Tackling Stackelberg Network Interdiction against a Boundedly Rational Adversary

Tien Mai[1] , **Avinandan Bose**[2] , **Arunesh Sinha**[3] , **Thanh Nguyen**[4] and **Ayushman Kumar Singh**[5]
[1]Singapore Management University
[2]University of Washington
[3]Rutgers University
[4]University of Oregon
[5]Indian Institute of Technology, Delhi
atmai@smu.edu.sg, avibose@cs.washington.edu, arunesh.sinha@rutgers.edu,
thanhhng@cs.uoregon.edu, ayushman.ksingh.iitdelhi@gmail.com

Abstract

This work studies Stackelberg network interdiction games — an important class of games in which a defender first allocates (randomized) defense resources to a set of critical nodes on a graph while an adversary chooses its path to attack these nodes accordingly. We consider a *boundedly rational* adversary in which the adversary's response model is based on a *dynamic* form of classic logit-based (quantal response) discrete choice models. The resulting optimization is non-convex and additionally, involves complex terms that sum over exponentially many paths. We tackle these computational challenges by presenting new efficient algorithms with solution guarantees. First, we present a near optimal solution method based on path sampling, piece-wise linear approximation and mixed-integer linear programming (MILP) reformulation. Second, we explore a dynamic programming based method, addressing the exponentially-many-path challenge. We then show that the gradient of the non-convex objective can also be computed in polynomial time, which allows us to use a gradient-based method to solve the problem efficiently. Experiments based on instances of different sizes demonstrate the efficiency of our approaches in achieving near-optimal solutions.

1 Introduction

Network interdiction is a well-studied topic in Artificial Intelligence. There are many practical problems [Smith and Song, 2020], such as in cyber systems, that can be modeled as a network interdiction problem. In literature, many variations in models of network interdiction exist, and consequentially, a variety of techniques have been used for solving different types of these problems. Our work focuses on a particular type in which there is a set of critical nodes to protect within a larger network. We employ a popular network interdiction model [Fulkerson and Harding, 1977; Israeli and Wood, 2002], where the interdictor (defender) uses a randomized allocation of limited defense resources for the critical nodes. The adversary traverses the graphs starting from an origin and reaching a destination. There is an interaction with the defender only if the adversary crosses any critical node. The interaction is modeled using a leader-follower (Stackelberg) game where the defender first randomly allocates resources and then the adversary chooses its path accordingly.

Motivated by the fact that human adversaries in real-world security domains often act non-optimally [Tambe, 2011], we model the adversary behavior in our game setting using a dynamic Quantal Response model (an instance of well-known dynamic discrete choice (DDC) models [Rust, 1987; Aguirregabiria and Mira, 2010]). While many real world security applications have benefited from bounded rational Quantal Response model in single shot game settings [Tambe, 2011; Yang *et al.*, 2012; Fang *et al.*, 2016; Bose *et al.*, 2022], to the best of our knowledge, existing works in sequential network interdiction unrealistically assume perfectly rational adversaries and make use of the linearity to utilize linear programming techniques to tackle the problem [Smith *et al.*, 2009; Smith and Song, 2020]. We are the first to explore the DDC model of bounded rational adversaries in the network interdiction setting and formulate the defender's problem as a nonlinear optimization, leading to the requirement of solving the network interdiction problems via nonlinear optimization techniques.

While there is a closed form of the DDC adversary choice probabilities in our game setting, which is mathematically interesting in itself, the closed form presents computational challenges as the naive computation of any such probability involves reasoning about exponentially many paths from origin to destination and is a non-convex problem. This presents challenges beyond those observed in the single shot setting with quantal responding adversary [Fang *et al.*, 2016; Mai and Sinha, 2022]. Thus, we address the challenge of solving such complex non-convex optimization problem for the defender with two different novel approximation algorithms.

First, we introduce an MILP-based method, named **LiSD** (**Li**nearization via **S**ampling and **D**iscretization). The solu-

tion of **LiSD** is a bounded approximation for the interdiction problem. **LiSD** is the result of an innovative combination of path-sampling with piece-wise linear approximation (PL) techniques. Path sampling tackles the computational challenge of exponentially many paths while PL provides a near-optimal defender strategy solution with a guaranteed bound.

Second, we propose an efficient **Dyn**amic **P**rogramming method, named **DynP**. Essentially, **DynP** provides a compact and tractable formulations of the defender utility function and the optimization objective's gradient even though these terms involve summing over exponentially many paths. This is accomplished by exploiting recursive relationships among adversary utility-related terms across different paths that involves in the defender's optimal strategy computation. By employing dynamic programming, we can follow a gradient descent approach that is computationally efficient at each step to optimize the defender strategy. Furthermore, while **DynP** is computationally efficient, it does not guarantee global optimality due to the non-convexity of the defender problem. We thus identify a special case in which the adversary can visit only one (any one) critical node and show that the optimization is unimodal in that case, implying that this problem can be solved optimally in a tractable manner using gradient descent. We further identify specific conditions under which the solution to the restricted problem provides approximation guarantees for the original unrestricted one.

Notation: Boldface characters represent matrices or vectors or sets, and a_i denotes the i-th element of $\mathbf{a}$ if $\mathbf{a}$ is indexable. We use $[m]$, to denote the set $\{1, \ldots, m\}$.

2 Related Work

Dynamic discrete choice (DDC) models. From the seminal work of [Rust, 1987], DDC models have been widely studied and used to analyze sequential looking-forward choice behaviors and have various applications, e.g., on fertility and child mortality [Wolpin, 1984], on job matching and occupational choice [Miller, 1984], on bus engine replacement [Rust, 1987], and on route choice analysis [Fosgerau *et al.*, 2013; Mai *et al.*, 2015]. Among existing DDC models, the logit-based DDC has been popular due to its closed-form formulation [Rust, 1987]. This model can be viewed as a dynamic version of the well-known multinomial logit (or Quantal Response) model [McFadden, 1981; Train, 2003]. In transportation modeling, logit-based DDC was utilized to develop models to predict people's boundedly rational path-choice behavior [Fosgerau *et al.*, 2013; Mai *et al.*, 2015]. As highlighted in [Zimmermann and Frejinger, 2020], such a model presents synergies with the *stochastic shortest path* problem [Bertsekas and Tsitsiklis, 1991].

Network interdiction. Our work is a boundedly rational version of the well-studied shortest path interdiction problem [Fulkerson and Harding, 1977; Israeli and Wood, 2002]. Existing work only consider perfectly rational adversaries [Smith *et al.*, 2009; Smith and Song, 2020]. The shortest path and other network interdiction problems with perfectly rational adversaries are generally NP-hard and have strong connections with the areas of bi-level optimization [Dempe *et al.*, 2015] and robust optimization [Ben-Tal and Nemirovski, 2002]. We refer the readers to [Smith and Song, 2020] for a comprehensive review. Our work explores the DDC framework to model bounded rational adversaries, resulting in a significantly more challenging defender problem as it involves complex nonlinear optimization. Besides, there are other variant models where the problem data is not perfectly known to players [Cormican *et al.*, 1998], or where the players repeatedly make their actions [Sefair and Smith, 2016], or where online learning is involved [Borrero *et al.*, 2016].

