2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

On the Induction of Cascading Failures in Transportation Networks

Griffin Kearney and Makan Fardad

Abstract— **We examine the effect of malicious attacks in disrupting optimal routing algorithms for transportation networks. We model traffic networks using the cell transmission model, which is a spatiotemporal discretization of kinematic wave equations. Here, vehicles are modeled as masses and roads as cells, and traffic flow is subject to conservation of mass and capacity constraints. At time zero a resource-constrained malicious agent reduces the capacities of cells so as to maximize the amount of time mass spends in the network. For the resulting set of capacities the network router then solves a linear program to determine the flow configuration that minimizes the amount of time mass spends in the network. Our model allows for the outright or partial failure of road cells at time zero, the effects of which can cause cascading failure in the network due to irreversible blockages resulting from congestion. This two-player problem is written as a max-min optimization and is reformulated to an equivalent nonconvex optimization problem with a bilinear objective and linear constraints. Linearization techniques are applied to the optimization problem to find local solutions. Analyzing illustrative examples shows that attackers with relatively small resource budgets can cause widespread failure in a traffic network.**

Index Terms— **Traffic network, cascading failures, flow networks, network interdiction**

I. Introduction

Flow networks are ones in which materials are introduced at source cells and, after being routed through the network, are removed at sink cells. Flow networks are described by directed graphs subject to (i) conservation of mass constraints, and (ii) capacity constraints on the amount of flow that can travel through each link. Highway networks, disaster evacuation plans, water supply networks, and (routing of data packets in) computer networks are all examples of flow networks. While our treatment of flow networks in this paper is general, we will focus on highway transportation networks to motivate the development.

Intriguingly, in transportation networks conservation of mass can lead to failures that are distant [1], [2], whereas failures spread from neighbor to neighbor in many models of failure propagation (e.g., networks governed by threshold dynamics in which a component fails if at least a certain fraction of its neighbors have failed [3]–[8]). By way of example, consider a transportation network in which a link experiences failure due to malicious activity or an accident; this increases the flow on alternate routes, which may result in the failure of a link (possibly far removed from the first) whose capacity does not allow for the increased flow of traffic, and so on. In this context, herein we adopt the point of view of a malicious adversary seeking to identify those few links whose failure maximally disrupts traffic flow.

The cell transmission model was developed by Daganzo in two seminal papers [9], [10] via a spatiotemporal discretization of the hydrodynamic model of traffic flow. The cell transmission model captures complex traffic behavior and congestion effects, including transient phenomena and the propagation of shocks. Reference [11] exploited the piecewise-linear relationships between vehicle flow and density, inherent in the cell transmission model, to formulate the optimal traffic assignment problem as a linear program. In a highly influential body of work [1], [2], [12], [13], Como *et al.* recently investigated questions of resilience, throughput maximization, and decentralized routing in transportation networks. An important result in [2], [14] was the uncovering of distributed routing policies that depend on local information, maximize throughput, and maximally delay congestion effects under adversarial perturbations to the links' capacities.

In this paper we study transportation networks that model the flow of vehicles on a network of highways. Our goal is to find a small set of roadways to attack at time zero such that the resulting vehicle congestion will be amplified and propagated by the network's natural dynamics in order to maximally disrupt the flow of traffic. This problem is combinatorial in nature and intractable in general.

In Section II we summarize a model, which is an adaptation of Ziliaskopoulos [11], as a starting point for our analysis. Our formulation is based on this established model and that of Como *et al.*, however we increase the severity of congestion effects by allowing blockages from traffic accumulation to persist indefinitely. We further extend the framework discussed in Section II in order to model the effects of an attack by a resource-constrained malicious agent. The resource constraints of the malicious agent limit the scale and severity of permissible attacks which are used to disrupt the network. In our framework, the malicious agent and traffic router are treated as players engaged in a two-player game and the resulting augmented model is formulated as a max-min program.

In Section III we utilize the dual reformulation of the problem to represent the max-min program as a nonconvex quadratic program. In Section IV we apply our work to illustrative examples which permit an exhaustive search for globally optimal solutions in order to validate our results.

