Optimal Interdiction of Illegal Network Flow

Qingyu Guo[1], Bo An[2], Yair Zick[3], Chunyan Miao[2]
[1]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, NTU, Singapore
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3]School of Computer Science, Carnegie-Mellon University, USA
[1,2]{qguo005,boan,ascymiao}@ntu.edu.sg,[3]yairzick@cmu.edu

Abstract

Large scale smuggling of illegal goods is a long-standing problem, with $\$1.4b$ and thousands of agents assigned to protect the borders from such activity in the US-Mexico border alone. Illegal smuggling activities are usually blocked via inspection stations or ad-hoc checkpoints/roadblocks. Security resources are insufficient to man all stations at all times; furthermore, smugglers regularly conduct surveillance activities. This paper makes several contributions toward the challenging task of optimally interdicting an illegal network flow: i) A new Stackelberg game model for network flow interdiction; ii) A novel Column and Constraint Generation approach for computing the optimal defender strategy; iii) Complexity analysis of the column generation subproblem; iv) Compact convex nonlinear programs for solving the subproblems; v) Novel greedy and heuristic approaches for subproblems with good approximation guarantee. Experimental evaluation shows that our approach can obtain a robust enough solution outperforming the existing methods and heuristic baselines significantly and scale up to realistic-sized problems.

1 Introduction

Stopping undesirable behavior on a network is a problem pertaining to many security domains: disruption of enemy supply chains, infectious disease control, and the interception of illegal goods such as drugs or weapons. Physical networks are typically defended via *checkpoints*: physical roadblocks or inspection points positioned at various points in the network. In all network interdiction scenarios, the objective of those defending the network is to minimize the amount of flow through the network. From a resource optimization perspective, this is an extremely challenging task. There are several factors interrelating here: first, defender resources are limited; second, placing a guard at some point on the network does not guarantee that any smuggler passing through will be caught - it merely increases the likelihood of a successful capture; finally, it is natural to assume that smugglers have prior knowledge of defenders' positioning via intelligence gathering. The defender is thus placed at a disadvantage: not only does she have limited capability to stop an attack, her agents' moves are also monitored; thus, smugglers may successfully evade checkpoints if these are constantly positioned in a predictable manner. Indeed, randomized allocation strategies are imperatively needed [GAO, 2009]. Our work addresses the following challenge:

> *Devise a formal methodology for guarding a large, complex network against undesirable flow, considering action observability and limited resources.*

Our Contributions: In this paper, we introduce a novel Network Flow Interdiction Game (*NFIG*) model, where the defender allocates a fixed number of security resources on the network, while the adversary commits to a feasible network flow. To compute the optimal defender strategy, we first provide a standard minimax bilevel formulation and reformulate it as a linear program (*LP*). However, due to the huge number of constraints and variables caused by exponentially large numbers of defender strategies and network paths, the *LP* is hard to solve. To overcome the computational challenge, we propose a *Column and Constraint Generation (CCG)* algorithm, with the following key contributions: i) we show that the algorithm converges to the Stackelberg equilibrium with finite iterations; ii) we show the NP-hardness of the *Column Generation* subproblem, and provide several novel algorithms to solve it, including an exact compact convex nonlinear program and a greedy algorithm with constant-factor approximation guarantee; iii) we provide an exact compact convex nonlinear program for the *Constraint Generation* subproblem as well as a fast local search based heuristic algorithm; iv) extensive experimental evaluation shows that our *CCG* framework can scale up to realistic-sized *NFIG* instances and significantly outperform existing approaches.

Related Work: The *network interdiction problem* is well studied [Church *et al.*, 2004]. In this model, we are given a weighted, directed or undirected graph, and our goal is minimizing the flow through the graph via deletion of edges/nodes (other goals — such as maximizing the shortest path, increasing detection probability — and other interdiction methods — such as decreasing edge capacity — have been studied as well [Altner *et al.*, 2010; Ball *et al.*, 1989; Israeli and Wood, 2002; Malik *et al.*, 1989; Wood, 1993; Guo *et al.*, 2016]. Other works study *stochastic network interdiction* where the interdiction action is successful with some known probability p [Pan *et al.*, 2001; Pan, 2005]. How-

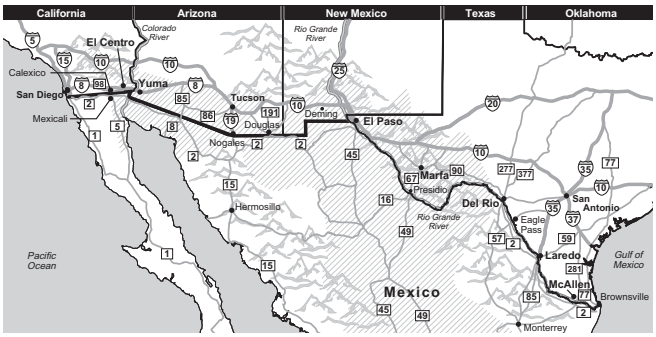


Figure 1: Topography and Road Systems in the Southern Border.

ever, none of them consider the randomized resource allocation strategy, i.e., the mixed strategy, due to the impossibility to change the resource allocation frequently, especially for interdiction actions such as destroying arcs. Moreover, randomized resource allocation is known to be necessary to improve the interdiction efficiency; for example, in the smuggling domain, checkpoint operation status can actually change daily [GAO, 2009]. However, randomization does lead to a drastic increase in problem complexity: while in the deterministic network interdiction model an adversary's strategy has a compact representation (as a network flow), it is not the case in our setting.

