2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

On the Optimal Interdiction of Transportation Networks

Tianyun Zhang and Makan Fardad

Abstract— **We consider the optimal interdiction problem in transportation networks as a game in which an attacker acts as the player who goes first and, subject to budget constraints, fails nodes (partially or fully) at time zero so as to maximize the total travel time of the mass. A centralized network operator then acts as the player who goes second and, subject to the system's dynamics, routes the mass so as to minimize its total travel time. We prove that the attacker's best action is to find the most consequential nodes and employ his resources to fail them fully, so that the optimal attack is both sparse and binary. We then propose an algorithm to numerically solve the optimal interdiction problem, and demonstrate the utility of our approach through illustrative examples.**

Index Terms— **Cascading failures, flow networks, linear programming, network interdiction, optimization, sparsity, traffic networks.**

I. Introduction

Transportation networks, also known as flow networks, are networks in which mass enters through source nodes and on-ramps, is routed through nodes/cells and directed links, and is removed at sink nodes and off-ramps. The flow of mass is subject to (i) conservation of mass constraints, and (ii) link capacity constraints. Traffic networks, water supply networks, and (routing of data packets in) computer networks are all examples of transportation networks.

The cell transmission model of mass transfer developed by Daganzo [1], [2] captures complex traffic behavior and transient phenomena, such as congestion effects and the propagation of shocks. Ziliaskopoulos [3] uses the cell transmission model to formulate the optimum traffic assignment problem as a linear program. In an influential sequence of papers [4]–[7], Como *et al.* and Savla *et al.* analyze the robustness and resilience of transportation networks under decentralized routing. In particular, they propose routing policies that depend only on local information and maximally delay congestion effects under adversarial perturbations to the capacities of cells. More recently, [8] considers the problem of designing optimal routing policies in a network where a fraction of vehicles will choose to ignore these policies and act selfishly. The paper also studies network resilience by finding changes in said fraction that lead to a given cell reaching its jam mass and thus failing. Similar to our work in its use of a min-max formulation, [9] designs robot trajectories in the presence of undetectable attacks.

Financial support from the National Science Foundation under awards CAREER CMMI-1750531 and EAGER ECCS-1545270 is gratefully acknowledged.

T. Zhang and M. Fardad are with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244. E-mails: tzhan120@syr.edu, makan@syr.edu.

In this paper, we study interdiction or attack on transportation networks, which for the sake of concreteness we consider to be highway traffic networks. In this context it is of interest to find a small set of cells whose failure at time zero, amplified and propagated by the system's dynamics, maximally disrupts the flow of traffic. This problem is combinatorial in nature and intractable in general. Our work follows [10] in formulating the optimal interdiction problem as a min-max optimization problem and subsequently employing duality to transform it to a standard bilinear optimization problem.

We demonstrate that even without an explicit promotion of sparsity in the formulation, the solution to the optimal interdiction problem is both sparse and binary. The solution is sparse in the sense that the attacker's best use of resources is to find the small set of most consequential cells in the network, and it is binary in the sense that the attacker's best choice is to fail these cells fully (as opposed to partially).

Furthermore, motivated by the block coordinate gradient descent (BCGD) and block coordinate descent (BCD) algorithms [11], we solve the bilinear problem by iteratively updating one set of variables through a gradient-based step and then finding the globally optimal solution in the other set of variables. Our numerical experiments demonstrate that our approach performs better in comparison with methods reported in earlier work [10], and we find the globally optimal solution in the small networks that we tested and for which the global optimum could be verified through exhaustive search.

The rest of the paper is organized as follows. In Section II we introduce the standard cell transmission model and also our augmentation of it, which allows for the irreversible failure of nodes/cells. In Section III we formulate the optimal interdiction problem and follow [10] in employing duality to reformulate the problem as a bilinear program. In Section IV we prove the sparsity and binary nature of the solution of the optimal interdiction problem. In Section V we propose a numerical algorithm to solve the bilinear optimization problem. In Section VI we use examples to illustrate the effectiveness of our method.

II. Dynamic Model of Transportation Networks & Informal Statement of Optimal Interdiction Problem

In this section we first introduce the cell transmission model. We augment the model in a way that allows for the irreversible failure of cells through two mechanisms: being attacked by an adversary and reaching the jam threshold through the accumulation of mass. We then discuss a meaningful formulation of the optimal interdiction problem subject to attacker resource constraints.