Network security games and others. Our work also relates to static Stackelberg security game models with Quantal Response adversaries [Yang *et al.*, 2011; Yang *et al.*, 2012; Haghtalab *et al.*, 2016; Mai and Sinha, 2022; Černỳ *et al.*, 2021; Milec *et al.*, 2020; Bose *et al.*, 2023b]. In dynamic models named as network security games [Jain *et al.*, 2011], the set-up is different from our work as in this work the *rational* adversary aims to reach a target and stop, whereas in our work the *boundedly rational* adversary can attack multiple targets. Other related works along this line only consider *zero-sum* network security game setting [Xue *et al.*, 2021; Xue *et al.*, 2022]. A Quantal Response type relaxation for network security game was also studied, where the focus in on smart predict and optimize [Wang *et al.*, 2020], however, the optimize part is done using standard non-linear solver such sequential quadratic program with no guarantees.

There are other related game models where players act in a graph-based environment, including pursuit-evasion and security patrol games [Zhang *et al.*, 2019; Basilico *et al.*, 2009; Basilico *et al.*, 2017]. However, these works do not consider the attacker's bounded rationality. Additionally, their strategy spaces and problem settings are characterized differently which involve aspects of real-time information or alarm signals., etc.

3 Problem Formulation

3.1 Stackelberg Network Interdiction Games

Our network interdiction problem is a leader-follower game with a single adversary. The game is played on a network (graph) $(\mathcal{S}, \mathcal{A})$ where $\mathcal{S}$ is a set of nodes $\mathcal{S} = \{1, 2, \ldots, |\mathcal{S}|\}$, and $\mathcal{A}$ is a set of arcs. We formulate the problem as a two-player network interdiction game . The follower (adversary) takes a path through this network, which is sampled from a distribution as described below. The origin $s_o \in \mathcal{S}$ is a given starting node. In our problem, we also assume the existence of a sink (or destination) node $s_d \in \mathcal{S}$ that the adversary ultimately reaches. Let $\mathcal{L}$ be the set of critical nodes (i.e., subset of nodes in the network) that the defender can interfere or alter. From the leader's (defender's) viewpoint, the aim is to assign M resources to nodes $s \in \mathcal{L}$; each such assignment is a defender pure strategy. Further, nodes and resources are of certain types such that nodes of a given type can only be protected by resources of that same kind. Let there be K types of nodes. Let the number of resources of each type k be M_k, hence $\sum_{k\in[K]} M_k = M$. Also, let $\{\mathcal{L}_k\}_{k\in[K]}$ be a partition of the set of nodes $\mathcal{L}$ by the types of the nodes.

A mixed strategy is a randomized allocation resulting in a coverage vector $\mathbf{x} = \{x_s,\ s \in \mathcal{L}, \sum_{s\in\mathcal{L}_k} x_s \leq M_k, \forall k \in [K]\}$ where x_s is the marginal probability of covering node s, which then impacts the adversary's path choice probabilities. Given a node $s \in \mathcal{S}$, if the adversary crosses this node, then the defender gets a node-specific reward $r^l(s, x_s)$. The *Stackelberg equilibrium* can be computed by solving the following problem [Yang *et al.*, 2012; Mai and Sinha, 2022]:

$$\max_{\mathbf{x}} \quad \mathcal{F}^l(\mathbf{x}) = \sum\nolimits_{\tau\in\Omega} R^l(\tau|\mathbf{x}) P^f(\tau|\mathbf{x}) \qquad \text{(OPT)}$$

$$\text{subject to} \quad \sum\nolimits_{s\in\mathcal{L}_k} x_s \leq M_k,\ \forall k \in [K] \qquad (1)$$

$$x_s \in [L^x, U^x],\ \forall s \in \mathcal{L},$$

where $R^l(\tau|\mathbf{x}) = \sum_{s\in\mathcal{L}\cap\tau} r^l(s, x_s)$ is the defender's accumulated reward on path τ and $P^f(\tau|\mathbf{x})$ is the probability the attacker follows the path τ (of which computation is discussed in the behavior modeling part). Here, $[L^x, U^x]$ represent the required lower bound and upper bound on the coverage probability for each node in the critical set $\mathcal{L}$.

3.2 Boundedly Rational Adversary Behavior

We model the adversary's bounded rational behavior using the dynamic discrete choice framework (and specifically the logit-based recursive path choice model [Fosgerau *et al.*, 2013]). A known property in this setting is that the bounded rational adversary chooses a policy that is equivalent to a static multinomial logit (MNL) discrete choice model over all possible paths [Fosgerau *et al.*, 2013].

Concretely, let $U(\tau|s_0, \mathbf{x}) = \sum_{s\in\tau} v(s; \mathbf{x})$ be the deterministic long-term utility of the adversary when starting in s_0; if $s_0 = s_o$, then we simply write $U(\tau|\mathbf{x})$. Here, $v(s; \mathbf{x})$ is the adversary's utility associated with node s when the defender's strategy is $\mathbf{x}$. Given $\mathbf{x}$, the probability the adversary follows a path τ can be computed as follows [Fosgerau *et al.*, 2013]:

$$P^f(\tau|\mathbf{x}) = \frac{e^{U(\tau;\mathbf{x})/\mu}}{Z}, \text{ where } Z = \sum\nolimits_{\tau\in\Omega} e^{U(\tau;\mathbf{x})/\mu}, \qquad (2)$$

given Ω is the set of all possible paths and μ is the parameter which governs the follower's rationality. Thus, we can view the logit-based dynamic discrete choice formulation as a soft version of the shortest weighted path problem from the source s_o to destination s_d. Given the adversary behavior model, the adversary's expected utility can be computed as an expectation over all paths, as follows:

$$\mathcal{E}^f(\mathbf{x}) = \sum\nolimits_{\tau\in\Omega} P^f(\tau|\mathbf{x}) U(\tau; \mathbf{x})$$

Our Prop. 1 shows that the adversary's expected utility approaches the best accumulated utility (smallest path weight) as μ tends to zero (we drop the fixed strategy $\mathbf{x}$ for simplicity).

Proposition 1. *Let $\tau^* = argmax_{\tau\in\Omega} U(\tau)$ (i.e., the best path for the adversary) and $L^* = |U(\tau^*)|$. Let $\Omega^* = \{\tau;\ U(\tau) = L^*\}$ and $\alpha = U(\tau^*) - \max_{\tau\in\Omega\setminus\Omega^*} U(\tau)$. We obtain:*

$$|\mathcal{E}^f - U(\tau^*)| \leq (L^* + 1)/(1 + {|\Omega^*|}/{|\Omega\setminus\Omega^*|} e^{\alpha/\mu}).$$

As a result, $\lim_{\mu\to 0} \mathcal{E}^f = U(\tau^*)$.[1]

[1] All proofs, if not presented, are included in the appendix.

4 Common Binary Search Framework

Overall, (OPT) is computationally challenging since the objective not only involves an exponential number of paths in the network but also is non-convex. To address this computational challenge, we propose two new different algorithms which share the common underlying binary search framework. The purpose is to reduce the original fractional (OPT) to a simpler non-fractional problem. These algorithms then differ in applying different efficient techniques to solve each binary search step. We elaborate them in subsequent sections.