We model NFIG as a *Stackelberg security game*; security games try and find the optimal resource allocation to defend some valuable facility, based on game-theoretic methodologies [Tambe, 2011; Yin *et al.*, 2014]. However, in standard security game models, attacker strategies are typically compactly representable (e.g. a distribution over possible targets) [An *et al.*, 2013; Yin *et al.*, 2015; Gan *et al.*, 2015]. Some recent work proposes network security games [Tsai *et al.*, 2010; Jain *et al.*, 2011; 2013]; here, the defender allocates security resources on a (transportation) network and the attacker chooses a reliable path to escape, and both players have exponentially large strategy spaces; our model differs from these works in several important ways. Most importantly, our goal is minimizing the flow through a network, rather than preventing a single attacker from escaping; thus, the methodology proposed in existing works cannot be directly applied to our setting.

Motivating Domain: Our model is quite general, and can easily apply to various network security domains; to illustrate the underlying issues and motivation for our model choice, we now briefly present the problem in the context of stemming the drug flow through the US-Mexico border patrol. Figure 1 shows the major roads in the southern border area [GAO, 2009]. Drug smuggling in the southern border is dominated by 4 large well-organized and independent cartels: Sinaloa, Juarez, Gulf and Los Zetas, each of them controls the drug trafficking over 1000-mile border and has its own *major area of influence* in the United States [Beittel, 2015; Rosenberg, 2015]. These organizations typically use commercial, private and rental vehicles to smuggle illegal goods via *land points of entry (sources)*. The goods travel along different paths on the road network to different *destination cities (sinks)* [DOJ, 2010; Rosenberg, 2015]. Traffickers typically choose multiple transport paths, in order to avoid raising obvious suspicion by the inspection stations, and to mitigate losses in case of capture [Steinrauf, 1991].

In 2009, the Border Patrol operated 39 *tactical checkpoints* with flexible and daily changeable operation status in the southern border, which are generally operated at fixed locations. However, due to the shortage of staff, canines and basic facilities (i.e. *security resources*), only few of them are actually operational [GAO, 2009]. With this limitation for tactical checkpoints, *randomized allocation strategies* of limited security resources are essential; smugglers have the ability to observe the operational status of checkpoints, and make their best decision on trafficking activities, such as the amount of drugs to move through different POEs and paths. Indeed, a testimony before Congress by the Arizona Attorney General, revealed the sophisticated surveillance and communication technologies used to monitor security vulnerabilities and adjust plans to increase the chance of success [Kibble, 2009].

2 Preliminaries

Our proposed *NFIG* models an attacker and a defender who take actions on a capacitated graph $G = (V, E)$, with nodes set V and edges set E, and a capacity vector $\mathbf{c}$, where capacity c_e represents the maximum amount of adversary flow passing through edge e without arousing obvious suspicion by inspection facilities (permanent checkpoints, sensors, etc). We assume that the graph has a unique *source node* (POE) $s \in V$ and a unique *sink node* $t \in V$ (drug distribution city in smuggling scenario). The unique source/sink assumption is no loss of generality: a graph with multiple source nodes and sink nodes can be transformed into a single-source-sink graph by adding two new nodes s and t as the new unique source and sink nodes respectively, connecting s to each source node and t to each sink node with proper capacitated edges. Let $\mathcal{P}$ denote the set of all s-t paths in G. For a path $p \in \mathcal{P}$, we say node $v \in p$ (edge $e \in p$) if p passes through v (e). Let I denote the set of all *inspection stations*, such as the tactical checkpoints in the border patrol scenario, and an inspection station is operated if the defender assigns a security resource on it, such as staffs, canines, and inspection facilities. Each station $i \in I$ is characterized by a location, either a node or an edge in the graph (with a bit abuse of notation, we use i interchangeable with its location, such that $i \in V \cup E$), and a constant parameter $\tau_i \in [0, 1]$ denoting the proportion of adversary flow interdicted at i when operated, i.e., *inspection probability*. For example, in the border patrol scenario, the officers at checkpoint i can conduct regular inspections on passing vehicles and an illegal trafficker will be caught with some probability τ_i depending on the officers' experience, which can be treated as a constant factor. Let $k < |I|$ be the number of security resources owned by the defender.

Strategies: A defender pure strategy $S = \langle S_i \rangle$ is an allocation of k security resources to k inspection stations, i.e., $\sum_{i \in I} S_i = k$, where $S_i \in \{0, 1\}$ and $S_i = 1$ indicates that the inspection station i is operated. The defender pure strategy space is denoted by $\mathcal{S}$. A mixed defender strategy

$\mathbf{x} = \langle x_S \rangle$ is a probability distribution over all pure strategies where x_S denotes the probability that S is played. Let $\mathcal{X}$ denote the defender mixed strategy space.

An attacker strategy is a network flow $\mathbf{f}$ with f_p representing the amount of adversary flow passing along path $p \in \mathcal{P}$.[1] In order to avoid raising obvious suspicion, a feasible attacker strategy must satisfy the capacity constraint on each edge. Let $\mathcal{F}$ denote the set of all feasible attacker strategies, i.e.,

$$\mathcal{F} = \{\mathbf{f} \geq \mathbf{0} : \sum\nolimits_{p \in \mathcal{P}: e \in p} f_p \leq c_e \quad \forall e \in E\}. \tag{1}$$

Utility: Since the attacker aims at maximizing his successful drug flow while the defender wants to minimize it, we assume *NFIG* a zero-sum game. Given a pure defender strategy S and an attacker feasible flow $\mathbf{f}$, the attacker's utility is the sum of successful flows on all paths, i.e., $U_a(S, \mathbf{f}) = \sum_{p \in \mathcal{P}} \Phi(S, p) f_p$, where

$$\Phi(S, p) = \prod\nolimits_{i \in I: i \in p} (1 - \tau_i)^{S_i} \tag{2}$$

and $\Phi(S, p)$ represents the proportion of adversary flow on path p not interdicted by the operated inspection stations given S. The defender utility is $U_d(S, \mathbf{f}) = -U_a(S, \mathbf{f})$.