The network is characterized by a directed graph, where we think of nodes as cells and of edges as allowing for flows between neighboring cells. (In this work we use the words cell and node interchangeably.) The temporal dynamics for the cell transmission model are governed in part by [3], [12]

$$x_i(t) = x_i(t-1) + y_i(t-1) - z_i(t-1) \tag{1a}$$

(conservation of mass on cell i)

$$y_i(t) = v_i(t) + \textstyle\sum_j f_{ji}(t) \tag{1b}$$

(total inflow to cell i = on-ramp flow + rerouted flow)

$$z_i(t) = w_i(t) + \textstyle\sum_j f_{ij}(t) \tag{1c}$$

(total outflow from cell i = off-ramp flow + rerouted flow)

for every i and t, where x_i, y_i, and z_i, respectively denote the mass (i.e., number of vehicles) on, the inflow to, and the outflow from, cell i; v_i, w_i respectively denote the mass entering the network from on–ramp, and leaving the network from off–ramp, corresponding to cell i; f_{ij} denotes the mass routed from cell i to adjacent cell j. The dynamics are additionally constrained to [3], [12]

$$x_i(t) \geq 0,\ v_i(t) \geq 0,\ w_i(t) \geq 0,\ f_{ij}(t) \geq 0 \tag{2a}$$

(positivity of mass)

$$y_i(t) \leq \kappa_i,\ z_i(t) \leq \kappa_i \tag{2b}$$

(inflow, outflow cannot exceed flow-capacity of cell)

$$y_i(t) \leq \phi_i - x_i(t),\ z_i(t) \leq x_i(t) \tag{2c}$$

(inflow cannot exceed remaining mass-capacity of cell, outflow cannot exceed mass on cell)

with the further restrictions that $f_{ij}(\cdot) = 0$ if cells i and j are not adjacent, $w_i(\cdot) = 0$ if cell i does not have an off–ramp, and $v_i(\cdot)$ specified *a priori*. The parameter ϕ_i denotes the amount of mass that results in cell i being jammed. Inequalities (2b)–(2c) result from piecewise linear "supply" and "demand" functions [12].

We assume that all mass enters the network from on-ramps and possibly a source cell and that it leaves the network through a sink cell. The source and sink cells have very large capacities. Without loss of generality, we take the cell with the lowest index to be the source (when a source cell is present) and the cell with the largest index to be the sink.

As in [10], we further augment the dynamics (1)–(2) with

$$z_i(t) \leq \phi_i - x_i(t), \tag{3}$$

$$\kappa_i \in \{0, \psi_i\}, \tag{4}$$

where ψ_i denotes the maximum possible amount of mass that can flow in or out of cell i during one time step, and ϕ_i is the same jam mass as before. The constraints in (3)–(4), together with (1)–(2), capture two methods by which a cell irreversibly fails:

- at time 0 an attacker reduces the capacity of cell i to zero, $\kappa_i = 0$;
- at some time $t_0 \geq 1$ and as a result of the network's dynamics the accumulated mass on cell i reaches the jam threshold, $x_i(t_0) = \phi_i$.

In both scenarios, once a cell has failed *no mass can either enter or leave it* thereafter.

We assume that the attacker operates under a limited budget e and that the ith element of the vector c characterizes the cost for the attacker of reducing κ_i from ψ_i to 0. This means that the attacker is subject to the budget constraint

$$c^T(\mathbb{1} - \kappa/\psi) \leq e,$$

where $\mathbb{1}$ is the column vector of all ones and division by a vector is element-wise. The above inequality is equivalent to

$$q^T\kappa \geq d \tag{5}$$

with $q := c/\psi$ and $d := \mathbb{1}^T c - e$.

Defining $x(t)$ as the vector whose ith entry is $x_i(t)$, with similar definitions for vectors $y(t), z(t), f(t), v(t), w(t), \kappa$, and taking $l = [1, \ldots, 1, 0]^T$ so that $l^T x(t)$ equals the total mass at time t on all cells except the sink cell, our main problem in this work can be (informally) stated as follows.

