Operations Research Letters 44 (2016) 114–120

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

On the power of randomization in network interdiction

Dimitris Bertsimas, Ebrahim Nasrabadi *, James B. Orlin

Sloan School of Management and Operations Research Center, Bldg. E40-147, Massachusetts Institute of Technology, Cambridge, MA 02139, United States

ARTICLE INFO

Article history:
Received 3 November 2013
Received in revised form
9 November 2015
Accepted 10 November 2015
Available online 2 December 2015

Keywords:
Network flows
Interdiction
Game theory
Approximation

ABSTRACT

In this paper, we introduce the *randomized network interdiction* problem that allows the interdictor to use randomness to select arcs to be removed. We model the problem in two different ways: *arc-based* and *path-based* formulations, depending on whether flows are defined on arcs or paths, respectively. We present insights into the modeling power, complexity, and approximability of both formulations.

1. Introduction

Network flows have applications in a wide variety of contexts (see, e.g., [1]). In some applications, it is useful to consider the perspective of someone who wants to restrict flows in a network. For example, law enforcement wants to inhibit the flow of illegal drugs. Water management experts want to control flows to avoid floods. Health agencies need to protect against contagion. Here, it is important to consider the problem of limiting flows in the network from the perspective of an *interdictor*, who is capable of limiting capacity in arcs or eliminating arcs. Such problems have been applied in many application areas such as military planning [21], controlling infections in a hospital [3], controlling floods [15], protecting critical infrastructures [13,18], and drug interdiction [19].

Motivated by the above mentioned applications, *network interdiction problems* have been well studied in the literature (see, e.g., [4,7,9–11,16,17,23]). In this paper, we focus on the basic model of network interdiction, where the interdiction of an arc requires exactly one unit of resource: a *flow player* attempts to maximize the amount of material transported through a capacitated network, while an *interdictor* tries to limit the flow player's achievable value by interdicting a certain number, say Γ, of arcs. This problem is also known as the Γ-most vital arcs problem (see, e.g., [15]). Wollmer [22] presents a polynomial time algorithm for solving this problem on planar graphs. On general networks, Wood [23] and Phillips [14] independently show that the problem is strongly NP-complete. Burch et al. [6] develop approximation algorithms for general instances of the network interdiction problem. In particular, they consider the case where each arc has a removal cost and its capacity can be reduced partially or completely, and there is a limited budget to attack the network and reduce the arc capacities. They provide a polynomial-time algorithm, based on a linear relaxation of an integer optimization formulation, that leads to either an approximation or pseudo-approximation result for the resulting problem.

Network interdiction can be viewed as a game between the interdictor and the flow player. This problem assumes the interdictor moves first and then the flow player determines a maximum flow in the remaining network. A closely related problem arises when a flow must be routed before arcs are removed. In this case, the flow player might be interested to find solutions which are robust against any failure of arcs. Aneja et al. [2] study this problem in a path-based formulation and show that the resulting problem is solvable in polynomial-time for the special case of $\Gamma = 1$. This problem was further expanded to an arc-based formulation by Bertsimas et al. [5], who introduce the concepts of robust and adaptive maximum flows. They establish structural and computational results for both the robust and adaptive maximum flow problems and their corresponding minimum cut problems.

Our contribution. The network interdiction problem addresses a minimax objective against a flow player, which selects adaptively a flow after observing the removed arcs. This problem requires the interdictor to choose a specific *pure strategy*. We propose a new modeling framework that permits the interdictor to use randomness to choose arcs. More precisely, the interdictor assigns a probability to each pure strategy and selects a pure strategy

* Corresponding author.
E-mail addresses: dbertsim@mit.edu (D. Bertsimas), enasrabadi58@gmail.com (E. Nasrabadi), jorlin@mit.edu (J.B. Orlin).