Given a defender mixed strategy $\mathbf{x}$ and an attacker feasible flow $\mathbf{f}$, the attacker's and defender's expected utilities are: $U_a(\mathbf{x}, \mathbf{f}) = \sum_{S \in \mathcal{S}} x_S U_a(S, \mathbf{f})$ and $U_d(\mathbf{x}, \mathbf{f}) = -U_a(\mathbf{x}, \mathbf{f})$.

Equilibrium: *NFIG* is a leader-follower game, for which the widely adopted solution concept is the Stackelberg equilibrium (*SE*). Let $y(\mathbf{x})$ denote the best response flow against defender mixed strategy $\mathbf{x}$. A strategy profile $\langle \mathbf{x}^*, \mathbf{f}^* \rangle$ forms an *SE*, if: i) $\mathbf{f}^* = y(\mathbf{x}^*)$, and ii) $U_d(\mathbf{x}^*, \mathbf{f}^*) \geq U_d(\mathbf{x}, y(\mathbf{x}))$ for all $\mathbf{x} \in \mathcal{X}$. With the zero-sum assumption, the *SE* can be obtained by the following minimax formulation:

$$\min\nolimits_{\mathbf{x} \in \mathcal{X}} \max\nolimits_{\mathbf{f} \in \mathcal{F}} U_a(\mathbf{x}, \mathbf{f}). \tag{3}$$

Since $\mathcal{X}, \mathcal{F}$ are convex sets, and $U_a(\mathbf{x}, \mathbf{f})$ is a linear function in $\mathbf{x}$ when fixing $\mathbf{f}$ and vice versa, according to von Neumann's minimax theorem [Nlkaido, 1954], we have:

$$\min\nolimits_{\mathbf{x} \in \mathcal{X}} \max\nolimits_{\mathbf{f} \in \mathcal{F}} U_a(\mathbf{x}, \mathbf{f}) = \max\nolimits_{\mathbf{f} \in \mathcal{F}} \min\nolimits_{\mathbf{x} \in \mathcal{X}} U_a(\mathbf{x}, \mathbf{f}) \tag{4}$$

A direct consequence of the minmax theorem, critical for our key solution approach (Section 3.2), is the equivalence of *SE* and the Nash equilibrium (*NE*), where $\mathbf{x}^*$ is the best response against $\mathbf{f}^*$ and $\mathbf{f}^*$ is also the best response against $\mathbf{x}^*$.

3 Solution Approach

In this section, we first provide an LP formulation to compute the equilibrium based on the standard minimax formulation. Since this LP has exponentially large number of variables and constraints, we then propose a novel Column and Constraint Generation (*CCG*) to further improve the scalability.

[1]Note that the compact representation $\mathbf{f} = \langle f_e \rangle$ is infeasible in *NFIG*. Please see Section B of Online Appendix for the explanation available at: http://www.ntu.edu.sg/home/boan/papers/IJCAI16_Flowinterdiction_Appendix.pdf.

3.1 LP Formulation

We start from the standard minimax formulation for *SE*:

$$\min_{\mathbf{x}} \max_{\mathbf{f}} \quad \sum\nolimits_{S \in \mathcal{S}, p \in \mathcal{P}} \Phi(S, p) x_S f_p \tag{5a}$$

$$\text{s.t.} \quad \sum\nolimits_{S \in \mathcal{S}} x_S = 1 \tag{5b}$$

$$\sum\nolimits_{p \in \mathcal{P}: e \in p} f_p \leq c_e \quad \forall e \in E \tag{5c}$$

$$\mathbf{x} \geq \mathbf{0}, \mathbf{f} \geq \mathbf{0}. \tag{5d}$$

The objective in Eq.(5a) is the attacker's expected utility $U_a(\mathbf{x}, \mathbf{f})$. The bilevel minimax program (5) can be reformulated as a linear program by replacing the inner program with its dual, and the resulting linear program (*LP*) is as follows:

$$\min_{\mathbf{x}, \mathbf{u}} \quad \sum\nolimits_{e \in E} c_e u_e \tag{6a}$$

$$\text{s.t.} \quad \sum\nolimits_{S \in \mathcal{S}} x_S = 1 \tag{6b}$$

$$\sum\nolimits_{S \in \mathcal{S}} x_S \Phi(S, p) \leq \sum\nolimits_{e \in p} u_e \quad \forall p \in \mathcal{P} \tag{6c}$$

$$\mathbf{x} \geq \mathbf{0}, \mathbf{u} \geq \mathbf{0}. \tag{6d}$$

The dual solution with respect to inequality (6c) provides the equilibrium attacker flow $\mathbf{f}$, defined over all paths in $\mathcal{P}$.

3.2 Column and Constraint Generation

The linear program (6) is challenging to solve due to the exponentially large number of variables and constraints ($|\mathcal{S}|, |\mathcal{P}|$). To address this challenge, we propose a novel algorithm called *Column and Constraint Generation (CCG)*, and prove the correctness of the *CCG* algorithm based on the property that the *SE* of SNIG is the same as the *NE* (Section 2).