Essentially, we write the objective of (OPT) as follows:

$$\mathcal{F}^l(\mathbf{x}) = \frac{\sum_{\tau\in\Omega} R^l(\tau|\mathbf{x}) \exp\left({U(\tau;\mathbf{x})}/{\mu}\right)}{\sum_{\tau\in\Omega} \exp\left({U(\tau;\mathbf{x})}/{\mu}\right)}$$

$\mathcal{F}^l(\mathbf{x})$ has a fractional non-convex form. A typical way to simplify this structure is to use the Dinkelbach transform and a binary search algorithm [Dinkelbach, 1967] to convert the original problem into a sequence of simpler ones. We use binary search to write (OPT) equivalently as: $\max_\lambda \left\{\lambda \;\middle|\; \exists \mathbf{x} \text{ s.t. } \mathcal{F}^l(\mathbf{x}) \geq \lambda\right\}$ which is equivalent to finding a maximum value of $\lambda \in \mathbb{R}$ such that the following subproblem:

$$\max_{\mathbf{x}} \Big\{ \sum_{\tau\in\Omega} R^l(\tau|\mathbf{x}) \exp\big(\frac{U(\tau;\mathbf{x})}{\mu}\big) - \lambda \sum_{\tau\in\Omega} \exp\big(\frac{U(\tau;\mathbf{x})}{\mu}\big) \Big\} \quad (3)$$

has a non-negative optimal objective value. Overall, (3) is still non-convex, but no longer fractional. In addition, the set Ω of all feasible paths can be huge and may not be enumerable. Therefore, we propose two different algorithms (as elaborated next) to tackle these challenges in solving (3).

5 Linearization via Sampling and Discretizing

We describe our first near-optimal method, **LiSD**, which involves exploring path-sampling with piece-wise linear approximation (PL) techniques to approximate (3) by a MILP. Path sampling tackles the computational challenge of exponentially many paths while PL provides a near-optimal defender strategy solution with a guaranteed bound for (3).

5.1 Sample Average Approximation

We first approximate the sum over Ω via sample average approximation. That is, we select a feasible solution $\mathbf{x}_0$ to create a fixed distribution over paths in Ω. By dividing the objective of (3) by $\sum_{\tau\in\Omega} \exp({U(\tau;\mathbf{x}_0)}/{\mu})$, which is a constant, we aim to maximize the following objective function:

$$G(\mathbf{x}, \lambda) = \mathbb{E}_{\tau\sim\mathcal{D}(\mathbf{x}_0)}\big[R^l(\tau|\mathbf{x}) \exp\big(\widetilde{U}(\tau|\mathbf{x})\big) - \lambda \exp\big(\widetilde{U}(\tau|\mathbf{x})\big)\big] \quad (4)$$

where $\widetilde{U}(\tau|\mathbf{x}) = \frac{U(\tau;\mathbf{x})}{\mu} - \frac{U(\tau;\mathbf{x}_0)}{\mu}$, and $\mathcal{D}(\mathbf{x}_0)$ is the distribution over paths τ with probabilities $P^f(\tau|\mathbf{x}_0)$ (Eq. 2).

We now can approximate the objective function $g(\mathbf{x}, \lambda)$ by sample average approximation. Specifically, let $\tau_1, \ldots, \tau_N$ be N samples from $\mathcal{D}(\mathbf{x}_0)$, we approximate $g(\mathbf{x}, \lambda)$ by:

$$\widehat{G}^N(\mathbf{x}, \lambda) = \frac{1}{N} \sum_{n\in[N]} \Big[R^l(\tau_n|\mathbf{x}) e^{\widetilde{U}(\tau_n|\mathbf{x})} - \lambda e^{\widetilde{U}(\tau_n|\mathbf{x})}\Big] \qquad (5)$$

Essentially, the approximation $\widehat{G}^N(\mathbf{x},\lambda)$ converges to $G(\mathbf{x},\lambda)$ almost surely as $N\to\infty$ and the approximation errors can be bounded as shown in Proposition 2.

Proposition 2. *For any given $\xi>0$, we have:*

$$\mathbb{P}\Big(\big|\widehat{G}^N(\boldsymbol{x},\lambda)-G(\boldsymbol{x},\lambda)\big|\geq\xi\Big)\leq 2\exp\left(-\frac{2N\xi^2}{\mathcal{M}^2}\right)$$

where $\mathcal{M}=\max_{\tau,\boldsymbol{x}}\big\{J(\tau,\boldsymbol{x})\big\}-\min_{\tau,\boldsymbol{x}}\big\{J(\tau,\boldsymbol{x})\big\}$

$$J(\tau,\boldsymbol{x})=R^l(\tau|\boldsymbol{x})\exp\big(\widetilde{U}(\tau|\boldsymbol{x})\big)-\exp\big(\widetilde{U}(\tau|\boldsymbol{x})\big)$$

Proposition 2 implies that $\widehat{G}^N(\mathbf{x},\lambda)$ will converge to the true function $G(\mathbf{x},\lambda)$ in probability with an exponential rate as the number of samples N increases. This is a direct result from Hoeffding's inequality [Hoeffding, 1994].

5.2 Piece-wise Linear (PL) Approximation

We now further approximate $\widehat{G}^N(\mathbf{x},\lambda)$ by a PL function, allowing the subproblem to be solved to near-optimality via a MILP solver. First, for each τ_n, we introduce new variables $u_n=R^l(\tau_n|\mathbf{x})$ and $v_n=\frac{\widetilde{U}(\tau_n|\mathbf{x})}{\mu}$. We now can re-write the objective function, $\widehat{G}^N(\mathbf{x},\lambda)$ accordingly, as follows:

$$\widehat{G}^N(\mathbf{x},\lambda)=\frac{1}{N}\sum\nolimits_{n\in[N]}\Big(u_n\exp(v_n)-\lambda\exp(v_n)\Big)$$

Let L_n and U_n be an lower and upper bounds of v_n. The PL approximation can be done by partitioning each interval $[L_n,U_n]$ into K sub-intervals of equal size, and introducing K binary variables $z_n^1,\ldots,z_K^n$ such that $z_n^1\geq z_n^2\geq\ldots\geq z_n^K$, to represent each interval. Intuitively, $z_n^k=1$ implies the k^{th} sub-interval involves in the approximation of $\exp(v_n)$ and $z_n^k=0$, otherwise. Let $\Delta_n={}^{(U_n-L_n)}\!/_K$ (i.e., the size of each interval) and δ_n^k, $k\in[K]$ is the slop of function e^{v_n} in the interval $[L_n+\Delta_n(k-1),L_n+\Delta_n k]$:

$$\delta_n^k=\frac{\exp(L_n+\Delta_n k)-\exp(L_n+\Delta_n(k-1))}{\Delta_n}$$

Each component $\exp(v_n)$ can be approximated as follows:

$$\exp(v_n)\approx\exp(L_n)+\Delta_n\sum\nolimits_{k\in[K]}\delta_n^k z_n^k$$

We then can re-write the sub-problem (3) as follows:

$$\max_{\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{v}}\ \frac{1}{N}\sum\nolimits_{n\in[N]}(u_n-\lambda)\Big(\exp(L_n)+\Delta_n\sum\nolimits_{k\in[K]}\delta_n^k z_n^k\Big)\tag{MINLP}$$

$$\text{s.t. } z_n^k\geq z_n^{k+1};\ k\in[K-1],n\in[N]\tag{6}$$

$$u_n=R^l(\tau_n|\mathbf{x})\text{ and }v_n={}^{\widetilde{U}(\tau_n|\mathbf{x})}\!/_\mu\tag{7}$$

$$v_n=L_n+\Delta_n\sum\nolimits_{k\in[K]}z_n^k+\kappa_n\tag{8}$$

$$\mathbf{x}\in\mathcal{X},\mathbf{z}_n\in\{0,1\}^K,\kappa_n\in[0,\Delta_n]\tag{9}$$

which maximizes the piece-wise approximation of $\widehat{G}^N(\mathbf{x},\lambda)$. The additional variable κ_n captures the gap between v_n and the binary approximation $L_n+\Delta_n\sum_{k\in[K]}z_n^k$.