Financial support from the National Science Foundation under award ECCS-1609916 is gratefully acknowledged.

G. Kearney and M. Fardad are with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244. E-mail: (G. Kearney) gmkearne@syr.edu; (M. Fardad) makan@syr.edu

Finally, we include directions and motivation for future work on transportation networks in Section V.

II. Problem Formulation

In order to provide physical motivation to support the problem formulation we restrict our focus to traffic networks, but our models can be generalized to larger classes of transportation and flow networks with appropriate constraint modifications.

A. Preliminaries

We assume that the traffic network under consideration contains N cells and that a single stretch of roadway can be modeled as a series of consecutive cells [11]. In this model we represent cells and junctions as the nodes and edges of a graph, respectively. We also assume that the on-ramp topology is represented by the nonzero entries of the vector $v(t)$ such that $v_i(t)$ represents the incoming mass to node i at time t from outside the network. Similarly, the off-ramp topology is also specified and represented by the non-zero entries of a vector $w(t)$ such that $w_i(t)$ represents the maximum allowable traffic which can exit the network from node i at time t.

We take $x_i(t)$ to be the mass occupying node i, $y_i(t)$ to be the incoming mass to node i, and $z_i(t)$ to represent the outgoing mass from node i, all at time t. We denote the mass flow from node i to node j at time t as $f_{ij}(t)$, and we associate each node of the network with two parameters: ϕ_i and κ_i, which represent the maximum mass-capacity and maximum flow-capacity of node i, respectively. Finally, in order to restrict the scale of attacks on the network we introduce two additional parameters: a vector c with c_i quantifying the cost to fully fail node i, and a scalar parameter b representing the resource budget available to the attacker.

It is convenient to introduce two additional nodes besides those contained in the network: a source and a sink node. The sum of the incoming on-ramp mass is first generated at the source node at each time step. The mass is then routed into the network from the source node based on the on-ramp topology. Similarly, all traffic mass leaving the network based on the off-ramps is collected at the sink node. These nodes will be indexed with 0 and $N+1$, respectively.

We proceed by discussing the routing scheme which governs the passsage of traffic through the network. We assume that the routing protocol prescribes traffic patterns by minimizing an objective function of the form:

$$\sum_{t=1}^{\hat{t}} p^T x(t)$$

where we choose a penalty vector p such that the minimization of the objective results in the optimal routing of mass from the source node to the sink node; optimal routing of mass is problem specific and thus selection of p should be done on a case-by-case basis. Furthermore, mass transfer between nodes of the network is governed by [15]:

$$y_i(t) = v_i(t) + \sum_j f_{ji}(t),$$

$$z_i(t) = w_i(t) + \sum_j f_{ij}(t),$$

and

$$x_i(t) = x_i(t-1) + y_i(t-1) - z_i(t-1).$$

In order to respect the network topology, we require that $f_{ij}(t)$ be zero if there is no link from node i to node j.

We next describe the constraints that capture mass conservation, network limitations, and the effect of traffic congestion: we restrict that all flows be non-negative, $f_{ij}(t) \geq 0$, and that the amount of mass present at a node does not exceed the capacity of that node at any time

$$x(t) \leq \phi.$$

We further maintain that no more mass leaves a node than what is present at any time

$$z(t) \leq x(t),$$

and we include additional constraints to model congestion at a node as it approaches its maximum capacity

$$y(t) \leq \alpha_y(\phi - x(t)),$$

with α_y a scalar used to represent how acute the effects of congestion are. Unlike the work of [2], we include a similar congestion constraint on the outflow of nodes

$$z(t) \leq \alpha_z(\phi - x(t)).$$

This constraint causes congestion-based road failures in our model to be irreversible, by locking mass in place once a node has become fully congested. Finally, we require that no node allows inflows or outflows which exceed the associated maximum flow-capacity

$$y(t) \leq \kappa, \;\; z(t) \leq \kappa.$$

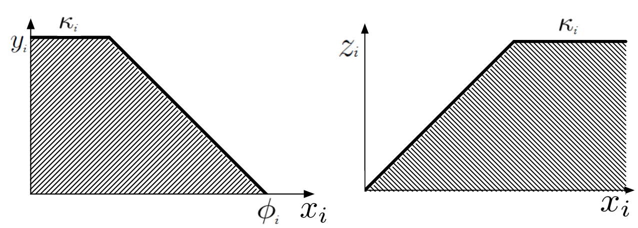

Fig. 1: Plots depicting feasible mass flow supply y and mass flow demand z as functions of cell mass x, respectively