The algorithm *CCG* is sketched in Algorithm 1. Initially, a small space $\langle \mathcal{S}', \mathcal{P}' \rangle$ is generated with arbitrary candidate defender strategies and s-t paths. Then *CCG* solves a restricted version of *NFIG* with $LP(\mathcal{S}', \mathcal{P}')$, where the defender pure strategy space is $\mathcal{S}'$ and the attacker flow is restricted to paths only in $\mathcal{P}'$, i.e., Eqs.(6a)–(6d) with $\langle \mathcal{S}, \mathcal{P} \rangle$ replaced by $\langle \mathcal{S}', \mathcal{P}' \rangle$, and obtains primal and dual solutions $(\mathbf{x}, \mathbf{u}, \mathbf{f})$. This restricted *NFIG* can be solved efficiently since the game is very small. Obviously, the obtained primal solution $\mathbf{x}$ and dual solution $\mathbf{f}$ form an *SE* of the restricted *NFIG*, but not necessarily the original *NFIG*. Therefore, two subproblems *ColG* (Column Generation) and *ConG* (Constraint Generation) are proposed to generate useful defender pure strategies and s-t paths into $\langle \mathcal{S}', \mathcal{P}' \rangle$ in order to guide the *SE* of the restricted *NFIG* to the one of the original *NFIG*. Respectively, *ColG* generates a defender strategy $S \in \mathcal{S}$ which is a best response against $\mathbf{f}$, and *ConG* generates an s-t path $p \in \mathcal{P}$ with maximal *reduced cost* which measures the degree of violating inequality (6c).

The key of the *CCG* algorithm is that once *ColG* generates the pure strategy S already in $\mathcal{S}'$, the current solution $\mathbf{x}$ of restricted *NFIG* is the defender best response against the adversary flow $\mathbf{f}$ in the original *NFIG* (PROPOSITION 1), and similarly, if *ConG* generates an s-t path p with non-positive reduced cost, the current equilibrium flow $\mathbf{f}$ of restricted *NFIG* is the attacker best response flow against $\mathbf{x}$ in the original *NFIG* (PROPOSITION 7). Therefore, the *NE* of the original

Algorithm 1: CCG Algorithm for NFIG

Initialize $\mathcal{S}'$ by generating arbitrary defender strategies;
Initialize $\mathcal{P}'$ by generating arbitrary s-t paths;
repeat
  $(\mathbf{x}, \mathbf{u}, \mathbf{f}) \leftarrow LP(\mathcal{S}', \mathcal{P}')$;
  $S \leftarrow ColG(\mathbf{f})$;
  $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S\}$;
  $p \leftarrow ConG(\mathbf{x}, \mathbf{u})$;
  $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{p\}$;
until convergence;
return $(\mathbf{x}, \mathbf{f})$.

NFIG is achieved when *ColG* generates a pure strategy S already in $\mathcal{S}'$ and *ConG* generates an s-t path p with non-positive reduced cost. Since the *NE* is the same as the *SE*, the current solution pair $(\mathbf{x}, \mathbf{f})$ forms the *SE* of the original *NFIG*.

Comparison with Double Oracle: The *double oracle (DO)* algorithm, first proposed by McMahan *et al.* [2003], is a standard method for solving two-player zero-sum games with large scale strategy spaces, and is widely adopted by security game research [Jain *et al.*, 2011; 2013; Vanek *et al.*, 2012; Wang *et al.*, 2016]. In *DO*, both players commit to mixed strategies. Thus, to apply *DO* to *NFIG*, the attacker's strategy is a probability distribution over all feasible flows, rather than a single flow. Besides, although the *CoreLP* of *DO* can be obtained from minimax formulation with the same manner as *LP* (6), each constraint is associated with a feasible flow (pure strategy) restricting the attacker's utility no smaller than that flow. Thus, a best-response flow is solved in *DO*'s oracle, which is inefficient due to no compact representation of flows. As comparison, *ConG* solves an s-t path instead.

The central challenge and novelty of *CCG* is how to efficiently generate the pure defender strategies and s-t paths to add to $\mathcal{S}'$ and $\mathcal{P}'$. Thus, we present the corresponding *ColG* and *ConG* algorithms in the following sections.

4 Column Generation (ColG)

This section concerns the *ColG* subproblem of Algorithm 1, which can be stated as follows: given an attacker feasible flow $\mathbf{f}$ defined over $\mathcal{P}'$, generate the defender pure strategy $S \in \mathcal{S}$ blocking the largest amount of flow, i.e., $S = \arg\max_{S' \in \mathcal{S}} U_d(S', \mathbf{f})$. PROPOSITION 1 shows that once ColG generates the pure strategy S already in $\mathcal{S}'$, the current equilibrium mixed strategy $\mathbf{x}$ of restricted *NFIG* is the best response against $\mathbf{f}$ of the original *NFIG*.

Proposition 1. *If* ColG *generates the pure strategy S already in $\mathcal{S}'$, the current equilibrium mixed strategy $\mathbf{x}$ of restricted* NFIG *is the best response against $\mathbf{f}$ of the original* NFIG.

We conduct thorough analysis of *ColG*'s computational complexity, and show that it is NP-hard for general graphs, while polynomial-time solvable for the tree-like networks. For the ease of reading, we put proofs of all propositions and theorems in Section A of Online Appendix[1].

Theorem 2. *The* ColG *subproblem is NP-hard.*

Algorithm 2: Greedy Algorithm for Column Generation

Initialize $S' = \emptyset$;
while $|S'| < k$ **do**
  $i^* = \arg\max_{i \notin S': S' \cup \{i\} \in \mathcal{S}} U_d(S' \cup \{i\}, \mathbf{f})$;
  $S' = S' \cup \{i^*\}$;
return S'.