Finally, there are only some bi-linear terms left to be linearized in the objective function. We do that using McCormick inequalities. Specifically, let L_n^u and U_n^u be lower and upper bounds of u_n, we introduce new variables s_n^k to present $(u_n-\lambda)z_n^k$, we can now linearize the bi-linear term $(u_n-\lambda)z_n^k$ with the following additional constraints:

$$s_n^k\leq(U_n^u-\lambda)z_n^k;\ s_n^k\geq(L_n^u-\lambda)z_n^k\tag{10}$$

$$s_n^k\leq(u_n-\lambda)-(L_n^u-\lambda)(1-z_n^k)\tag{11}$$

$$s_n^k\geq(u_n-\lambda)-(U_n^u-\lambda)(1-z_n^k)\tag{12}$$

The above three constraints guarantee that when $z_n^k=1$, then $s_n^k=u_n-\lambda$. Conversely, when $z_n^k=0$, then $s_n^k=0$.

By combining the above new variable s_n^k and constraints with (MINLP), we obtain the MILP reformulation:

$$\max_{\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{v},\mathbf{s}}\ \frac{1}{N}\sum\nolimits_{n\in[N]}\Big((u_n-\lambda)e^{L_n}+\Delta_n\sum\nolimits_{k\in[K]}\delta_n^k s_n^k\Big)\tag{MILP}$$

s.t. Constraints (6–12) are satisfied.

We further establish a performance bound for PL approximation. We first remark, from the definition of $\widehat{G}^N(\mathbf{x},\lambda)$, that:

$$\begin{cases}\widehat{G}^N(\mathbf{x},\lambda)\geq 0 & \text{if }\lambda\leq\min_{n,\mathbf{x}}R^l(\tau_n|\mathbf{x})=\min_n\{L_n^u\}\\ \widehat{G}^N(\mathbf{x},\lambda)\leq 0 & \text{otherwise.}\end{cases}$$

So, it is sufficient to consider $\lambda\in[\min_n\{L_n^u\},\max_n\{U_n^u\}]$. This allows us to state Proposition 3 below.

Proposition 3. *Assume that $\lambda\in\big[\min_n\{L_n^u\},\max_n\{U_n^u\}\big]$, let $\widehat{\boldsymbol{x}}^{NK}$ be an optimal solution to* (MILP) *and $\boldsymbol{x}^*$ be optimal for average approximation sub-problem $\max_{\boldsymbol{x}}\widehat{G}^N(\boldsymbol{x},\lambda)$, then we obtain the following inequality:*

$$\Big|\widehat{G}^N(\widehat{\boldsymbol{x}}^{NK},\lambda)-\widehat{G}^N(\boldsymbol{x}^*,\lambda)\Big|\leq\frac{2BN}{K}$$

where $B=\big(\max_n\{U_n^u\}-\min_n\{L_n^u\}\big)\max_n\big\{e^{U_n}(U_n-L_n)\big\}$.

From an intuitive standpoint, augmenting K would diminish the approximation error of the PL approximation. Conversely, augmenting N has a dual effect: while it lessens the error arising from path sampling, it simultaneously heightens the cumulative error stemming from all the samples. In fact, to drive the bound closer to zero, Proposition 3 indicates that it's necessary that the rate of increase for K should surpass that of N. We further investigate this dual effect by looking at the quality of a solution returned from (MILP) w.r.t the original sub-problem $\max_{\mathbf{x}}G(\mathbf{x},\lambda)$. A performance bound is provided in Theorem 1, which implies that, under the condition $N\leq\frac{\xi K}{6B}$, the solution given by the PL approximation will converge in probability to a true optimal solution, with an exponential rate.

Theorem 1. *Assume that $\lambda\in\big[\min_n\{L_n^u\},\max_n\{U_n^u\}\big]$. Let $\widehat{\boldsymbol{x}}^{NK}$ be an optimal solution to* (MILP) *and $\boldsymbol{x}^*$ be optimal for $\max_{\boldsymbol{x}}G(\boldsymbol{x},\lambda)$, then given any $\xi>0$, if we choose N,K such that $\frac{N}{K}\leq\frac{\xi}{6B}$, then we have:*

$$\mathbb{P}(|G(\widehat{\boldsymbol{x}}^{NK},\lambda)-G(\boldsymbol{x}^*,\lambda)|\geq\xi)\leq 4e^{-\frac{2N\xi^2}{9\mathcal{M}^2}}.$$

This result can be employed to establish (theoretical) estimates for N and K to achieve a desired performance.

Corollary 1. *For any given $\alpha, \beta > 0$, $\beta \in (0,1)$, if we choose $N \geq \ln\left(\frac{4}{\beta}\right)\frac{9\mathcal{M}^2}{2\alpha^2}$ and $K \geq \frac{6NB}{\alpha}$, then $\left|G(\widehat{\boldsymbol{x}}^{NK},\lambda) - G(\boldsymbol{x}^*,\lambda)\right| \leq \alpha$ occurs with probability $1-\beta$.*

The above estimates might shed light on how N, K depends on the performance criteria α, β. We note that these estimates would be conservative, as in practice we may need much smaller N, K to achieve the desired performance. A final note is that one can employ an off-the-shelf solver (e.g. CPLEX or GUROBI) to solve (MILP). Although this program would be large in size, SOTA solvers can efficiently handle very large MILPs, aided by powerful machines.

6 Dynamic Programming Based Solution

The above MILP approximation involves binary variables and would be intractable in large scenarios. We thus propose an alternative new algorithm, ***DynP*** that also follows binary search, but at each binary step, (i) it presents a non-trivial compact representation of the objective function based on the creation of a *dynamic program,* which handles an exponential number of paths; and (ii) it applies a gradient ascent-based method to efficiently solve the resulting compact problem.

6.1 Compact Representation

We can rearrange terms in the objective of sub-problem (3) according to critical nodes as follows:

$$g(\mathbf{x},\lambda) = \sum\nolimits_{s\in\mathcal{L}} \sum\nolimits_{\substack{\tau\in\Omega\\ \tau\ni s}} r^l(s,x_s)\exp\left({}^{U(\tau;\mathbf{x})}/_{\mu}\right) - \lambda\Big[\sum\nolimits_{\tau'\in\Omega}\exp\left({}^{U(\tau';\mathbf{x})}/_{\mu}\right)\Big] \tag{13}$$

Since $g(\mathbf{x},\lambda)$ is differentiable, this maximization problem can be solved for a local maximum by a gradient-based method. One of the key challenges is the computation of $g(\mathbf{x},\lambda)$, which, if done naively, would require enumerating exponentially many paths on Ω. We next show that $g(\mathbf{x},\lambda)$ has a compact form, which allows us to compute $g(\mathbf{x},\lambda)$ and its gradient efficiently via dynamic programming.

For a compact representation of $g(\mathbf{x},\lambda)$, we introduce the following new terms for all nodes $s, s' \in \mathcal{S}$:

$$Z_s = \sum_{\tau\in\Omega^{s_d}(s)} \exp\left({}^{U(\tau;\mathbf{x})}/_{\mu}\right) \text{ and } Y^s_{s'} = \sum_{\tau\in\Omega(s',s)} \exp\left({}^{U(\tau;\mathbf{x})}/_{\mu}\right)$$

where $\Omega^{s_d}(s)$ is the set of all paths from s to the destination s_d and $\Omega(s',s)$ is the set of all paths from s' to s.