B. Optimal Routing

For convenience we encapsulate all network optimization variables in the vector u such that

$$u = \begin{bmatrix} x \\ y \\ z \\ f \end{bmatrix} \text{ and } x = \begin{bmatrix} x(1) \\ x(2) \\ . \\ . \\ . \\ x(\hat{t}) \end{bmatrix}$$

where the entries of y, z, and f are defined in a similar manner to x. It is standard to construct a vector g such that the objective above is written as $g^T u$ in order to absorb the summation over time; thus we consider the routing program to be of the form

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & g^T u \\ \text{subject to} \quad & Au \leq H\kappa + b \\ & Gu = h \\ & u \geq 0 \end{aligned} \tag{1}$$

where the constraint matrices and vectors are constructed using the physical flow limitations discussed in the above preliminaries.

C. Optimal Attacking

We extend the routing model (1) to allow for attacks on the road network. We do this by allowing the malicious agent to reduce the flow-capacity κ of roads within the network at time $t = 0$, and we restrict the magnitude of the network attacks by modeling resource constraints. For instance, if an attacker seeks to block the roads of a city, then they would be limited by the number of barricades available to them. In doing this we suppose that $\hat{\kappa}$ is the flow-capacity vector of the network without disruption and κ will thus be treated as a vector of continuous variables which satisfy

$$0 \leq \kappa \leq \hat{\kappa}.$$

We define a cost vector c with c_i being defined as the resource cost to reduce κ_i to zero. γ is defined as the maximum budget available to the attacker. Therefore, we write the resulting resource constraint as

$$c^T(\mathbb{1} - \frac{\kappa}{\hat{\kappa}}) \leq \gamma \implies d^T\kappa \leq \beta,$$

where $d = -\frac{c}{\hat{\kappa}}^T$, $\beta = \gamma - c^T\mathbb{1}$, and division by vectors is performed entry-wise. Thus, the problem formulation with the inclusion of optimal attacks takes the form

$$\begin{aligned} \underset{\kappa}{\text{maximize}} \ \underset{u}{\text{minimize}} \quad & g^T u \\ \text{subject to} \quad & Au \leq H\kappa + b \\ & Gu = h \\ & u \geq 0 \\ & d^T\kappa \leq \beta \\ & 0 \leq \kappa \leq \hat{\kappa} \end{aligned} \tag{2}$$

In the next section we provide a strategy for finding locally optimal solutions to (2).

A solution of (2) can be considered as rendering a lower-bound on what a malicious attacker can achieve, in the sense that the attacker can impede traffic and cause cascading failures more severely than what (2) uncovers. This is because in reality there is no optimal centralized router with full knowledge of the mass and failures in the network and with full actuation capacity of the mass.

III. Dual Reformulation and Numerical Algorithm

We analyze (2) using the dual reformulation which is briefly reviewed. After performing such a transformation we arrive at a bilinear objective for which we find locally optimal points using linearization methods.

A. Dual Reformulation

Theorem 3.1: If the feasible set specified by the constraints of a linear program is non-empty and bounded, then

$$\begin{aligned} \underset{\mathrm{x}}{\text{minimize}} \quad & \mathrm{f}^T\mathrm{x} \\ \text{subject to} \quad & \mathrm{Ax} \geq \mathrm{b} \\ & \mathrm{Gx} = \mathrm{c} \\ & \mathrm{x} \geq 0 \end{aligned} \tag{P}$$

is equivalent to

$$\begin{aligned} \underset{\mu,\upsilon}{\text{maximize}} \quad & \mathrm{b}^T\mu + \mathrm{c}^T\upsilon \\ \text{subject to} \quad & \mathrm{A}^T\mu + \mathrm{G}^T\upsilon \leq \mathrm{f} \\ & \mu \geq 0 \end{aligned} \tag{D}$$

We omit a proof of Theorem 3.1 for brevity, and the details can be found in [16]. Theorem 3.1 requires that the feasible set be bounded which we next prove for (2).