Theorem 3. *If the network G is tree-like, i.e., $G - \{t\}$ is a tree, then there exists a dynamic programming algorithm able to solve the* ColG *subproblem in polynomial time.*

4.1 Convex Integer Nonlinear Formulation

Disregarding the NP-hardness of *ColG* subproblem, we provide a compact integer program, with nonlinear objective function $U_a(S, \mathbf{f})$, to solve it exactly. The program is proved to be convex (PROPOSITION 4) and hence can be solved to optimality with many commercial solvers, like KNITRO.

$$\min_{S} \quad \sum\nolimits_{p \in \mathcal{P}'} \prod\nolimits_{i \in I: i \in p} (1 - \tau_i)^{S_i} f_p \tag{7a}$$

$$\text{s.t.} \quad \sum\nolimits_{i \in I} S_i \leq k \tag{7b}$$

$$S \in \{0, 1\}^{|I|}. \tag{7c}$$

Proposition 4. *Given an attacker flow $\mathbf{f} \in \mathcal{F}$, the objective function* (7a) *is convex over decision variable $S \in [0, 1]^{|I|}$.*

4.2 Greedy Algorithm

We also propose a polynomial-time greedy algorithm of computing the approximate best response defender pure strategy, as shown in Algorithm 2 (we slightly abuse notation and let S represent the set of operated inspection stations). Starting from an empty set $S = \emptyset$, we iteratively assign a security resource on an inspection station i^* which brings the maximal marginal utility $U_d(S \cup \{i\}, \mathbf{f}) - U_d(S, \mathbf{f})$, until all k resources are assigned. The approximate pure strategy S' of Algorithm 2 and the approximate mixed strategy $\mathbf{x}'$ computed by *CCG* with *ColG* solved by Algorithm 2 are proved to achieve competitive ratios (THEOREMS 5 & 6).

Theorem 5. *The utility of S' obtained by Algorithm 2 is bounded by $U_d(S', \mathbf{f}) - U_d(\emptyset, \mathbf{f}) \geq (1 - \frac{1}{e})(U_d(S^*, \mathbf{f}) - U_d(\emptyset, \mathbf{f}))$, where S^* is the optimal solution for* ColG.

Theorem 6. *Let $(\mathbf{x}^*, \mathbf{f}^*)$ be the equilibrium solution of an* NFIG*, and let $(\mathbf{x}', \mathbf{f}')$ be the solution computed by* CCG *algorithm with* ColG *subproblem solved by greedy Algorithm 2. Then $U_d(\mathbf{x}', \mathbf{f}') - U_d(\emptyset, \mathbf{f}') \geq (1 - \frac{1}{e})(U_d(\mathbf{x}^*, \mathbf{f}^*) - U_d(\emptyset, \mathbf{f}'))$.*

5 Constraint Generation (ConG)

Given the optimal primal solution $(\mathbf{x}, \mathbf{u})$ of $LP(\mathcal{S}', \mathcal{P}')$, the *ConG* subproblem generates an s-t path $p \in \mathcal{P}$ that has maximal reduced cost defined as: $R(p) = \sum_{S \in \mathcal{S}'} x_S \Phi(S, p) - \sum_{e \in p} u_e$, i.e., $p = \max_{p' \in \mathcal{P}} R(p')$. Note that $R(p) \leq 0$ if and only if inequality (6c) associated with p is satisfied by $(\mathbf{x}, \mathbf{u})$. We show that if *ConG* generates an s-t path p such that $R(p) \leq 0$, the current equilibrium flow $\mathbf{f}$ of the restricted *NFIG* is the best response against $\mathbf{x}$ of the original *NFIG*.

Algorithm 3: *VNS* for *ConG*

```
1 Generate a random s − t path p;
2 p ← LocalSearch(p);
3 while no termination condition is met do
4     p′ ← randomly pick one from N^l(p);
5     p′ ← LocalSearch(p′);
6     if R(p′) > R(p) then p ← p′;
7 if R(p) > 0 then return p
```

Proposition 7. *If the* ConG *generates an s-t path p with non-positive maximal reduced cost, the current equilibrium flow* $\mathbf{f}$ *of the restricted* NFIG *is the attacker best response pure strategy against* $\mathbf{x}$ *of the original* NFIG.

5.1 Convex Integer Nonlinear Formulation

We first propose a compact convex integer program to solve the *ConG* problem exactly. Let binary variable $\boldsymbol{\gamma} \in \{0,1\}^{|V|+|E|}$ denote an s-t path p, such that for every node $v \in V$ and edge $e \in E$, $\gamma_v = 1$ if v is on path p and $\gamma_e = 1$ if e is on path p. We say node $v \in e$ if edge e is adjacent with v. The convex formulation is as follows:

$$\max_{\boldsymbol{\gamma}} \quad \sum\nolimits_{S\in\mathcal{S}'} x_S \prod\nolimits_{i\in S}(1-\tau_i)^{\gamma_i} - \sum\nolimits_{e\in E} u_e\gamma_e \tag{8a}$$

$$\text{s.t.} \quad \sum\nolimits_{e:s\in e} \gamma_e = 1, \sum\nolimits_{e:t\in e} \gamma_e = 1 \tag{8b}$$

$$\sum\nolimits_{e:v\in e} \gamma_e = 2\gamma_v \quad \forall v \in V, v \neq s,t \tag{8c}$$

$$\gamma_e \leq \gamma_v, \gamma_e \leq \gamma_{v'} \quad \forall e \in E : e = (v,v') \tag{8d}$$

$$\gamma_s = \gamma_t = 1 \tag{8e}$$

$$\boldsymbol{\gamma} \in \{0,1\}^{|V|+|E|}. \tag{8f}$$

The convexity of the objective (8a) can be easily verified in the same way of proving PROPOSITION 4. Eqs.(8b)–(8e) ensure that $\boldsymbol{\gamma}$ denotes a feasible s-t path p in the sense that: i) s and t are on p (8e); ii) exactly one edge adjacent with s (t) is on p (8b); iii) if node v, except s and t, is on p, then exactly two edges adjacent with v are on p (8c); and iv) if node v is not on p, then no edge adjacent with v is on p (8c).