The objective $g(\mathbf{x},\lambda)$ can be now re-formulated as follows:

$$g(\mathbf{x},\lambda) = \sum\nolimits_{s\in\mathcal{L}} r^l(s,x_s)Y^s_{s_o}Z_s - \lambda Z_{s_o}, \tag{14}$$

where s_o is the origin. Although these new terms still involve exponentially many paths in $\Omega^{s_d}(s)$ and $\Omega(s',s)$, they can be computed efficiently via dynamic programming.

Indeed, $\{Z_s\}_s$ can be computed recursively as follows:

$$Z_s = \begin{cases} \sum_{s'\in N(s)} \exp\left({}^{v(s;\mathbf{x})}/_{\mu}\right) Z_{s'} & \text{if } s \neq s_d \\ 1 & \text{if } s = s_d, \end{cases}$$

Algorithm 1: Dynamic **P**rogramming based algorithm (**DynP**) to solve *Maximizing* $g(\boldsymbol{x},\lambda)$

Input: $\lambda \in \mathbb{R}$ and an initial value of $\mathbf{x}$
while *not converged* **do**
  Given $\mathbf{x}$, solve the system $\mathbf{H} = (\mathbf{I}-\mathbf{M})^{-1}\mathbf{B}$ and $\mathbf{J}^{\mathbf{H},j} = (\mathbf{I}-\mathbf{M})^{-1}\mathbf{J}^{\mathbf{M},j}\mathbf{H}$ for all j
  Compute $g(\mathbf{x},\lambda)$ and $\frac{\partial g(\mathbf{x},\lambda)}{\partial x_s}$ using Eq. 14, 15.
  Update $\mathbf{x}$ using a projected gradient method
end

where $N(s) = \{s' \in \mathcal{S} |\ (s,s') \in \mathcal{A}\}$, is the set of possible next nodes that can be reached in one hop from node $s \in \mathcal{S}$.

Let $\mathbf{M}$ be a matrix of size $|\mathcal{S}\times\mathcal{S}|$ with entries defined as:

$$M_{ss'} = \exp\left({}^{v(s|\mathbf{x})}/_{\mu}\right)\ \forall\ s\in\mathcal{S}, s'\in N(s)$$

Then $\mathbf{Z} = \{Z_s, s\in\mathcal{S}\}$ is a solution to the linear system $\mathbf{Z} = \mathbf{MZ} + \mathbf{b}$, where $\mathbf{b}$ is of size $|\mathcal{S}| \times 1$ with zero entries except $b_{s_d} = 1$. Similarly, we can compute $\mathbf{Y}^s = \{Y^s_{s'}\}_{s'}$ recursively:

$$Y^s_{s'} = \begin{cases} \sum_{s''\in N(s')} \left(\exp\left({}^{v(s';\mathbf{x})}/_{\mu}\right)\right) Y_{s''} & \text{if } s' \neq s \\ 1 & \text{if } s' = s. \end{cases}$$

Clearly, $\mathbf{Y}^s$ is a solution to the linear system $\mathbf{Y}^s = \mathbf{MY}^s + \mathbf{b}^s$, where $\mathbf{b}^s$ is of size $|\mathcal{S}|$ with zeros everywhere except $b^s_s = 1$.

Since $\mathbf{Y}^s$ and $\mathbf{Z}$ are solutions to the systems $\mathbf{Y}^s = \mathbf{MY}^s + \mathbf{b}^s$ and $\mathbf{Z} = \mathbf{MZ} + \mathbf{b}$, respectively, $\forall s \in \mathcal{S}$, the objective $g(\mathbf{x},\lambda)$ can be computed via solving $|\mathcal{L}|+1$ system of linear equations. Finally, we see that all the above linear systems rely on the common matrix $\mathbf{M}$. We can group them all into only one linear system. Let $\mathbf{H}$ be a matrix of size $(|\mathcal{S}|) \times (|\mathcal{L}|+1)$ in which the 1st to $|\mathcal{L}|$-th columns are vectors $\mathbf{Y}^s$, $s \in \mathcal{L}$ and the last column is $\mathbf{Z}$. Let $\mathbf{B}$ be a matrix of size $(|\mathcal{S}|)\times(|\mathcal{L}|+1)$ in which the 1st to $|\mathcal{L}|$-th columns are vectors $\mathbf{b}^s$, $s \in \mathcal{L}$ and the last column is $\mathbf{b}$. We see that $\mathbf{H}$ is a solution to the linear system $(\mathbf{I}-\mathbf{M})\mathbf{H} = \mathbf{B}$. Thus, in general, we can solve only one linear system to obtain all $\mathbf{Y}^s$ and $\mathbf{Z}$. This way should be scalable when the size of $\mathcal{L}$ increases.

6.2 Gradient Computation

We aim at employing the gradient-based approach to solve the binary search step: $\max_{\mathbf{x}}\{g(\mathbf{x},\lambda)\}$ (aka. Eq. 3). The core is to compute the gradient $\left\{{}^{\partial g(\mathbf{x},\lambda)}/_{\partial x_s}\right\}$. According to Eq. 14, this gradient computation requires differentiating through the matrices $\mathbf{Z}$ and $\{\mathbf{Y}^s\}$ (or equivalently, differentiating through the matrix $\mathbf{H}$). We first present our Proposition 4:

Proposition 4. *$(\boldsymbol{I}-\boldsymbol{M})$ is invertible in a cycle-free network.*

Prop. 4 allows us to compute the matrix $\mathbf{H}$ as: $\mathbf{H} = (\mathbf{I}-\mathbf{M})^{-1}\mathbf{B}$. By taking the derivatives of both sides w.r.t x_j, $j \in \mathcal{L}$, we obtain the following: for all $j \in \mathcal{L}$,

$$\mathbf{J}^{\mathbf{H},j} = (\mathbf{I}-\mathbf{M})^{-1}\mathbf{J}^{\mathbf{M},j}(\mathbf{I}-\mathbf{M})^{-1}\mathbf{B} = (\mathbf{I}-\mathbf{M})^{-1}\mathbf{J}^{\mathbf{M},j}\mathbf{H},$$

where $\mathbf{J}^{\mathbf{H},j}$ and $\mathbf{J}^{\mathbf{M},j}$ are the gradient matrices of $\mathbf{H}$ and $\mathbf{M}$ w.r.t x_j, i.e., $\mathbf{J}^{\mathbf{H},j}$ is a matrix of size $|\mathcal{S}|\times(|\mathcal{L}|+1)$ with entries $\mathbf{J}^{\mathbf{H},j}_{ss'} = {}^{\partial H_{ss'}}/_{\partial x_j}$, and $\mathbf{J}^{\mathbf{M},j}$ is a matrix of size $(|\mathcal{S}| \times |\mathcal{S}|)$

with entries $\mathbf{J}^{\mathbf{M},j}_{ss'} = {}^{\partial M_{ss'}}/_{\partial x_j}$, for any $j \in \mathcal{L}$. Let $\mathbf{R}^l(\mathbf{x})$ be a matrix of size $1 \times |\mathcal{L}|$ with entries $r^l(s, x_s)$ for $s \in \mathcal{L}$. We use $A_{S,T}$ to denotes a sub-matrix of A which uses the rows in set S and columns in set T. If S or T is a singleton, e.g., $S = \{s_o\}$ or $T = |\mathcal{L}| + 1$, then we write it as s_o or $|\mathcal{L}| + 1$.