Optimal Interdiction Problem: Given the temporal evolution model and failure constraints described by (1)–(5), for the time horizon $0, 1, \ldots, \bar{t}$ *find a sparse set of cells whose failure at time* 0 *maximizes the total travel time* $\sum_{t=0}^{\bar{t}} l^T x(t)$.

We formulate the optimal interdiction problem as a game in which an attacker acts as the player who goes first and, subject to the constraints (4)–(5), fails the most critical nodes at time 0 so as to *maximize* the total travel time of the mass. A centralized network operator or trajectory planner then acts as the player who goes second and, subject to the constraints (1)–(3), routes the mass so as to *minimize* its total travel time.

The optimal interdiction problem can therefore be thought of as

$$\underset{\kappa}{\text{maximize}} \quad \min_{x,y,z,f,w} \left(\textstyle\sum_{t=0}^{\bar{t}} l^T x(t)\right) \tag{6}$$

where the inner minimum is taken over the constraints (1)–(3), given initial conditions $x(0), y(0), z(0)$ and prescribed on-ramp flows; the outer maximization is performed over the constraints (4)–(5).

Based on our informal statement of the optimal interdiction problem above, one may be inclined to formulate (6) in a way that *forces* a sparse set of failures. This could be done for example by including in the objective function a sparsity-promoting term that counts and penalizes the number of nonzero entries in the vector $\psi - \kappa$. We will demonstrate in Section IV, however, that (6) indeed has sparse solutions without the need for an explicit promotion of sparsity in its formulation.

III. Formal Statement of Optimal Interdiction & Its Reformulation as Bilinear Program

In this section we mathematically formulate optimal interdiction as a max-min problem with a linear objective and linear constraints. We then employ duality to obtain an equivalent formulation as a maximization problem with a bilinear objective and linear constraints.

Stacking the optimization variables $x(t), y(t), z(t), f(t), w(t)$ into the vector $\mathrm{u}(t)$, and stacking $\mathrm{u}(1), \mathrm{u}(2), \ldots, \mathrm{u}(\bar{t})$ to form the vector u, we formally state the optimal interdiction problem as

$$\begin{array}{lll} \underset{\kappa}{\text{maximize}} & \underset{u}{\text{minimize}} & p^T u \\ \text{subject to} & & Au = b, \quad Gu \leq H\kappa + h \\ & & 0 \leq \kappa \leq \psi, \quad q^T\kappa \geq d \end{array} \tag{7}$$

where A, b, G, H, h, p are appropriately defined matrices and vectors so that $Au = b$ and $Gu \leq H\kappa + h$ are compact representations respectively of the equality constraints (1) and inequality constraints (2)–(3). Here, for each i we have relaxed the constraint $\kappa_i \in \{0, \psi_i\}$ to $0 \leq \kappa_i \leq \psi_i$. We will demonstrate in the next section that despite lacking any explicit promotion of sparsity in the formulation and despite the relaxation of all binary constraints, the solution to (7) is in fact sparse and that all but (at most) one of the κ_i belong to $\{0, \psi_i\}$. This is the main theoretical contribution of the present work.

We next employ duality as in [10] to turn the max-min problem (7) into a standard maximization problem. Problem (7) is equivalent to

$$\begin{array}{ll} \underset{\kappa,\lambda,\nu}{\text{maximize}} & -b^T\nu - h^T\lambda - \lambda^T H\kappa \\ \text{subject to} & A^T\nu + G^T\lambda = -p, \quad \lambda \geq 0 \\ & 0 \leq \kappa \leq \psi, \quad q^T\kappa \geq d \end{array} \tag{8}$$

where ν and λ respectively are the dual variables corresponding to the equality and first inequality constraints in (7). Note that to find the optimal u one would use the optimal κ found from (8) to solve the inner minimization problem in (7), i.e., minimize $p^T u$ subject to the first two constraints in (7).

The objective function in (8) is bilinear in the variables λ and κ and is therefore nonconcave. In general it is intractable to find the global maximum of a nonconcave function. In the next section we propose an effective numerical method to solve (8), which is the main algorithmic contribution of this work. For small examples, where an exhaustive search is feasible, we demonstrate that the solution found by our algorithm is the same as the globally optimal solution.