Notice that the program (8) without constraint (8d) can also solve the *ConG* subproblem exactly. The integer programs are usually solved with the popular *branch and bound (B&B)* framework and the efficiency of *B&B* depends on the quality of the upper bound obtained by solving the linear relaxation of the program (8). Thus, inequality (8d) is proposed to tighten the upper bound by cutting off some fractional solutions with too high objective values, such an example is $\boldsymbol{\gamma}$ with: i) $\gamma_e = 1$ for $e' = (s,v')$, $e'' = (v'',t)$; ii) $\gamma_v = 0.5$ for v' and v''; iii) $\gamma_v = 0$, $\gamma_e = 0$ for all other nodes and edges.

5.2 Variable Neighborhood Search (VNS)

Although the convex program (8) can solve the *ConG* subproblem exactly, it still cannot scale up to solve large scale networks. Observe, however, in each iteration of *CCG* algorithm, we actually need not find the path with maximal reduced cost; rather, any path with a large enough (positive) reduced cost would suffice. Thus, a fast heuristic search for generating such paths is satisfactory for most iterations, except when the heuristic is unable to find a path with positive reduced cost. In this case we can call the convex program (8). To this end, we propose a heuristic search algorithm (Algorithm 3) based on *Variable Neighborhood Search (VNS)* framework [Hansen and Mladenović, 2001].

Algorithm 4: *LocalSearch* (p)

```
1 Repeat
2     p* ← arg max_{p′∈N^l(p)} R(p′);
3     if R(p*) > R(p) then p ← p*;
4     else return p;
```

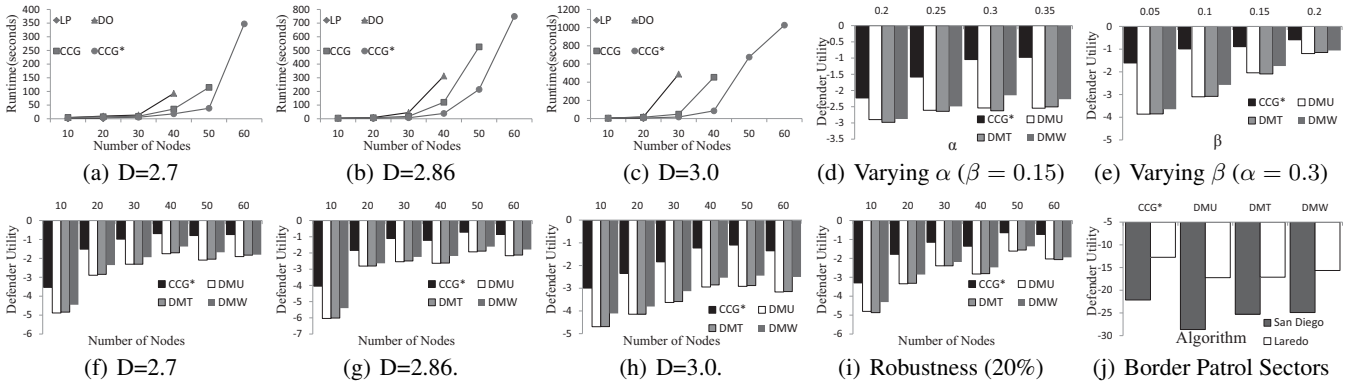
Two key components of *VNS* are: i) $\mathcal{N}^l(p)$, denoting the set of neighbor paths of p within distance l; and ii) *LocalSearch* (p) to find a local optimum starting from p (Algorithm 4). We say that path p' is a neighbor path within distance l of p if p' can be obtained by replacing a v'-v'' sub-path of p with another v'-v'' path of length no larger than l. The *LocalSearch* (p) iteratively searches for a neighbor path p^* with maximal reduced cost (Line 2) and updates current incumbent p if p^* is a better solution (Line 3), until a local optimum is obtained whose reduced cost is higher than all its neighbor paths.

After initialized with an arbitrary local optimum (Lines 1–2), the *VNS* randomly picks a neighbor path p' of p and applies *LocalSearch* (p') to find a local optimum (Lines 4–5). Afterwards, the incumbent p is updated. This loop is repeated until a termination condition is met: i) current incumbent p has a positive reduced cost (Line 7); or ii) for c_{max} consecutive iterations, the incumbent p is not updated.

6 Experimental Evaluation

We evaluate the performance of our approach through extensive experiments. We use CPLEX (version 12.6) to solve linear programs and KNITRO (version 9.0.0) to solve nonlinear programs. All computations were performed on a 64-bit PC with 16 GB RAM and a quad-core 3.4 GHz processor. All values are averaged over 40 instances unless otherwise specified. All random planar graphs are generated by the Waxman geographical model (*WG*) suitable for modeling highway networks [Waxman, 1988]. In the *WG* model, $|N|$ nodes (cities) are placed in a rectangular domain uniformly, and each pair of nodes at Euclidean distance d is jointed by an edge with probability $p = e^{-\lambda d}$, where λ is a constant adjusted to achieve the desirable average degree D. The two nodes with maximal Euclidean distance are set as the *source* and *sink* nodes of graph G. By default, the instances are parameterized as follows: the number of inspection stations $|I| = \lfloor \alpha(|V|+|E|) \rfloor$ and the number of resources $k = \lfloor \beta(|V|+|E|) \rfloor$, where α and β are tunable parameters. All inspection stations are randomly placed on the graph. The edge capacity c_e is randomly chosen in $[0, 10]$, and inspection probability $\tau_i \sim [0.4, 0.6]$. The l and c_{max} are set as 4 and $|V|$ respectively in *VNS*.