As a result, we now can compute the required gradient as follows for all $s \in \mathcal{L}$ where $\circ$ denotes Hadamard product:

$$\frac{\partial g(\mathbf{x}, \lambda)}{\partial x_s} = \left(\mathbf{R}^l(\mathbf{x}) \circ \mathbf{J}^{\mathbf{H},s}_{s_o,\mathcal{L}} + \mathbf{J}^{R,s} \circ \mathbf{H}_{s_o,\mathcal{L}}\right) \times \mathbf{H}_{\mathcal{L},|\mathcal{L}|+1} + \left(\mathbf{R}^l(\mathbf{x}) \circ \mathbf{H}_{s_o,\mathcal{L}}\right) \times \mathbf{J}^{\mathbf{H},j}_{\mathcal{L},|\mathcal{L}|+1} - \lambda \mathbf{J}^{\mathbf{H}}_{s_o,|\mathcal{L}|+1} \tag{15}$$

We summarize the main steps to optimize $g(\mathbf{x}, \lambda)$ in Alg. 1.

Remark 1. *Alg. 1 only guarantees a local optimum due to the non-convexity of $g(\boldsymbol{x}, \lambda)$. The complexity is determined by the matrix inversion which, in worst case, is in $O(|\mathcal{S}|^3)$. The gradient descent loop runs $O({}^1/_\epsilon)$ to provide an additive ϵ approximation. Thus, the total complexity is $O\left(({}^1/_\epsilon)|\mathcal{S}||\mathcal{A}|\right)$. In practice, the gradients can be found via auto differentiation techniques, providing significantly more speed-up.*

6.3 A Natural Special Case

Separation of critical resources and/or privileges is an important concept in security [Lin *et al.*, 2023]. Following this principle, we analyze a special yet natural security design scenario where that the critical nodes $\mathcal{L}$ are well separated. Specifically, we assume that the cost of travelling between nodes in $\mathcal{L}$ is high. More formally, given a critical node $s \in \mathcal{L}$, let $\Delta^+(s)$ be the set of paths that cross s and at least another critical node in $\mathcal{L}$. Let $\beta_1, \beta_2 > 0$ such that:

$$\beta_1 = \max_{\mathbf{x}} \max_{s \in \mathcal{L}} \left\{ \frac{\sum_{\tau \in \Delta^+(s)} \exp\left({}^{U(\tau;\mathbf{x})}/_\mu\right)}{\sum_{\tau \in \Delta(s)} \exp\left({}^{U(\tau;\mathbf{x})}/_\mu\right)} \right\}$$
$$\beta_2 = \max_{\mathbf{x}} \left\{ \frac{\sum_{\tau \in \bigcup_s \{\Delta^+(s)\}} \exp\left({}^{U(\tau;\mathbf{x})}/_\mu\right)}{\sum_{\tau \in \bigcup_s \{\Delta(s)\}} \exp\left({}^{U(\tau;\mathbf{x})}/_\mu\right)} \right\}, \tag{16}$$

Intuitively, β_1 and β_2 are expected to be small if the cost of traveling between any two critical nodes in $\mathcal{L}$ is large. That is, $\beta_1, \beta_2 \to 0$ as $\sum_{\tau \in \Omega(s,s')} \exp\left({}^{U(\tau;\mathbf{x})}/_\mu\right) \to 0$, where $\Omega(s, s')$ consists of all paths from s to s', for any $s, s' \in \mathcal{L}$. Surprisingly, even though **DynP** only finds a locally optimal solution for (OPT) due to its non-convexity, we show that assuming small β_1 and β_2 provides approximation guarantees for the globally optimal solution value.

For this approximation, we need mild assumptions that the utilities have a linear form: $v(s; \mathbf{x}) = w^f_s x_s + t^f_s$ and $r^l(s; \mathbf{x}) = r^l(s, x_s) = w^l_s x_s + t^l_s$ for some constants $w^f_s, t^f_s, w^l_s, t^l_s$. We assume that $w^f_s < 0$ and $w^l_s > 0$, i.e., more resources x_s at s will lower adversary's utilities, and increase the defender's utility. This setting is intuitive for security settings [Yang *et al.*, 2012; Mai and Sinha, 2022].

We first introduce a restricted interdiction problem that can be solved *optimally* in a tractable manner using our efficient gradient descent-based method. We then present an important theoretical result showing how the restricted problem's solution yields an approximate solution of with the original problem for well separated critical nodes.

Let $\Delta(s)$ be the set of paths that cross a critical node s and do not cross any other node in $\mathcal{L}$. We consider the following restricted interdiction problem:

$$\max_{\mathbf{x}} \widetilde{\mathcal{F}}(\mathbf{x}) = \frac{\sum_{s \in \mathcal{L}, \tau \in \Delta(s)} r^l(s, x_s) \exp\left({}^{U(\tau;\mathbf{x})}/_\mu\right)}{\sum_{s \in \mathcal{L}, \tau \in \Delta(s)} \exp\left({}^{U(\tau;\mathbf{x})}/_\mu\right)} \tag{Approx-OPT}$$
$$\text{s.t.} \sum\nolimits_{s \in \mathcal{L}_k} x_s \le M_k, \ \forall k \in [K]$$
$$x_s \in [L^x, U^x], \ \forall s \in \mathcal{L}.$$

Intuitively, in this restricted problem (Approx-OPT), the adversary's path choices are restricted to a subspace of paths in the network which only cross a *single* critical node in $\mathcal{L}$. We denote by $\mathcal{X}$, the feasible set of the defender's interdiction strategies $\mathbf{x}$ that satisfy the constraints in (Approx-OPT).

Solution Relation with Original Problem (OPT)

We now theoretically analyze (Approx-OPT)'s solution in relation to our original problem (OPT). We prove that:

Theorem 2. *Let $\boldsymbol{x}^*$ be an approx. solution to* (Approx-OPT)*: $\max_{\boldsymbol{x} \in \mathcal{X}} \widetilde{\mathcal{F}}(\boldsymbol{x})$ such that $\widetilde{\mathcal{F}}(\boldsymbol{x}^*) \ge (1 - \epsilon) \max_{\boldsymbol{x}} \widetilde{\mathcal{F}}(\boldsymbol{x})$ for given $\epsilon > 0$, let $\kappa = \max_{\boldsymbol{x} \in \mathcal{X}} \sum_{s \in \mathcal{L}} |r^l(s, x_s)|$ be the maximal absolute reward that the defender can possibly achieve at a critical node, then we obtain:*

$$\mathcal{F}^l(\boldsymbol{x}^*) \ge \frac{(1 - \epsilon) \max_{\boldsymbol{x}} \left\{\mathcal{F}^l(\boldsymbol{x})\right\}}{(1 + \beta_1)(1 + \beta_2)} - \kappa \frac{\epsilon + \beta_1 + \beta_2 + \beta_1 \beta_2}{(1 + \beta_1)(1 + \beta_2)}.$$

Additionally, if $\boldsymbol{x}^*$ is an approx. solution with an additive error $\epsilon > 0$, we obtain the following bound:

$$\mathcal{F}^l(\boldsymbol{x}^*) \ge \left({}^1/_\eta\right) \max_{\boldsymbol{x}} \{\mathcal{F}^l(\boldsymbol{x})\} - \kappa\left({}^{(\eta-1)}/_\eta\right),$$

where $\eta = (1 + \beta_1)(1 + \beta_2)\left(1 + \frac{\epsilon}{\kappa + \min_{\boldsymbol{x} \in \mathcal{X}} \widetilde{\mathcal{F}}(\boldsymbol{x})}\right)$.