IV. Guaranteed Sparsity and Binary Property of Failures

This section contains our main theoretical results. We demonstrate that even without an explicit promotion of sparsity in (7)–(8), the solution to the optimal interdiction problem is both sparse and binary. The solution is sparse in the sense that the attacker's best use of resources is to find the set of most consequential cells in the network, and it is binary in the sense that the attacker's best choice is to fail these cells fully (as opposed to partially).

Proposition 1: When the budget is not enough to fail all cells, there is an optimal solution of (8) that satisfies $q^T\kappa = d$, i.e., there is an optimal solution of

$$\begin{array}{ll} \underset{\kappa,\lambda,\nu}{\text{maximize}} & -b^T\nu - h^T\lambda - \lambda^T H\kappa \\ \text{subject to} & A^T\nu + G^T\lambda = -p, \quad \lambda \geq 0 \\ & 0 \leq \kappa \leq \psi, \quad q^T\kappa = d \end{array} \tag{9}$$

that solves (8). Furthermore, this solution has the property that all but (at most) one of the κ_i belong to $\{0, \psi_i\}$.

Proof: Problem (8) is equivalent to

$$\begin{array}{lll} \underset{\lambda,\nu}{\text{maximize}} & \underset{\kappa}{\text{maximize}} & -b^T\nu - h^T\lambda - \lambda^T H\kappa \\ \text{subject to} & & A^T\nu + G^T\lambda = -p, \quad \lambda \geq 0 \\ & & 0 \leq \kappa \leq \psi, \quad c^T(\mathbb{1} - \kappa/\psi) \leq e \end{array}$$

in which the inner maximization problem is

$$\begin{array}{ll} \underset{\kappa}{\text{maximize}} & -\lambda^T H\kappa \\ \text{subject to} & 0 \leq \kappa \leq \psi, \quad c^T(\mathbb{1} - \kappa/\psi) \leq e. \end{array} \tag{10}$$

Setting $\omega^T = \lambda^T H$, and denoting the ith element of a vector a by a_i, the last problem becomes

$$\begin{array}{ll} \underset{\kappa}{\text{maximize}} & -\sum_i \omega_i \kappa_i \\ \text{subject to} & 0 \leq \kappa_i \leq \psi_i, \ i = 1, 2, \ldots \\ & \sum_i c_i(1 - \kappa_i/\psi_i) \leq e \end{array}$$

which is further equivalent to

$$\begin{array}{ll} \underset{\kappa}{\text{maximize}} & \sum_i \omega_i(\psi_i - \kappa_i) \\ \text{subject to} & 0 \leq \psi_i - \kappa_i \leq \psi_i, \ i = 1, 2, \ldots \\ & \sum_i (c_i/\psi_i)(\psi_i - \kappa_i) \leq e. \end{array}$$

Setting $\theta_i = c_i/\psi_i$ and $\mu_i = \psi_i - \kappa_i$, the above problem can be rewritten as

$$\begin{array}{ll} \underset{\mu}{\text{maximize}} & \sum_i \omega_i \mu_i \\ \text{subject to} & 0 \leq \mu_i \leq \psi_i, \ i = 1, 2, \ldots \\ & \sum_i \theta_i \mu_i \leq e. \end{array}$$

Since both c_i and ψ_i are positive for every i then θ_i is positive for $i = 1, 2, \ldots$, and the last problem is equivalent to

$$\begin{array}{ll} \underset{\mu}{\text{maximize}} & \sum_i (\omega_i/\theta_i)\theta_i \mu_i \\ \text{subject to} & 0 \leq \theta_i \mu_i \leq \theta_i \psi_i, \ i = 1, 2, \ldots \\ & \sum_i \theta_i \mu_i \leq e. \end{array}$$

Setting $\tau_i = \theta_i \mu_i$ and recalling that $\theta_i \psi_i = c_i$, the above equation can be rewritten as

$$\begin{aligned}\underset{\tau}{\text{maximize}} \quad & \textstyle\sum_i(\omega_i/\theta_i)\tau_i \\ \text{subject to} \quad & 0 \le \tau_i \le c_i,\ i=1,2,\ldots \\ & \textstyle\sum_i\tau_i \le e.\end{aligned} \tag{11}$$

It can be shown that all the elements of H in (7) are non-negative. Since the elements of λ also are non-negative and $\omega^T = \lambda^T H$, it follows that the elements of ω are non-negative and therefore ω_i/θ_i is non-negative for $i = 1,2,\ldots$. This implies that the objective function in (11) is monotonically non-decreasing in every τ_i.