We compare the scalability of three versions of our algorithms: i) *CCG*: Algorithm 1 with *ColG* and *ConG* solved by programs (7) and (8); ii) *CCG**: Algorithm 1, where *ColG* and *ConG* are first solved by greedy methods (Algorithms 2 and 3), and then call the programs (7) and (8) when greedy



(a) D=2.7 (b) D=2.86 (c) D=3.0 (d) Varying α ($\beta = 0.15$) (e) Varying β ($\alpha = 0.3$)

(f) D=2.7 (g) D=2.86. (h) D=3.0. (i) Robustness (20%) (j) Border Patrol Sectors

Figure 2: Runtime: 2(a)–2(c), Defender Utility: 2(f)–2(h), Vary α and β: 2(d)–2(e), Robustness: 2(i), Real Application: 2(j).

methods return a pure strategy S already in $\mathcal{S}'$ or path p with $R(p) \leq 0$; iii) *LP*: the linear program (6), with the benchmark *double oracle (DO)* where two oracles solve the best response pure strategies (S, $\mathbf{f}$) for both players [Jain *et al.*, 2011].

To evaluate the solution quality of our approach, we implement three heuristic strategies as baselines. The quality of a solution $\mathbf{x}$ is measured by $U_d(\mathbf{x}, y(\mathbf{x}))$. The baseline mixed strategies are: i) *DMU* with marginal coverage probability of each inspection station equal to $\frac{k}{|I|}$; ii) *DMT* where marginal coverage probability of each inspection station is normalized inspection probability; iii) *DMW* where marginal coverage probability of each inspection station is normalized inspection weight. The inspection weight w_i of station i is defined as the maximum amount of flow that i can interdict, i.e., the production of τ_i and the maximum amount of flow passing through that edge or node; Given the marginal coverage probabilities, the defender mixed strategies are generated by the *Combo Sampling* algorithm [Tsai *et al.*, 2010].

Scalability Analysis. We compare the scalability of our algorithms on *WG* graphs with varying degrees: $D = 2.86$ which is the mean degree of road network [Gastner and Newman, 2006], and $D \in \{2.7, 3.0\}$. $\alpha = 0.3$ and $\beta = 0.15$. The results are depicted in Figures 2(a)–2(c), where *DO* cannot scale up to 50 nodes for $D = 2.7$ and $D = 2.86$ and 40 nodes for $D = 3.0$ with runtime cap of 1800 seconds, while *LP* cannot scale up to 30 nodes with memory cap of 8GB. The results show that although our approaches (*CCG*,*CCG**) involve nonlinear programs (7) and (8), they still outperform *LP* and *DO* significantly due to the compact representation. Besides, *CCG** also outperforms *CCG* a lot which shows that the greedy methods (Algorithms 2 and 3) can obtain good enough solutions. Especially *CCG** can scale up to realistic-sized networks with 60 nodes and over *40* inspection stations and *20* resources, while the number of tactical checkpoints operated in the Southern border is 39 in 2009 [GAO, 2009].

Solution Quality Analysis. We also compare the solution qualities of our algorithms with three baseline strategies on *WG* graphs with varying degrees ($\alpha = 0.3$ and $\beta = 0.15$), demonstrated by Figures 2(f)–2(h). It is clear that *CCG** outperforms these baselines significantly. Besides, the baseline *DMW* works best among the three baselines since the inspection weights well measure the capability of inspection stations. Note that the numbers of inspection stations and resources are proportional with the network size, and hence there exists no monotone relationship between utility and network size.

Varying Values of α and β. Figures 2(d)–2(e) show the solution qualities of *CCG** and three baselines on *WG* graphs ($|V| = 40$, $D = 2.86$) with varying values of α and β, from which we can see: i) with fixed β, increasing α improves the defender utility for more available inspection stations; ii) with fixed α, increasing β also improves the defender utility for more available resources. Moreover, *CCG** outperforms these baselines significantly for all tested values of α and β.

Robustness. In reality, the attacker might not have perfect observation on the defender strategy. Thus, we compare the solution qualities of *CCG** and baselines on *WG* graphs ($|V| = 40$, $D = 2.86$, $\alpha = 0.3$ and $\beta = 0.15$) with the attacker's estimation $\mathbf{x}'$ of the defender strategy $\mathbf{x}$ randomly generated by: $\mathbf{x}' = \tilde{\mathbf{x}}/|\tilde{\mathbf{x}}|$ where $\tilde{x}_S \sim x_S \cdot [1-\delta, 1+\delta]$. The solution quality of $\mathbf{x}$ is measured by $U_d(\mathbf{x}, \mathbf{f}')$ with $\mathbf{f}'$ being the attacker best response against $\mathbf{x}'$. The result is depicted in Figure 2(i) with $\delta = 20\%$, and the result shows that our approach *CCG** is robust enough and outperforms baselines significantly under a high level of observation uncertainty.

Application on Southern Border Network. We also conduct experiments on two sectors in Southern Border Patrol: Laredo and San Diego sectors. Please see Section C of Online Appendix[1] for details of these sectors. The results are depicted in Figure 2(j), showing that our approach significantly outperforms the baseline methods for the realistic networks.