Proposition 3.2: The feasible set of (2) is bounded.

Proof: It is sufficient to show that the optimization variables contained in u are bounded from above and below. By construction we have that $u \geq 0$ thus we need only show that u is bounded above.

We have $x(t) \leq \phi$, by construction, and since $z(t) \leq x(t)$ then we immediately conclude that $z(t) \leq \phi$. In addition to this, $y(t) \leq \alpha_y(\phi - x(t)) = \alpha_y\phi - \alpha_y x(t)$ and since $x(t) \geq 0$ then $-\alpha_y x(t) \leq 0$ so that we conclude $y(t) \leq \alpha_y\phi$. Finally, observe that $f_{ij}(t) \leq \sum_j f_{ij}(t) + w_i(t) = z_i(t) \leq \phi_i$. Thus $f_{ij}(t)$ is bounded above for all i, j implying that $f(t)$ is bounded above for all t. Since $x(t)$, $y(t)$, $z(t)$, and $f(t)$ are bounded above, then u is as well, which completes the proof. ■

We utilize Theorem 3.1 to rewrite (2), as in the following Proposition.

Proposition 3.3: Problem (2) is equivalent to

$$\begin{aligned}\underset{\kappa,\lambda,\nu}{\text{maximize}} \quad & -\kappa^T H^T\lambda - b^T\lambda + h^T\nu \\ \text{subject to} \quad & -A^T\lambda + G^T\nu \le g \\ & \lambda \ge 0 \\ & d^T\kappa \le \beta \\ & 0 \le \kappa \le \hat{\kappa}\end{aligned} \tag{3}$$

Proof: For any value of κ we may reformulate the inner minimization of (2) using Theorem 3.1, since the feasible set is bounded by Proposition 3.2, to write

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

This maximization problem can then be absorbed into the exterior maximization with respect to κ [16]. ■

In the formulation (3) the variables λ, and ν are the dual variables induced by the constraints on u in (2). Computing the global maximum of a constrained nonconcave bilinear objective is computationally intractable in general. In what follows, we utilize numerical techniques to compute locally optimal solutions of (3).

B. Linearization Method for Solving (3)

We describe the method used for iteratively linearizing the objective of (3) in order to compute solutions to the quadratic program. The objective is linearized about an initial set of feasible variables, and the resulting linear program is solved for an optimal variable update. The linearization point is then updated using the solution of the linear program and this procedure is repeated until the algorithm converges. Numerical investigations such as the examples contained in the following section provide promising results, but the solution of the proposed scheme is not guaranteed to be globally optimal in general.

Let

$$-\kappa^T H^T\lambda = \frac{1}{2}v^T Q v$$

with

$$Q = \begin{bmatrix} 0 & 0 & -H \\ 0 & 0 & 0 \\ -H^T & 0 & 0\end{bmatrix}, v = \begin{bmatrix}\lambda \\ \nu \\ \kappa\end{bmatrix}$$

and the 0's of Q representing zero block matrices of the appropriate dimensions. We denote the objective with $J(v)$ and write

$$J(v) = (\frac{1}{2}v^T Q v + q^T v) \text{ with } q = \begin{bmatrix} -b \\ h \\ 0\end{bmatrix}$$

and consider our problem as one in the variable v by recasting the constraints in a similar manner. Thus we treat the problem as one of the form

$$\begin{aligned}\text{maximize} \quad & J(v) \\ \text{subject to} \quad & Mv \le r\end{aligned} \tag{4}$$

where M and r are a matrix and vector derived from the linear inequality constraints of (3). The gradient of the objective is $\nabla J(v) = Qv + q$, and we linearize the problem by taking $v = \overline{v} + \widetilde{v}$. Treating v in such a way allows us to approximate J by Taylor series about the point $\overline{v}$ using the first order term and we take $\widetilde{v}$ to be the running variable of the optimization. Doing this results in

$$\begin{aligned}\text{maximize} \quad & [\nabla J(\overline{v})]^T\widetilde{v} \\ \text{subject to} \quad & M\widetilde{v} \le r - M\overline{v}\end{aligned} \tag{5}$$