http://dx.doi.org/10.1016/j.orl.2015.11.005
0167-6377/

randomly according to these probabilities. We refer to the resulting problem as the *randomized* network interdiction problem. This provides a more realistic model for various applications such as protecting critical infrastructures against terrorism or enemy's attacks. We also consider a further modification that requires the flow player to send flow on paths, rather than the more typical arc-based model. We present results on the modeling power, complexity, and approximability of both arc-based and path-based formulations. In particular, we prove that $Z_{\mathrm{NI}}/Z_{\mathrm{RNI}} \le \Gamma + 1$, $Z_{\mathrm{NI}}/Z^{\mathrm{Path}}_{\mathrm{RNI}} \le \Gamma + 1$, $Z_{\mathrm{RNI}}/Z^{\mathrm{Path}}_{\mathrm{RNI}} \le \Gamma$, where Z_{NI}, Z_{RNI}, and $Z^{\mathrm{Path}}_{\mathrm{RNI}}$ are the optimal values of the network interdiction problem and its randomized versions in arc-based and path-based formulations, respectively. We also show that these bounds are tight. Further, we provide a $(\Gamma + 1)$-approximation for Z_{NI}, a Γ-approximation for Z_{RNI}, and a $\left(1 + \lfloor \Gamma/2\rfloor \cdot \lceil \Gamma/2\rceil/(\Gamma + 1)\right)$-approximation for $Z^{\mathrm{Path}}_{\mathrm{RNI}}$.

2. Network interdiction

Let $G = (V, E)$ be a directed graph with *node set* V and *arc set* E. Each arc $e \in E$ has a *capacity* $u_e \in \mathbb{R}_+$ setting an upper bound on the amount of flow on arc e. There are two specific nodes, a *source* s and a *sink* t. W denote an arc e from a node v to a node w by $e := (v, w)$. We use $\delta^+(v) := \{(v, w) \in E \mid w \in V\}$ and $\delta^-(v) := \{(w, v) \in E \mid w \in V\}$ to denote the sets of arcs leaving node v and entering node v, respectively. We assume without loss of generality that there are no arcs into s and no arcs out of t, that is, $\delta^-(s) = \delta^+(t) = \emptyset$.

2.1. Arc-based formulation

An *s-t-flow* (or simply a *flow*) x is a function $x : E \to \mathbb{R}_+$ which assigns a nonnegative value to each arc so that $x_e \le u_e$ for each $e \in E$, and in addition for each node $v \in V \setminus \{s, t\}$, the following *flow conservation constraint* holds:

$$\sum_{e\in\delta^-(v)} x_e - \sum_{e\in\delta^+(v)} x_e = 0.$$

We refer to x_e as the *flow* on arc e. We denote the set of all s-t-flows by $\mathcal{X}$. The *value* $\mathrm{Val}(x)$ of an s-t flow x is the net flow into t, that is, $\mathrm{Val}(x) := \sum_{e\in\delta^-(t)} x_e$. In the *maximum flow* problem (also referred to as the *nominal* problem), we seek an s-t flow x with maximum value $\mathrm{Val}(x)$.

We next assume that there is an interdictor, who wants to reduce the capacity of the network. Suppose that the interdictor is able to eliminate Γ $(1 \le \Gamma \le |E|)$ arcs in the network. The *network interdiction* problem is to find the Γ arcs whose removal from the network minimizes the maximum amount of flow that can be sent to the sink. To formulate this problem, we let

$$\Omega := \left\{\mu = (\mu_e)_{e\in E} \in \{0, 1\}^{|E|} \mid \sum_{e\in E} \mu_e = \Gamma\right\}$$

denote the set of all possible scenarios, that is, the set of all subsets of Γ arcs. The binary variable μ_e indicates whether or not arc e is to be removed, depending on whether $\mu_e = 1$ or $\mu_e = 0$, respectively. Given $\mu \in \Omega$, we denote by $E(\mu) := \{e \in E \mid \mu_e = 1\}$ the set of removed arcs and by $F(\mu) := \{e \in E \mid \mu_e = 0\}$ the set of available arcs after removing the arcs in the scenario μ. We also denote by $G(\mu) = (V, F(\mu))$ a network with arc set $F(\mu)$.