When the attack budget is not enough to fail all cells in the network, $e < \sum_i c_i$, from the monotonically non-decreasing property we conclude that there is a solution of (11) that satisfies $\sum_i \tau_i = e$. Clearly this implies that there is a solution of (10) which satisfies $c^T(\mathbb{1} - \kappa/\psi) = e$, or equivalently $q^T\kappa = d$. This proves that there is an optimal solution of (9) that solves (8).

Moreover, when $e < \sum_i c_i$, a solution of

$$\begin{aligned}\underset{\tau}{\text{maximize}} \quad & \textstyle\sum_i(\omega_i/\theta_i)\tau_i \\ \text{subject to} \quad & 0 \le \tau_i \le c_i,\ i=1,2,\ldots \\ & \textstyle\sum_i\tau_i = e\end{aligned}$$

is given by finding the index i_1 for which ω_i/θ_i is largest among all i and setting $\tau_{i_1} = c_{i_1}$ if $c_{i_1} \le e$ and $\tau_{i_1} = e$ if $c_{i_1} > e$. If $c_{i_1} \le e$ we proceed by finding the index i_2 for which ω_i/θ_i is second-largest among all i and setting $\tau_{i_2} = c_{i_2}$ if $c_{i_1} + c_{i_2} \le e$ and $\tau_{i_2} = e - c_{i_1}$ if $c_{i_1} + c_{i_2} > e$. This procedure is repeated until $\sum_i \tau_i = e$. Thus all but (at most) one of the τ_i belong to $\{0, c_i\}$, with those τ_i corresponding to the largest values of ω_i/θ_i equal to c_i and those τ_i corresponding to the smallest values of ω_i/θ_i equal to 0. This implies that for every $\lambda \ge 0$ there is a solution of

$$\begin{aligned}\underset{\kappa}{\text{maximize}} \quad & -\lambda^T H\kappa \\ \text{subject to} \quad & 0 \le \kappa \le \psi, \quad c^T(\mathbb{1} - \kappa/\psi) = e.\end{aligned}$$

with the property that all but (at most) one of the κ_i belong to $\{0, \psi_i\}$, which in turn proves the same property for problem (9). The proof of the proposition is now complete. ■

Proposition 1 demonstrates the sparsity of optimal failures. The attacker orders the nodes in terms of their importance, as determined by the coefficients in the objective function of (11), and fails them fully in descending order of importance until he has exhausted his budget. Since it is only meaningful that the attacker has limited resources/budget, this results in a sparse set of failed nodes.

V. Proposed Algorithm for Solving Problem (9)

This section contains our main algorithmic results. We solve the bilinear problem by iteratively updating one set of variables through a gradient-based step and then finding the globally optimal solution in the other set of variables. Numerical experiments demonstrate that our approach performs better in comparison with methods reported in earlier work [10].

Although the objective function in (9) is bilinear in the optimization variables, the constraints on κ and $\{\lambda, \nu\}$ are independent. If we fix one set of variables and optimize in the other, we can decompose problem (9) into two linear programs

$$\begin{aligned}\underset{\kappa}{\text{maximize}} \quad & -\lambda^T H\kappa \\ \text{subject to} \quad & 0 \le \kappa \le \psi, \quad q^T\kappa = d\end{aligned} \tag{12}$$

and

$$\begin{aligned}\underset{\lambda,\nu}{\text{maximize}} \quad & -b^T\nu - h^T\lambda - \lambda^T H\kappa \\ \text{subject to} \quad & A^T\nu + G^T\lambda = -p, \quad \lambda \ge 0.\end{aligned} \tag{13}$$

In our experiments we find that if we update the variables by iteratively solving linear programs (12) and (13) we rapidly converge to a sub-optimal solution of problem (9), which inhibits the search for the global optimum.