7 Conclusion

This paper studies the problem of optimally interdicting an illegal network flow. We introduce a novel Stackelberg game model called *NFIG*, and propose a *Column and Constraint Generation (CCG)* algorithm to solve it. The computational complexity of its *ColG* subproblem is analyzed. Several novel algorithms are provided to solve the two subproblems including convex optimization, $1 - \frac{1}{e}$ approximation method and a heuristic search algorithm. Experimental evaluation shows that *CCG* scales up to realistic-sized networks and significantly outperforms existing methods and heuristics.

Acknowledgements

This research is supported by NRF2015NCR-NCR003-004, the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative. The authors would like to thank the anonymous reviewers for their helpful comments.

References

[Altner *et al.*, 2010] Douglas S Altner, Özlem Ergun, and Nelson A Uhan. The maximum flow network interdiction problem: Valid inequalities, integrality gaps, and approximability. *Operations Research Letters*, 38(1):33–38, 2010.

[An *et al.*, 2013] Bo An, Matthew Brown, Yevgeniy Vorobeychik, and Milind Tambe. Security games with surveillance cost and optimal timing of attack execution. In *AAMAS*, pages 223–230, 2013.

[Ball *et al.*, 1989] Michael O Ball, Bruce L Golden, and Rakesh V Vohra. Finding the most vital arcs in a network. *Operations Research Letters*, 8(2):73–76, 1989.

[Beittel, 2015] June S. Beittel. *Mexico: Organzied Crime and Drug Trafficking Organizations*. Congressional Research Service, 2015.

[Church *et al.*, 2004] R.L. Church, M.P. Scaparra, and R.S. Middleton. Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers*, 94(3):491–502, 2004.

[DOJ, 2010] DOJ. *National Drug Threat Assessment 2010.* U.S. Department of Justice National Drug Intelligence Center, 2010.

[Gan *et al.*, 2015] Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. Security games with protection externalities. In *AAAI*, pages 914–920, 2015.

[GAO, 2009] GAO. *BORDER PATROL: Checkpoints Contribute to Border Patrols Mission, but More Consistent Data Collection and Performance Measurement Could Improve Effectiveness.* U.S. Government Accountability Office, 2009.

[Gastner and Newman, 2006] Michael T Gastner and Mark EJ Newman. The spatial structure of networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 49(2):247–252, 2006.

[Guo *et al.*, 2016] Qingyu Guo, Bo An, Yevgeniy Vorobeychik, Long Tran-Thanh, Jiarui Gan, and Chunyan Miao. Coalitional security games. In *AAMAS*, 2016.

[Hansen and Mladenović, 2001] Pierre Hansen and Nenad Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.

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

[Jain *et al.*, 2011] Manish Jain, Dmytro Korzhyk, Ondrej Vanek, Vincent Conitzer, Michal Pechoucek, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *AAMAS*, pages 327–334, 2011.

[Jain *et al.*, 2013] Manish Jain, Vincent Conitzer, and Milind Tambe. Security scheduling for real-world networks. In *AAMAS*, pages 215–222, 2013.

[Kibble, 2009] Kumar C. Kibble. *Law enforcement responses to mexican drug cartels.* US Department of Homeland Security, 2009.

[Malik *et al.*, 1989] Kavindra Malik, AK Mittal, and Santosh K Gupta. The k most vital arcs in the shortest path problem. *Operations Research Letters*, 8(4):223–227, 1989.

[McMahan *et al.*, 2003] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *ICML*, pages 536–543, 2003.

[Nlkaido, 1954] Hukukane Nlkaido. On von neumann's minimax theorem. *Pacific J. Math*, 4:65–72, 1954.

[Pan *et al.*, 2001] F Pan, W Charlton, and DP Morton. Stochastic network interdiction of nuclear material smuggling. *Network Interdiction and Stochastic Integer Programming*, pages 1–19, 2001.

[Pan, 2005] F. Pan. *Stochastic network interdiction: models and methods*. PhD thesis, University of Texas, Austin, 2005.

[Rosenberg, 2015] Chuck Rosenberg. *2015-National Drug Threat Assessment Summary.* U.S. Department of Justice Drug Enforcement Administration, 2015.

[Steinrauf, 1991] Robert L Steinrauf. Network interdiction models. Technical report, DTIC Document, 1991.

[Tambe, 2011] Milind Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[Tsai *et al.*, 2010] Jason Tsai, Zhengyu Yin, Jun-young Kwak, David Kempe, Christopher Kiekintveld, and Milind Tambe. Urban security: Game-theoretic resource allocation in networked domains. In *AAAI*, pages 881–886, 2010.

[Vanek *et al.*, 2012] Ondrej Vanek, Zhengyu Yin, Manish Jain, Branislav Bosanský, Milind Tambe, and Michal Pechoucek. Game-theoretic resource allocation for malicious packet detection in computer networks. In *AAMAS*, pages 905–912, 2012.

[Wang *et al.*, 2016] Zhen Wang, Yue Yin, and Bo An. Computing optimal monitoring strategy for detecting terrorist plots. In *AAAI*, 2016.

[Waxman, 1988] Bernard M Waxman. Routing of multipoint connections. *Selected Areas in Communications, IEEE Journal on*, 6(9):1617–1622, 1988.

[Wood, 1993] R Kevin Wood. Deterministic network interdiction. *Mathematical and Computer Modelling*, 17(2):1–18, 1993.

[Yin *et al.*, 2014] Yue Yin, Bo An, and Manish Jain. Game-theoretic resource allocation for protecting large public events. In *AAAI*, pages 826–834, 2014.

[Yin *et al.*, 2015] Yue Yin, Haifeng Xu, Jiarui Gan, Bo An, and Albert Xin Jiang. Computing optimal mixed strategies for security games with dynamic payoffs. In *IJCAI*, pages 681–688, 2015.