Problem (5) is a linear program and is solved for $\widetilde{v}$. The vector $\overline{v}$ is then updated using this optimal solution, and the linear program is solved again using the updated $\overline{v}$. This procedure is iterated until $\widetilde{v}$ is sufficiently small. The process is summarized in Algorithm 1.

Algorithm 1 Solution of (4)

1: **given** Q, q, M, and r
2: **set** $0 < \mu$, $\overline{\delta} \ll 1$
3: **initialize** $\overline{v}$
4: **while** $\|\widetilde{v}\|_2 > \overline{\delta}$ **do**
5: Compute $\widetilde{v}$ by solving (5)
6: Set $\overline{v} := \overline{v} + \mu\widetilde{v}$
7: Update objective and constraints of (5) with new $\overline{v}$
8: **end while**
9: **output** $\overline{v}$

The algorithm converges if μ is set sufficiently small. The value of $\overline{v}$ to which the algorithm converges contains the dual variables and the modified flow-capacity vector. The primal variables can be recovered using the dual variables, or by solving the linear program (1) while setting κ to the modified values contained in the output $\overline{v}$. For the purposes of our traffic investigation, solving (1) is less susceptible to numerical error since recovering the primal variables from the dual solution directly involves the inversion of poorly conditioned matrices recovered from the complimentary slack condition.

The solution of the algorithm does depend on the choice of initialization point. Different strategies for initialization point selection are numerically investigated in the examples in Section IV, and we provide a brief description of the two techniques employed before proceeding. The techniques are utilized to initialize the vector κ, while the vector λ is initialized to the dual variables which result from the solution of the dual problem of (1) using $\kappa = \hat{\kappa}$. We denote the initialized value of κ for the iterative linearization as κ_0.

1) Uniform Reduced Initialization: As the name suggests, in this method we initialize κ by scaling the nominal flow capacities such that $\kappa_0 = \epsilon\hat{\kappa}$ where ϵ is a scalar chosen from the interval $[0, 1]$. For the purposes of the following examples we select ϵ such that the attacker budget constraint is satisfied, but (5) can be initialized using other values of ϵ.

2) Biased Reduced Initialization: The biased reduced initialization is a slight modification to the previous uniform reduction. In this method we select a subset of the network nodes denoted by K. We then proportionally reduce the flow capacities for nodes in K, and initialize κ to the nominal flow capacities for nodes not in K. If $i \in K$ $(\kappa_0)_i = \epsilon\hat{\kappa}_i$ and $(\kappa_0)_i = \hat{\kappa}_i$ if $i \notin K$ where ϵ is taken from the interval $[0, 1]$. For the purposes of the following examples K is selected to contain any nodes which satisfy the condition $c_i \leq \gamma$, and ϵ is selected to respect the attacker budget constraint. This condition is equivalent to reducing the initialized flow capacities of all nodes which could be fully failed proportionally to the prescribed attacker budget.

IV. Examples

A. Example 1

The first example is the 10 node traffic network shown in Figure 2 which is based on a network presented in [17]. This network was selected because it exhibits intuitive solutions and is sufficiently small to allow the use of exhaustive methods to show that the solutions computed using our scheme are globally optimal.

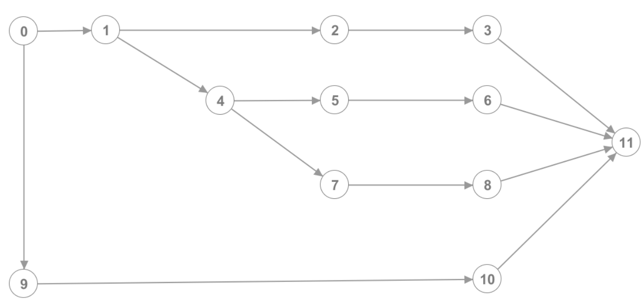

Fig. 2: A 10 node network based on [17].