Motivated by block coordinate gradient descent (BCGD) and block coordinate descent (BCD) [11], we aim to solve problem (9) by iterating between updating κ using a gradient-based step and finding a globally optimal solution of $\{\lambda, \nu\}$. Generally, projected gradient descent is used as a one-step update for constrained problems; it first employs gradient descent to update the variables and then obtains their Euclidean projection onto the constraint set. In this paper we use the adaptive moment estimation (Adam) algorithm [13] instead of gradient descent in the update of κ. We will refer to this procedure as the projected Adam algorithm.

The Adam algorithm adds bias-correction terms based on the root mean square prop (RMSProp) [14] and the adaptive gradient (Adagrad) [15] algorithms, and is known to be robust and well-suited for a wide range of convex and non-convex problems [13]. In the case of solving (9), the momentum term in Adam helps avoid early convergence to a sub-optimal point. In our experiments we observe that using the projected Adam algorithm results in solutions that are far superior to those found by projected gradient descent.

To begin, we use the fact that maximizing the objective in (12) is equivalent to minimizing $\lambda^T H\kappa$. In our projected Adam algorithm we first use Adam to update κ, for which we compute

$$g^{(k+1)} = \nabla_\kappa\, \lambda^T H\kappa = H^T\lambda,$$

$$\rho^{(k+1)} = \beta_1\rho^{(k)} + (1-\beta_1)g^{(k+1)},$$

$$\sigma^{(k+1)} = \beta_2\sigma^{(k)} + (1-\beta_2)g^{(k+1)} \circ g^{(k+1)},$$

$$\hat{\rho}^{(k+1)} = \frac{\rho^{(k+1)}}{1-\beta_1^{k+1}}, \quad \hat{\sigma}^{(k+1)} = \frac{\sigma^{(k+1)}}{1-\beta_2^{k+1}},$$

with the update of κ given by

$$\kappa^{(k+1)} = \kappa^{(k)} - \frac{\eta \hat{\rho}^{(k+1)}}{\sqrt{\hat{\sigma}^{(k+1)}} + \epsilon \mathbb{1}}. \tag{14}$$

Here, $\circ$ denotes element-wise vector multiplication and division by vectors is performed element-wise; ϵ is a parameter with small positive values (to prevent division by zero in (14) and generally chosen to be 10^{-8}), the initial values of $\rho^{(0)}$ and $\sigma^{(0)}$ are zero, and the parameters β_1 and β_2 respectively are chosen to be 0.9 and 0.999; $\mathbb{1}$ is the column vector of all ones and η is the step size of the Adam algorithm.

Next, we find the Euclidean projection of $\kappa^{(k+1)}$ onto the constraint set by solving the quadratic program

$$\begin{aligned} &\underset{\kappa}{\text{minimize}} && \|\kappa - \kappa^{(k+1)}\|_2^2 \\ &\text{subject to} && 0 \le \kappa \le \psi, \quad q^T \kappa = d. \end{aligned} \tag{15}$$

In every iteration, after solving (15) we use $\kappa^{(k+1)}$ to denote the solution of (15) rather than the result of (14). We then set $\kappa = \kappa^{(k+1)}$ in problem (13) and solve the linear program to obtain $\{\lambda^{(k+1)}, \nu^{(k+1)}\}$. This concludes one iteration of our algorithm.

Algorithm 1 Proposed algorithm for solving (9)

```
1: initialize $\lambda^{(0)}$
2: for $k = 0, 1, \dots, k_{max}$ do
3:     Set $\lambda = \lambda^{(k)}$, find $\kappa^{(k+1)}$ using projected Adam algorithm (14)–(15).
4:     Set $\kappa = \kappa^{(k+1)}$, find $\{\lambda^{(k+1)}, \nu^{(k+1)}\}$ by solving linear program (13).
5: end for
```

Algorithm 1 summarizes our proposed iterative method for solving problem (9). In our experiments we observe that the number of needed iterations is reduced if we initialize $\lambda^{(0)}$ using the sub-optimal solution found by solving linear programs (12) and (13) iteratively, instead of a using a random point in the constraint set. The details of the initialization of $\lambda^{(0)}$ is discussed in Algorithm 2.