Note that $\max_{\mathbf{x} \in \mathcal{X}} \{\mathcal{F}^l(\mathbf{x})\}$ is the original problem (OPT) to find an optimal defender strategy. As stated previously, when the cost of traveling between any two critical nodes is high, (β_1, β_2) is close to zero, meaning the RHS of both inequalities in Theorem 2 will closely approximate the optimal solution value of (OPT).

Solving the Restricted Problem: To solve the restricted problem, we also apply binary search. It can be demonstrated that each sub-problem of the binary search can be effectively solved to optimality (or near-optimality) using a gradient-based method. Due to limited space, the details of this approach is provided in the appendix along with the sub-results that lead to the main Theorem 2.

7 Numerical Experiments

To illustrate the efficacy of our proposed algorithms, we perform experiments on synthetic data.

7.1 Experimental Settings

Data generation. We generate random graphs (cycle-free) with $|\mathcal{S}|$ vertices and edge probability p. We randomly choose $|\mathcal{L}|$ vertices (except source and destination) as the critical nodes that can be attacked. We set $|\mathcal{L}| = 0.8 \times |\mathcal{S}|$. In

addition, the defender weights $\{(w_j^l, t_j^l) \mid j \in [|\mathcal{L}|]\}$ are generated uniformly at random from the interval $[0, 1]$ and the adversary weights $\{(w_j^f, t_j^f) \mid j \in [|\mathcal{S}|]\}$ are generated at random from the interval $[-1, 0]$. Moreover, we used $p = 0.8, \mu = 2$.

Baseline. We approximate the sums over exponentially many paths in Equation 3 by sampling paths from the network and run gradient descent on this expression to estimate the optimal decision variable. To sample paths for the baseline, a resource allocation $\mathbf{x}$ is assigned to $\mathcal{L}$ and the follower is initially placed at the origin s_o. Its next node s_1 is sampled from the distribution

$$\pi^f(s|s_0, \mathbf{x}) = \frac{\exp\left(v(s;\mathbf{x})/\mu\right) Z_s}{\sum_{s' \in N(s_0)} \exp\left(v(s';\mathbf{x})/\mu\right) Z_{s'}},$$

where $s \in \{$*nodes having an edge to* $s_0\}$ and $N(s_0)$ is the set of outgoing nodes from s_0. Similarly $s_2, \ldots$ etc. are sampled till the destination s_d is reached. This sampling is repeated 1000 times per iteration and then average is taken to get the objective. Based on the gradients, the resource allocation $\mathbf{x}$ is updated which changes the transition probabilities and the process is repeated again until convergence. Ten different values of $\mathbf{x}$ were taken and the seed with the lowest loss was reported. We will compare this baseline against our near optimal MILP-based algorithm, **LiSD** and our dynamic programming based algorithm, **DynP**. To ensure fairness, all algorithms were run with the same number of epochs.

Choosing N and K for the LiSD To justify our choices of N and K for the **LiSP** described, we first fix $N = 90$ and vary K from 5 to 100. For each value of K, we generate 10 independent instances and solve them using the MILP approach. We then observe that the optimal values given by $K = 20$ are only 0.3% different from those given by the largest value of K (i.e., $K = 100$). Moreover, the optimal values given by $N = 90$ are only about 3% different from those from the largest value of N, i.e., $N = 150$. The numerical details can be found in the appendix. We therefore choose $N = 90$ and $K = 20$ for the **LiSD** approach. According to the above analyses, these choices would suffice to guarantee low practical approximation errors stemming from both path-sampling and PL approximation. We use GUROBI (a SOTA MILP solver) to solve (MILP). All our experiments were run on a 2.1 GHz CPU with 128GB RAM.

7.2 Numerical Comparison

We vary the number of nodes from 20 to 100. For each choice of number of nodes, we generate 20 independent instances and solve them by the three methods (Baseline, **DynP**, and **LiSD**). The rates of giving best objective values (over 20 instances) are reported in Table 1. **LiSD** consistently outperforms the other methods, by a large margin, in terms of the number of times it returns the best objective values. **DynP** performs better than Baseline for large-size settings, but worse than Baseline for small-size ones. Note that, among the four methods, only **LiSD** can guarantee near-optimal solutions. The computing times are not directly comparable, as GUROBI ran on several cores while the other gradient-based methods use only one CPU core. We however observe that, for instances of 100 nodes, the average computing times for the Baseline, **DynP**, and **LiSD** are approximately 3, 15 and 1.8 minutes.

# nodes	Baseline	(Ours) LiSD	(Ours) DynP
20	30%	**50%**	0%
40	20%	**65%**	15%
60	15%	**55%**	30%
80	20%	**45%**	35%
100	15%	**45%**	15%

Table 1: Rates of giving best objective values. Each measurement is computed using 20 independent instances.

# nodes	Baseline	(Ours) LiSD	(Ours) DynP
20	253.1%	**344.2%**	301.9%
40	54.5%	**63.2%**	55.9%
60	51.7%	**57.6%**	55.8%
80	25.3%	**34.7%**	30.4%
100	88.2%	**93.0%**	89.0%

Table 2: Average percentage improvements w.r.t the lowest objective values given by the four methods.

We further compare the objective values returned by the four methods by computing the percentage improvement of each objective w.r.t. the lowest objective value given by the four approaches. Specifically, we solve each instance to obtain 4 objective values. We then compute the percentage improvement of each objective value w.r.t the lowest value among the four values. The *average* percentage values are reported in Table 2 below, which show that **LiSD** performs the best, and **DynP** outperforms the Baseline.

8 Conclusion

Network interdiction game problems present a set of challenges that appear intractable to start with. In this work, we address some of these challenges by providing novel methods that efficiently solve a class of network interdiction problems with approximation guarantees. We are the first to study the dynamic Quantal Response model in the type of network interdiction studied in [Fulkerson and Harding, 1977; Israeli and Wood, 2002]. We believe this modeling and methodology contribution provides suggestions for other future research directions, such as a variant where the adversary's objective is to maximize a flow through the network or a setting where the leader would also need to make dynamic and time-dependent decisions. This online nature of the problem suggests possible future work in online learning problems such as [Bose *et al.*, 2023a]. It is interesting to analyse scenarios where network structures arise naturally such as ride pool matching [Bose and Varakantham, 2021] and are under threat from adversaries such as competing providers.

Acknowledgments

Dr. Tien Mai and Dr. Arunesh Sinha were supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme

(AISG Award No: AISG2- RP-2020-017). Dr. Nguyen was supported by grant W911NF-20-1-0344 from the US Army Research Office.

References

[Aguirregabiria and Mira, 2010] Victor Aguirregabiria and Pedro Mira. Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, 156(1):38–67, 2010.

[Basilico *et al.*, 2009] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, pages 57–64, 2009.

[Basilico *et al.*, 2017] Nicola Basilico, Giuseppe De Nittis, and Nicola Gatti. Adversarial patrolling with spatially uncertain alarm signals. *Artificial Intelligence*, 246:220–257, 2017.

