_{uv} + f'_{uv}) - \sum_{(v,u)\in\mathcal{A}} (p_{vu} f_{vu} + p'_{vu} f'_{vu}) \tag{13b}$$

$$\sum_{(u,v)\in\mathcal{A}} (g_{uv} + g'_{uv}) - \sum_{(v,u)\in\mathcal{A}} (\hat{p}_{vu} g_{vu} + \hat{p}'_{vu} g'_{vu}) = b_u \; \forall u \in \mathcal{N} \setminus t$$

$$f_{uv} + g_{uv} \leq 1 \quad \forall (u,v) \in \mathcal{A} \tag{13c}$$

$$f_{uv} + f'_{uv} + g_{uv} + g'_{uv} \leq y^j_{uv} \quad \forall (u,v) \in \mathcal{A} \tag{13d}$$

$$g_{uv} + g'_{uv} \leq z_{uv} \quad \forall (u,v) \in \mathcal{A} \tag{13e}$$

$$\sum_{(u,v)\in\mathcal{A}} z_{uv} \leq \Gamma_u \tag{13f}$$

Algorithm (2) RDO is a cutting-plane algorithm using a projected formulation into the space of the $\boldsymbol{x}$ variables.

Algorithm 2: RDO$(\boldsymbol{a}, \mathcal{Y})$

Input: Initialize MP (12)
1 Solve MP $\rightarrow \boldsymbol{x}^*, \boldsymbol{\alpha}^*$;
2 $k \leftarrow 0$ ▷ num. of violated inequalities found;
3 **for** $j = 1, \cdots, m$ **do**
4 Solve $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}^j)$ (13) $\rightarrow \boldsymbol{w}^*, \boldsymbol{z}^*$;
5 **if** $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}^j) > \alpha^*_j$ **then**
6 Add $(\boldsymbol{h}^\top + \boldsymbol{x}^\top P)\boldsymbol{w}^* \leq \alpha_j$ to MP (12) ;
7 $k \leftarrow k + 1$;
8 **end**
9 **end**
10 **if** $k > 0$ **then**
11 Go to step 1;
12 **end**
13 **return** $\boldsymbol{x}^*$ ▷ optimal defender allocation against $(\boldsymbol{a}, \mathcal{Y})$

In step 4, given a fixed $\boldsymbol{x}$ and $\boldsymbol{y}^j$, the optimal solutions of (13) is used to generated violated optimality cuts $(\boldsymbol{h}^\top + \boldsymbol{x}^\top P)\boldsymbol{w}^* \leq \alpha_j$ in step 6.

Theorem 3.1 *Algorithm (2) is guaranteed to converge to the optimal solution of RDO in a finite number of iterations.*

3.3 Robust Attacker's Oracle

Given the defender's mixed strategy $\boldsymbol{d}$ over $\mathcal{X}$ as input, the attacker computes the best "robust" pure strategy attack path $\boldsymbol{y}$ by solving the following MILP.

$$\max_{\boldsymbol{f},\boldsymbol{f'},\boldsymbol{g},\boldsymbol{g'},\boldsymbol{y},\boldsymbol{z}} \sum_{i=1}^{n} d_i \Big(\sum_{(u,t)\in\mathcal{A}} \omega_u f^i_{ut} \Big) \tag{14a}$$

$$\text{s.t.} \sum_{(u,v)\in\mathcal{A}} y_{uv} - \sum_{(v,u)\in\mathcal{A}} y_{v,u} = b_u \; \forall u \in \mathcal{N} \setminus t \tag{14b}$$

$$\sum_{(u,v)\in\mathcal{A}} (f^i_{uv} + f'^j_{uv}) - \sum_{(v,u)\in\mathcal{A}} (p_{vu} f^i_{vu} + p'_{vu} f'^i_{vu}) \tag{14c}$$

$$\sum_{(u,v)\in\mathcal{A}} (g^i_{uv} + g'^j_{uv}) - \sum_{(v,u)\in\mathcal{A}} (\hat{p}_{vu} g^i_{vu} + \hat{p}'_{vu} g'^i_{vu}) = b_u \; \forall u \in \mathcal{N} \setminus t, \forall i$$

$$f^i_{uv} + g^i_{uv} \leq 1 - x_{uv} \quad \forall (u,v) \in \mathcal{A}, \forall i \tag{14d}$$

$$f^i_{uv} + f'^i_{uv} + g^i_{uv} + g'^i_{uv} \leq y^i_{uv} \quad \forall (u,v) \in \mathcal{A}, \forall i \tag{14e}$$

$$g^i_{uv} + g'^i_{uv} \leq z^i_{uv} \quad \forall (u,v) \in \mathcal{A}, \forall i \tag{14f}$$

$$\sum_{(u,v)\in\mathcal{A}} z^i_{uv} \leq \Gamma_u \quad \forall i \tag{14g}$$

The key difference between RO (5) and RAO (14) is the inclusion of continuous "robust" flow variables $\boldsymbol{g}, \boldsymbol{g}'$ and binary robust arc selection variables $\boldsymbol{z}^i$, one set for each defense allocation $i = 1, \cdots, n$.