Algorithm 2 Initialization of $\lambda^{(0)}$

```
1: given $\kappa^{(0)} = \psi$
2: for $i = 0, 1, \dots, i_{max}$ do
3:     Solve problem (13) to find $\lambda^{(i)}$ and $\nu^{(i)}$, set $J_1^{(i)}$ as value of objective function.
4:     Solve problem (12) to find $\kappa^{(i+1)}$, set $J_2^{(i)}$ as value of objective function.
5:     if $J_1^{(i)} = J_2^{(i-1)}$ and $J_2^{(i)} = J_1^{(i)}$ then
6:         Break for loop.
7:     end if
8: end for
```

VI. Examples

In this section we illustrate the utility of Algorithms 1–2 for solving problem (9). We apply our approach to two different networks and compare the results we obtain with those reported in [10]. Our numerical experiments validate the theoretical results in Section IV. For all linear programs we use `CVXPY` [16], [17], a tool for convex programming in Python. We implement the Adam algorithm (14) in `Tensorflow` [18] and employ a step size of $\eta = 10$.

A. Example 1

We consider the network shown in Figure 1, which is taken from [10]. We prescribe that at each time step 2 units of mass enter nodes 1, 9 through their respective on-ramps. Node 11 is the sink cell. We take $\phi = 1.2\,\psi$ and $\bar{t} = 12$. The cost, flow-capacity, and initial mass vectors respectively are given by

$$c = \begin{bmatrix}3, 2, 1, 2, 2, 1, 2, 1, 3, 2\end{bmatrix}^T$$
$$\psi = \begin{bmatrix}4, 3, \tfrac{3}{2}, 3, 3, \tfrac{3}{2}, 3, \tfrac{3}{2}, 4, 3\end{bmatrix}^T$$
$$x(0) = \begin{bmatrix}2, 1, 1, 1, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, 2, 2\end{bmatrix}^T.$$

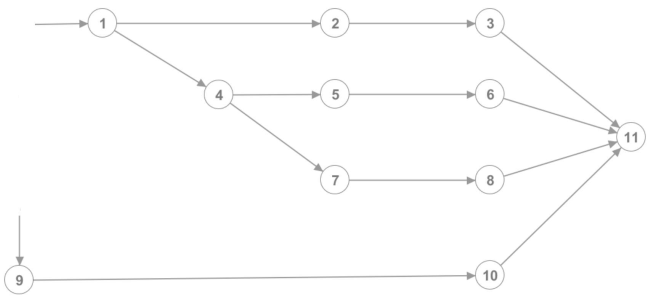

Fig. 1: Network from [10]

	Our method	URI [10]	BRI [10]	Exhaustive search
e	Failures	Failures	Failures	Failures
1	3	3	3	3
3	3,6,8	3,6,8	3,6,8	3,6,8
5	3,6,8,10	3,6,8,10	3,6,8,10	3,6,8,10

TABLE I: Comparison of different numerical algorithms for network in Figure 1.

Table I demonstrates that for budgets $e = 1, 3, 5$ both our numerical method and that proposed in [10] successfully find the globally optimal attack obtained through exhaustive search.

B. Example 2

We consider the network shown in Figure 2, which is a traffic network adapted by [10] from [19]. We prescribe that at each time step 2 units of mass enter nodes 1, 4 and 1 unit of mass enter node 3, all through their respective on-ramps. Node 18 is the sink cell. We take $\phi = 1.2\,\psi$ and $\bar{t} = 12$. The cost, flow-capacity, and initial mass vectors respectively are given by

$$c = \left[3,2,2,3,2,2,3,3,2,2,2,2,3,3,1,3,2\right]^T$$
$$\psi = \left[6,3,3,6,3,3,5,5,3,3,3,3,5,5,2,5,3\right]^T$$
$$x(0) = \left[2,3,\tfrac{3}{2},3,3,3,3,3,3,3,2,2,\tfrac{3}{2},2,2,2,2\right]^T.$$