The network interdiction problem is formulated as

$$\begin{aligned} Z_{\mathrm{NI}} := \min_{\mu\in\Omega} \quad \max \quad & \mathrm{Val}(x) \\ \text{s.t.} \quad & x \in \mathcal{X}, \\ & x_e = 0 \quad \forall e \in E(\mu). \end{aligned} \tag{1}$$

This problem determines the interdictor's best choice, assuming the flow player is in a position to select a maximum flow after observing the removed arcs. In many applications, the flow player has to make a decision before the interdictor selects her strategy. Here, the flow player might be interested in those solutions that are robust against any possible scenario. This leads to the following problem, referred to as the *adaptive maximum flow* problem:

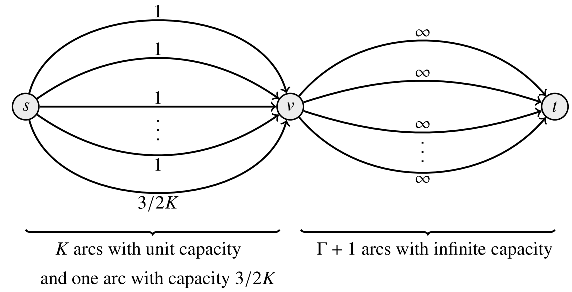

Fig. 1. Illustration of the difference between the network interdiction problem and maximum adaptive flow problem. The numbers on the arcs indicate the capacities. We have $Z_{\mathrm{NI}} = K + 1 - \Gamma$, while $Z_{\mathrm{ADP}} = \frac{5K}{2(\Gamma+1)}$.

$$Z_{\mathrm{ADP}} := \max_{x\in\mathcal{X}} \min_{\mu\in\Omega} \quad f(\mu, x), \tag{2}$$

where $f(\mu, x)$ is the maximum amount of flow that the flow player can push through the network with respect to the flow x if scenario μ is selected. Mathematically, the function f is given by

$$\begin{aligned} f(\mu, x) := \max \quad & \mathrm{Val}(y) \\ \text{s.t.} \quad & y \in \mathcal{X}, \\ & 0 \le y_e \le x_e \quad \forall e \in F(\mu) \\ & y_e = 0 \quad \forall e \in E(\mu). \end{aligned} \tag{3}$$

This problem is introduced by Bertsimas et al. [5], who establish structural properties and complexity results for the problem. In particular, they show that the adaptive maximum flow problem is NP-hard using a reduction from the network interdiction problem.

Note that $Z_{\mathrm{ADP}} \le Z_{\mathrm{NI}}$. This follows from the fact that Problem (1) is equivalent to

$$Z_{\mathrm{NI}} = \min_{\mu\in\Omega} \max_{x\in\mathcal{X}} \quad f(\mu, x). \tag{4}$$

To compare the difference between the network interdiction problem and the adaptive maximum flow problem, we consider a network with three nodes s, v, and t as shown in Fig. 1. There are K arcs with unit capacity and one arc with capacity $3/2K$ from s to t and there are $\Gamma+1$ arcs with infinite capacity from v to t. Let $\Gamma \ge 2$ and K be enough large. It is easy to see that $Z_{\mathrm{NI}} = K + 1 - \Gamma$, while $Z_{\mathrm{ADP}} = \frac{5K}{2(\Gamma+1)}$. Hence, $Z_{\mathrm{NI}}/Z_{\mathrm{ADP}} = \frac{2(\Gamma+1)(K+1-\Gamma)}{5K}$, and the ratio becomes close to $2(\Gamma + 1)/5$ when K gets large. An interesting question is: How large can $Z_{\mathrm{NI}}/Z_{\mathrm{ADP}}$ be in general? We will show later that this ratio is bounded by $\Gamma + 1$ and this bound is tight.

2.2. Path-based formulation

So far, we have considered flows in an arc-based formulation. We next focus on an alternative formulation of flows, in which the flow player must specify paths on which to route the material. This leads to a different model for the adaptive problem.

Let $\mathcal{P}$ denote the set of all s-t-paths (i.e., paths from s to t). For $P \in \mathcal{P}$, we write $e \in P$ to indicate that arc $e \in E$ lies on P. An s-t-(path-based) flow is a function $x : \mathcal{P} \to \mathbb{R}_+$ that assigns a nonnegative value to each path so that the total flow on each arc does not exceed the capacity of the arc, that is,

$$\sum_{P\in\mathcal{P}:e\in P} x_P \le u_e \qquad \forall e \in E.$$

The value of x is the sum of the flows on the paths, i.e., $\mathrm{Val}(x) = \sum_{P\in\mathcal{P}} x_P$. We use $\mathcal{X}_P$ to denote the set of all s-t-path-based flows.