As described in Section II, we introduce source and sink nodes represented by cells 0 and 11, respectively. Furthermore, we prescribe that 2 units of traffic mass enter nodes 1 and 9 at each time step. The cost and nominal flow-capacity vector are given by:

$$c = \left[3, 2, 1, 2, 2, 1, 2, 1, 3, 2\right]^T$$

$$\hat{\kappa} = \left[4, 3, 1.5, 3, 3, 1.5, 3, 1.5, 4, 3\right]^T$$

and we focus the examination on three scenarios corresponding to attacker budgets of $\gamma = 1$, 3, and 5. The resulting attacker strategy for each case is presented in Table I. The column headed with URI depicts the results using the uniform reduced initialization. The column headed with BRI depicts the results using the biased reduced initialization, and the column headed with GO depicts the global optimal found using an exhaustive search. The exhaustive search was carried out over all permissible full-node failures given the prescribed attacker budget; partial road failures were not scanned exhaustively. In the case where the solution of our exhaustive search is non-unique, we choose the one which most closely matches the result of our numerical algorithm.

For each of these scenarios we set $\hat{t} = 12$, and we fix the penalty vector such that

$$p = \left[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1\right]^T.$$

This choice of penalty vector will maximize the mass which successfully passes through the network to the sink node. We set the initial mass distribution to the final-time $\hat{t}$ mass distribution given by the solution of (1),

$$x(0) = \left[2, 1, 1, 1, 0.5, 0.5, 0.5, 0.5, 2, 2\right]^T.$$

When solving (1) to compute this mass distribution, we assume that the network starts with zero initial mass.

	URI		BRI		GO	
γ	Failures	J	Failures	J	Failures	J
1	3	-127	3	-127	3	-127
3	3,6,8	-55	3,6,8	-55	3,6,8	-55
5	3,6,8,10	0	3,6,8,10	0	3,6,8,10	0

TABLE I: Failure targets and resulting objective values for the network in Figure 2.

In each of the three cases our approach correctly identifies a globally optimal solution. The cell failures prescribed by our algorithm are non-fractional due in part the values of γ, but in the next example we demonstrate that the algorithm will prescribe fractional failures even if γ is selected to support full failures.

B. Example 2

In the second example we examine a traffic network adapted from [6] and depicted in Figure 3. We use node 0 to represent the source and node 18 to represent the sink. While it is impractical to perform a full exhaustive search, this network is small enough to permit a limited exhaustive search. In this case, we limit the exhaustive search to scan over all modified flow-capacity configurations with no more than 3 failures.

In this example we prescribe that at each time step 2 units of traffic mass enter nodes 1 and 4, and 1 unit of traffic mass enters node 3. The cost and nominal flow-capacity vector are given by:

$$c = \left[3, 2, 2, 3, 2, 2, 3, 3, 2, 2, 2, 2, 3, 3, 1, 3, 2\right]^T$$

$$\hat{\kappa} = \left[6, 3, 3, 6, 3, 3, 5, 5, 3, 3, 3, 3, 5, 5, 2, 5, 3\right]^T$$

and in the same style as the first example we examine three scenarios corresponding to attacker budgets of $\gamma = 3$, 4, and 6. The resulting attacker strategies are presented in the same way as the previous example in Table II. It should be noted that because of the larger size of this network a full exhaustive search is computationally prohibitive so the column headed ES includes the results of a search over all

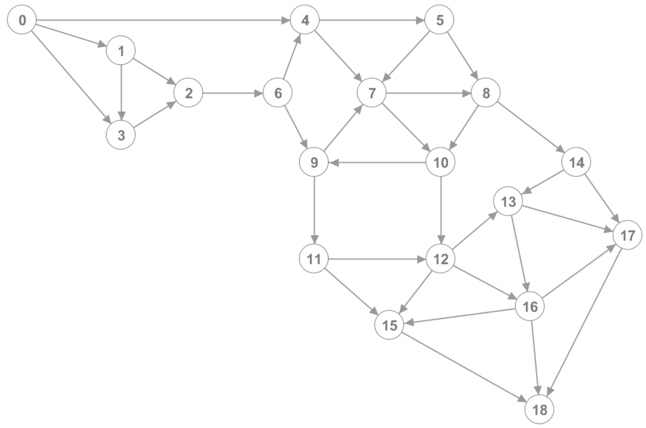

Fig. 3: A 17 node network based on [6, Chap. 19].