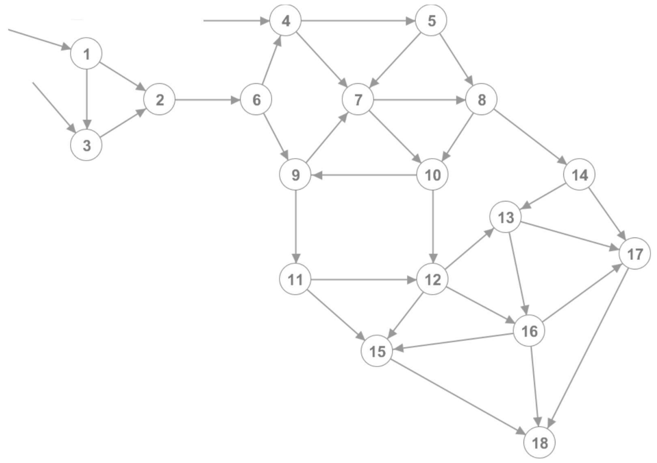

Fig. 2: Network from [10] based on [19, Chap. 19].

	Our method	URI [10]	BRI [10]	Exhaustive search
e	Failures	Failures	Failures	Failures
3	15,17	15, (16)	15, (16)	15,17
4	15,17	15, (16)	15, (16)	15,17
6	15,16,17	15,16,17	15,16,17	15,16,17

TABLE II: Comparison of different numerical algorithms for network in Figure 2.

Table II demonstrates that for budget $e = 6$ both our numerical method and that in [10] find the globally optimal attack obtained through exhaustive search. In this case, the budget is enough to fully block the network; it is clear that failing nodes 15, 16, 17 is optimal since these failures prevent any mass from leaving the network. For $e = 3, 4$ the problem is more challenging, as the budget is no longer enough to fully block the network. Still, our proposed approach successfully finds the globally optimal attack whereas the method in [10] does not.

VII. Conclusions & Future Work

In this paper, we study the optimal interdiction problem for transportation networks. We prove that the optimal solution is both sparse and binary, despite the absence of any sparsity/binary regularizers or constraints in the formulation. We also propose a numerical method to solve the bilinear program that is equivalent to the optimal interdiction problem, and demonstrate that it finds globally optimal solutions in the small networks we tested.

Since our numerical algorithm employs a combination of linear programming and a first-order gradient-based method, we expect that it will scale gracefully for large networks. Applying our approach to large networks is part of our current research efforts.

References

[1] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[2] C. F. Daganzo, "The cell transmission model, Part II: network traffic," *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 79–93, 1995.

[3] A. K. Ziliaskopoulos, "A linear programming model for the single destination system optimum dynamic traffic assignment problem," *Transportation science*, vol. 34, no. 1, pp. 37–49, 2000.

[4] G. Como, K. Savla, D. Acemoglu, M. A. Dahleh, and E. Frazzoli, "Robust distributed routing in dynamical networks – Part I: Locally responsive policies and weak resilience," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 317–332, 2013.

[5] G. Como, K. Savla, D. Acemoglu, M. A. Dahleh, and E. Frazzoli, "Robust distributed routing in dynamical networks – Part II: Strong resilience, equilibrium selection and cascaded failures," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 333–348, 2013.

[6] K. Savla, G. Como, and M. A. Dahleh, "Robust network routing under cascading failures," *IEEE Transactions on Network Science and Engineering*, vol. 1, no. 1, pp. 53–66, 2014.

[7] G. Como, E. Lovisari, and K. Savla, "Throughput optimality and overload behavior of dynamical flow networks under monotone distributed routing," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 57–67, 2015.

[8] G. Bianchin, F. Pasqualetti, and S. Kundu, "Resilience of traffic networks with partially controlled routing," in *Proceedings of the 2019 American Control Conference*, 2019, pp. 2670–2675.

[9] Y.-C. Liu, G. Bianchin, and F. Pasqualetti, "Secure trajectory planning against undetectable spoofing attacks," *Automatica*, vol. 112, pp. 1–10, 2020.

[10] G. Kearney and M. Fardad, "On the induction of cascading failures in transportation networks," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1821–1826.

[11] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[12] G. Como, E. Lovisari, and K. Savla, "Convex formulations of dynamic network traffic assignment for control of freeway networks," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 755–762.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conferenceon Learning Representations*, 2015.

[14] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, p. 8, 2012.

[15] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[16] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[17] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[18] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[19] D. Easley and J. Kleinberg, "Networks, crowds, and markets: reasoning about a highly connected world," 2010.