[Ben-Tal and Nemirovski, 2002] Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization–methodology and applications. *Mathematical programming*, 92(3):453–480, 2002.

[Bertsekas and Tsitsiklis, 1991] Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[Borrero *et al.*, 2016] Juan S Borrero, Oleg A Prokopyev, and Denis Sauré. Sequential shortest path interdiction with incomplete information. *Decision Analysis*, 13(1):68–98, 2016.

[Bose and Varakantham, 2021] Avinandan Bose and Pradeep Varakantham. Conditional expectation based value decomposition for scalable on-demand ride pooling. *arXiv preprint arXiv:2112.00579*, 2021.

[Bose *et al.*, 2022] Avinandan Bose, Arunesh Sinha, and Tien Mai. Scalable distributional robustness in a class of non-convex optimization with guarantees. *Advances in Neural Information Processing Systems*, 35:13826–13837, 2022.

[Bose *et al.*, 2023a] Avinandan Bose, Mihaela Curmei, Daniel L Jiang, Jamie Morgenstern, Sarah Dean, Lillian J Ratliff, and Maryam Fazel. Initializing services in interactive ml systems for diverse users. *arXiv preprint arXiv:2312.11846*, 2023.

[Bose *et al.*, 2023b] Avinandan Bose, Tracey Li, Arunesh Sinha, and Tien Mai. A fair incentive scheme for community health workers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14127–14135, 2023.

[Boyd *et al.*, 2004] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Černỳ *et al.*, 2021] Jakub Černỳ, Viliam Lisỳ, Branislav Bošanskỳ, and Bo An. Dinkelbach-type algorithm for computing quantal stackelberg equilibrium. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 246–253, 2021.

[Cormican *et al.*, 1998] Kelly J Cormican, David P Morton, and R Kevin Wood. Stochastic network interdiction. *Operations Research*, 46(2):184–197, 1998.

[Dempe *et al.*, 2015] Stephan Dempe, Vyacheslav Kalashnikov, Gerardo A Pérez-Valdés, and Nataliya Kalashnykova. Bilevel programming problems. *Energy Systems. Springer, Berlin*, 10:978–3, 2015.

[Dinkelbach, 1967] Werner Dinkelbach. On nonlinear fractional programming. *Management science*, 13(7):492–498, 1967.

[Fang *et al.*, 2016] Fei Fang, Thanh Nguyen, Rob Pickles, Wai Lam, Gopalasamy Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. Deploying paws: Field optimization of the protection assistant for wildlife security. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, pages 3966–3973, 2016.

[Fosgerau *et al.*, 2013] M. Fosgerau, E. Frejinger, and A. Karlström. A link based network route choice model with unrestricted choice set. *Transportation Research Part B*, 56:70–80, 2013.

[Fulkerson and Harding, 1977] Delbert Ray Fulkerson and Gary C Harding. Maximizing the minimum source-sink path subject to a budget constraint. *Mathematical Programming*, 13(1):116–118, 1977.

[Haghtalab *et al.*, 2016] Nika Haghtalab, Fei Fang, Thanh H. Nguyen, Arunesh Sinha, Ariel D. Procaccia, and Milind Tambe. Three strategies to success: Learning adversary models in security games. In *25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[Hoeffding, 1994] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.

[Israeli and Wood, 2002] Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks: An International Journal*, 40(2):97–111, 2002.

[Jain *et al.*, 2011] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 327–334, 2011.

[Leitmann, 1978] George Leitmann. On generalized stackelberg strategies. *Journal of optimization theory and applications*, 26(4):637–643, 1978.

[Lin *et al.*, 2023] Kunli Lin, Wenqing Liu, Kun Zhang, and Bibo Tu. Hyperps: A virtual-machine memory protection approach through hypervisor's privilege separation. *IEEE Transactions on Dependable and Secure Computing*, 20(4):2925–2938, 2023.

[Mai and Sinha, 2022] Tien Mai and Arunesh Sinha. Choices are not independent: Stackelberg security games with nested quantal response models.(2022). In *Proceedings of 36th AAAI Conference on Artificial Intelligence (AAAI), Vancouver, Canada*, pages 1–9, 2022.

[Mai *et al.*, 2015] Tien Mai, Mogens Fosgerau, and Emma Frejinger. A nested recursive logit model for route choice analysis. *Transportation Research Part B*, 75(0):100 – 112, 2015.

[McFadden, 1981] Daniel McFadden. Econometric models of probabilistic choice. In C. Manski and D. McFadden, editors, *Structural Analysis of Discrete Data with Econometric Applications*, chapter 5, pages 198–272. MIT Press, 1981.

[Milec *et al.*, 2020] David Milec, Jakub Černỳ, Viliam Lisỳ, and Bo An. Complexity and algorithms for exploiting quantal opponents in large two-player games. *arXiv preprint arXiv:2009.14521*, 2020.

[Miller, 1984] Robert A Miller. Job matching and occupational choice. *The Journal of Political Economy*, pages 1086–1120, 1984.

[Rust, 1987] John Rust. Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica*, 55(5):999–1033, 1987.

[Sefair and Smith, 2016] Jorge A Sefair and J Cole Smith. Dynamic shortest-path interdiction. *Networks*, 68(4):315–330, 2016.

[Smith and Song, 2020] J Cole Smith and Yongjia Song. A survey of network interdiction models and algorithms. *European Journal of Operational Research*, 283(3):797–811, 2020.

[Smith *et al.*, 2009] J Cole Smith, Churlzu Lim, and Aydın Alptekinoğlu. New product introduction against a predator: A bilevel mixed-integer programming approach. *Naval Research Logistics (NRL)*, 56(8):714–729, 2009.

[Tambe, 2011] Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press, 2011.

[Train, 2003] Kenneth Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2003.

[Wang *et al.*, 2020] Kai Wang, Andrew Perrault, Aditya Mate, and Milind Tambe. Scalable game-focused learning of adversary models: Data-to-decisions in network security games. In *AAMAS*, pages 1449–1457, 2020.

[Wolpin, 1984] Kenneth I Wolpin. An estimable dynamic stochastic model of fertility and child mortality. *The Journal of Political Economy*, pages 852–874, 1984.

[Xue *et al.*, 2021] Wanqi Xue, Youzhi Zhang, Shuxin Li, Xinrun Wang, Bo An, and Chai Kiat Yeo. Solving large-scale extensive-form network security games via neural fictitious self-play. *arXiv preprint arXiv:2106.00897*, 2021.

[Xue *et al.*, 2022] Wanqi Xue, Bo An, and Chai Kiat Yeo. Nsgzero: Efficiently learning non-exploitable policy in large-scale network security games with neural monte carlo tree search. *arXiv e-prints*, pages arXiv–2201, 2022.

[Yang *et al.*, 2011] Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Richard John. Improving resource allocation strategy against human adversaries in security games. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[Yang *et al.*, 2012] Rong Yang, Fernando Ordonez, and Milind Tambe. Computing optimal strategy against quantal response in security games. In *AAMAS*, pages 847–854, 2012.

[Zhang *et al.*, 2019] Youzhi Zhang, Qingyu Guo, Bo An, Long Tran-Thanh, and Nicholas R Jennings. Optimal interdiction of urban criminals with the aid of real-time information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1262–1269, 2019.

[Zimmermann and Frejinger, 2020] Maëlle Zimmermann and Emma Frejinger. A tutorial on recursive models for analyzing and predicting path choice behavior. *EURO Journal on Transportation and Logistics*, 9(2):100004, 2020.