3.4 Solution Approach for Robust NSGs

We initialize the robust optimization algorithm (ROB) with a single pure strategy corresponding to no-defenses $\mathcal{X} = \{\mathbf{0}\}$ and set $\mathcal{Y} = \emptyset$. ROB is structurally similar to NOM but with nominal MM, DO, and AO problem solves replaced by solves of their robust counterparts RMM, RDO, and RAO.

Algorithm 3: ROB

Input: Initial $\mathcal{X} = \{\boldsymbol{d} \leftarrow 1, \mathbf{0}\}, \mathcal{Y} = \emptyset$
1 $\mathcal{Y}^* \leftarrow$ RAO$(\boldsymbol{d}, \mathcal{X})$;
2 $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{\mathcal{Y}^*\}$;
3 **repeat**
4 $(\boldsymbol{d}, \boldsymbol{a}) \leftarrow$ RMM$(\mathcal{X}, \mathcal{Y})$;
5 $\mathcal{X}^* \leftarrow$ RDO$(\boldsymbol{a}, \mathcal{Y})$;
6 $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathcal{X}^*\}$;
7 $\mathcal{Y}^* \leftarrow$ RAO$(\boldsymbol{d}, \mathcal{X})$;
8 $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{\mathcal{Y}^*\}$;
9 **until** convergence;
10 **return** $(\boldsymbol{d}, \boldsymbol{a})$

The robust NSG formulation is naturally a nested problem as such ROB is a two-level algorithm. The outer level iterates between the RAO and RDO solves, with RDO solved using Benders decomposition that iteratively identifies violated *primal* optimality cuts.

Theorem 3.2 *Algorithm (3) ROB is guaranteed to converge to the optimal solution of RDO in a finite number of iterations.*

4 Computational Experiments

We now present computational results on some randomly generated instances to demonstrate the effectiveness of the proposed models and algorithms. All experiments were run with CPLEX 12.5 on a FILL IN MACHINE SPECS using a maximum of 16 threads and a max cutoff of 1 hour. We evaluated our formulations on two different graph structures, Erdős-Rényi (ER) random graphs with $p = .25$ and a directed grid-like setting where we divide the nodes into a series of bins, each node can only have edges to adjacent bins, and the source and target nodes are on opposite sides. This second structure generates graphs where all possible paths are of equal length, which allows us to avoid settings with small number of clearly optimal paths to defend. In both cases, the probability of the attacker successfully transitioning an undefended (defended) edge varies between .8 and .9 (.6 and .7) and target utilities varies between 10 and 20. All results reported are averaged from 30 runs.

4.1 Utility

First, we considered how much benefit we expect to gain by more accurately modeling uncertainty. To do this, we calculated the utility the defender expects to gain against an optimal attack against the following three defense strategies:

- NullUtility: A defender with no defense resources
- OptimalUtility: The optimal defense allocation
- BinaryUtility: The optimal defense allocation against the binary version of the problem

We calculated the fraction of the possible utility gain over the null setting that the binary setting achieved: $\frac{BinaryUtility - NullUtility}{OptimalUtility - NullUtility}$. Figure 1 shows this value for both 15 and 20 node grid-like graphs. On average, the optimal binary solution only seems to capture about half of the potential utility gain and seems to perform worse as the size of the graph increases. Additionally, while the binary solution seems to improve significantly as we increase the number of defense resources, we don't see a similar improvement in the grid-like case.

4.2 Runtime

Figures 2 (3) shows how runtime increases as we increase the number of nodes in the grid-like (ER) graph and the number of defense resources. Unsurprisingly, runtime increases with both number of nodes and number of defense resources. Perhaps more interesting is how much quickly the runtime increases in the grid-like graphs. This can intuitively be explained by the fact that there is higher variance in path quality in the ER graph (as the grid-like graph forces all paths to be of the same length). In fact, a large number of these paths will never be a best response for the adversary under any defense strategy. This reduces the number of iterations and the number of paths that needs to be added to the mini-max formulation.

Figure 4 shows runtime for the robust case on the grid-like graph. The robust formulation doesn't scale quite as well as the non-robust case, but we are still able to solve instances with up to 30 nodes and 2 defense resources in a reasonable amount of time on our more difficult graph type.

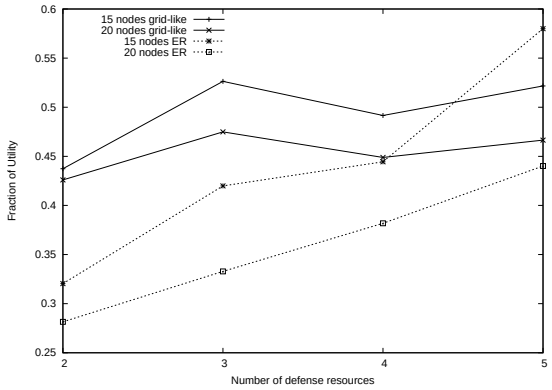

Figure 1: Utility gain (as a fraction of possible utility gain) from restricting to binary defense.