failure configurations of cardinality less than or equal to 3. We set $\hat{t} = 12$,

$$p = \left[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1\right]^T,$$

and $x(0)$ to the steady state mass distribution as described in the previous example.

	URI		BRI		GO	
γ	Failures	J	Failures	J	Failures	J
3	15,(16)	-278	15,(16)	-278	15,17	-156
4	15,16	-211	15,16	-211	15,17	-156
6	15,16,17	0	15,16,17	0	15,16,17	0

TABLE II: Failure targets and resulting objective values of the network shown in Figure 3. Failures listed within parentheses indicate a fractional failure.

In the case of the second example the algorithm computes sub-optimal solutions in the cases $\gamma = 3$ and $\gamma = 4$. However, when $\gamma = 6$ the globally optimal solution is computed.

V. Conclusions and Future Work

We develop a framework to search for optimal attacks that trigger cascading failures in traffic networks. We do this by modifying the well-known cell transmission model. We demonstrate that the resulting optimization problem naturally can be viewed as a two player max-min problem. We employ duality theory to equivalently reformulate this as a maximization problem with a bilinear objective and linear constraints. Globally optimal solutions of this problem are generally intractable to find, and therefore we utilize iterative linearization techniques to obtain locally optimal solutions. Each iteration of our algorithm involves solving a linear program, which scales gracefully with network size.

In the small scale examples discussed in Section IV we obtained globally optimal solutions in the case of Example 1 when the algorithm is initialized using the two methods discussed in Section III. On the other hand, in Example 2 we observed that the algorithm computed sub-optimal solutions using the same two initialization techniques. We expect that the application of this work to large, complex networks will require implementing more sophisticated initialization and linearization techniques.

References

[1] K. Savla, G. Como, and M. A. Dahleh, "Robust network routing under cascading failures," *IEEE Transactions on Network Science and Engineering*, vol. 1, pp. 53–66, 2014.

[2] 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. 2, pp. 57–67, 2015.

[3] M. Granovetter, "Threshold models of collective behavior," *American Journal of Sociology*, vol. 83, pp. 1420–1443, 1978.

[4] E. M. Rogers, *Diffusion of Innovations*. Free Press, 2003.

[5] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of KDD*, 2003.

[6] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[7] G. Kearney and M. Fardad, "On a framework for analysis and design of cascades on Boolean networks," in *Proceedings of the 55th IEEE Conference on Decision and Control*, Dec. 2016, pp. 997–1002.

[8] M. Fardad and G. Kearney, "On a linear programming approach to the optimal seeding of cascading failures," in *Proceedings of the 56th IEEE Conference on Decision and Control*, Dec. 2017, pp. 102–107.

[9] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research B*, vol. 28B, pp. 269–287, 1994.

[10] ——, "The cell transmission model, part II: Network traffic," *Transportation Research B*, vol. 29B, pp. 79–93, 1995.

[11] A. K. Ziliaskopoulos, "A linear programming model for the single destination system optimum dynamic traffic assignment problem," *Transportation Science*, vol. 34, pp. 37–49, 2000.

[12] 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.

[13] ——, "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.

[14] ——, "Robust distributed routing in dynamical networks with cascading failures," in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 7413–7418.

[15] G. Como, E. Lovisari, and K. Savla, "Convex formulations of dynamic network traffic assignment for control of freeway networks," in *53rd Annual Allerton Conference*, 2015, pp. 755–762.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[17] G. Como, "Resilient control of dynamical flow networks," *Presentation given at the University of Minnesota: Institute for Mathematics and its Applications (https://www.ima.umn.edu/2015-2016/W10.19-23.15/24190)*, 2015.