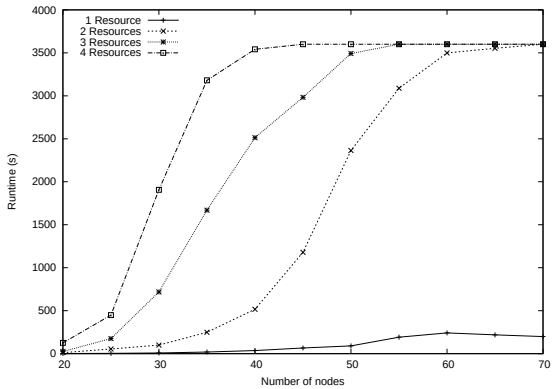

Figure 2: Average runtime of standard formulation as a function of number nodes for grid-like graphs.

5 Conclusion

In modeling security decisions in adversarial settings, it is crucial to consider uncertainties in the interaction between defender and attacker. Additionally, in many security context small perturbations to network parameters may severely impact the performance of defense allocations. This paper presents a significant advance towards addressing uncertainties in NSG. Specifically, we present novel models and algorithms for NSG with probabilistic evasion to account to the fact that in most cases the success of defense mitigation against specific attacks is not a binary outcome (fixed). In practice, the effectiveness of mitigation options can only be estimated probabilistically. We propose a model of attack interdiction that takes into account this uncertainty in outcomes, specifically considering the uncertain success probability of the attacker against randomized defense allocations. We give a double oracle formulation for solving network se-

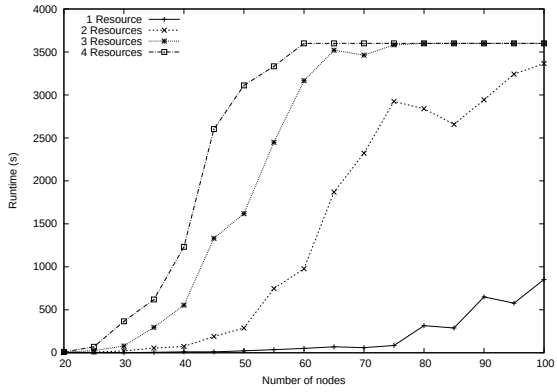

Figure 3: Average runtime of standard formulation as a function of number nodes for Erdős-Rényi graphs.

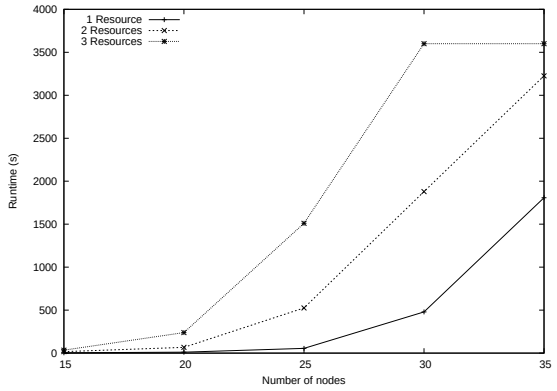

Figure 4: Average runtime of robust formulation as a function of number nodes for grid-like graphs.

curity games with probabilistic evasion. Finally, we discuss extending the model by adding additional uncertainty to the information that the defender has about the capabilities of the attacker and the effectiveness of mitigation options. Specifically, we consider the case when the evasion (or detection) probabilities are uncertain but fall within an interval of uncertainty. We then present a tractable approximation of this robust optimization model that permits us to retain the computational complexity of the original deterministic NSG problem and, simultaneously, allows for control of the conservative of the solution using a budget of uncertainty.

References

[1] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[2] S. DeNegre and T. Ralph. *A Branch-and-cut Algorithm for Integer Bilevel Linear Programs*, volume 47. Springer US, 2004.

[3] F. Fang, A. X. Jiang, and M. Tambe. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '13, pages 957–964, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[4] M. Jain, V. Conitzer, and M. Tambe. Security scheduling for real-world networks. *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2013.

[5] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordóòez. Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces*, 40(4):267–290, July 2010.

[6] J. Letchford and Y. Vorobeychik. Optimal interdiction of attack plans. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '13, pages 199–206, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[7] J. T. Moore and J. F. Bard. The mixed integer linear bilevel programming problem. *Operations Research*, 38(5):911–921, 1990.

[8] M. Scaparra and R. Church. A bilevel mixed integer program for critical infrastruture portection planning. *Computers & Operations Research*, 35:1905–1923, 2008.

[9] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.

[10] J. Tsai, Z. Yin, J. young Kwak, D. Kempe, C. Kiekintveld, and M. Tambe. Urban security: game-theoretic resource allocation in networked physical domains. *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2010.

[11] Y. Vorobeychik, B. An, and M. Tambe. Adversarial patrolling games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '12, pages 1307–1308, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[12] B. Zeng and Y. An. Solving bilevel mixed integer program by reformulations and decomposition. Technical report, University of South Florida, June 2014.