Trilevel Network Interdiction Game for the Minimum Cost Flow Problem with Input Dependence

Adam Rumpf[1] and Hemanshu Kaul[2]

[1]Department of Applied Mathematics, Florida Polytechnic University, [2]Department of Applied Mathematics, Illinois Institute of Technology

APPLIED MATHEMATICS
FLORIDA POLYTECHNIC UNIVERSITY

Introduction

As society continues its trend towards urbanization it also becomes more reliant on highly interdependent civil infrastructure networks, wherein the functionality of components in one system depend on delivery of resources from another (e.g. telecommunications and light rail networks that require delivery of electric power from the power grid). These systems are uniquely vulnerable to damage from natural disasters or from targeted attacks due to the phenomenon of *cascading failures* that can cause damage to extend far beyond the original source (see Figure 1).

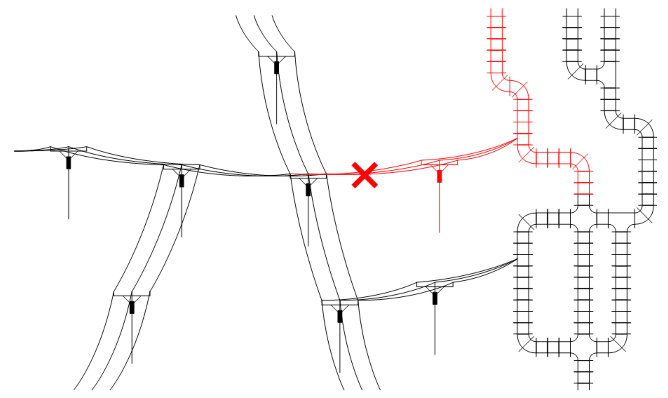

Figure 1: An illustration of cascading failures. Parts of a rail network may depend on delivery of electric power in order to function, meaning that damage to the electric power network may lead to loss of functionality in the rail network.

The goal of this study is to help to plan optimal fortifications to a collection of interdependent networks against a targeted attack or worst-case natural disaster. The model takes the form of a multilevel optimization problem which may be computationally intractable to solve exactly for large networks, and so approximation methods based on linear relaxation are developed and tested.

Network Interdiction Game

Our model for disaster preplanning and recovery consists of a trilevel *network interdiction game* on a flow network $G = (\mathcal{N}, \mathcal{A})$ representing a collection of interdependent civil infrastructure systems. It is a sequential game played by a defender and an attacker in three stages:

Stage 1. The defender chooses a limited set of arcs to defend.

Stage 2. The attacker chooses a limited set of undefended arcs to destroy.

Stage 3. The defender minimizes the flow cost on the resulting interdependent network.

The objective is the flow cost of the network at the end of Stage 3, which the defender wants to minimize (by protecting important arcs) and the attacker wants to maximize (by destroying important arcs).

Underlying Interdependent Network

The underlying network $G = (\mathcal{N}, \mathcal{A})$ is a generalized minimum cost flow network, equipped with net supply values b_i for each node $i \in \mathcal{N}$ and costs c_{ij} and capacities u_{ij} for each arc $ij \in \mathcal{A}$ as in the standard minimum cost flow problem. Interdependencies are represented using a **binary input dependence** model [3], which describes the phenomenon of one network component requiring delivery of resources from another network component in order to function.

We define a set $\mathcal{I} \subset \mathcal{A} \times \mathcal{A}$ of **parent** and **child** arc pairs (ij, kl). Constraints are added that cause a child $kl \in \mathcal{A}$ to become unusable if its parent $ij \in \mathcal{A}$ is not saturated.

Trilevel Optimization Model

The trilevel network interdiction game described above can be stated as the following multilevel *mixed integer-linear program (MILP)*.

$$\min_{\boldsymbol{\xi}\in\Xi} \max_{\boldsymbol{\psi}\in\Psi} \min_{\substack{\mathbf{x},\mathbf{s}\geq\mathbf{0} \\ \mathbf{y}\in\{0,1\}^p}} \sum_{ij\in\mathcal{A}} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{j:ij\in\mathcal{A}} x_{ij} - \sum_{j:ji\in\mathcal{A}} x_{ji} = b_i \quad \forall i \in \mathcal{N} \tag{2}$$

$$\psi_{ij} \leq 1 - \xi_{ij} \quad \forall ij \in \mathcal{A} \tag{3}$$

$$x_{ij} \leq u_{ij}(1 - \psi_{ij}) \quad \forall ij \in \mathcal{A} \tag{4}$$

$$x_{ij} + s_{ij} = u_{ij} \quad \forall ij : (ij, kl) \in \mathcal{I} \tag{5}$$

$$s_{ij} \leq u_{ij}(1 - y_{ij}) \quad \forall ij : (ij, kl) \in \mathcal{I} \tag{6}$$

$$x_{kl} \leq u_{kl}y_{ij} \quad \forall (ij, kl) \in \mathcal{I} \tag{7}$$

$\boldsymbol{\xi}$ is a binary vector of the defender's Stage 1 defenses, with $\xi_{ij} = 1$ indicating that arc ij is defended and Ξ being the set of feasible defenses. $\boldsymbol{\psi}$ is a binary vector of the attacker's Stage 2 attacks, with $\psi_{ij} = 1$ indicating that arc ij is destroyed and Ψ being the set of feasible attacks. $\mathbf{x}$ is the vector of flows set by the defender in Stage 3, while $\mathbf{s}$ and $\mathbf{y}$ are slack and binary indicator variables needed to enforce the binary input dependencies.

Objective (1) is the cost of the final flow network at the end of Stage 3. (2) are standard network flows conservation constraints. (3) enforce the fact that arcs defended in Stage 1 cannot be attacked in Stage 2, while (4) enforce the fact that arcs destroyed in Stage 2 cannot carry flow in Stage 3. Constraints (5)–(7) enforce the binary input dependencies, and result in a child becoming unusable if there is shortfall in delivery of flow to its parent.

Solution Algorithms

The overall trilevel model can be viewed as a bilevel program whose upper level is the defender's Stage 1 problem. Its lower level is, itself, another bilevel program whose upper level is the attacker's Stage 2 problem and whose lower level is the defender's Stage 3 recourse.

Cutting Plane Algorithm for the Bilevel Subproblem

The Stage 2–3 bilevel subproblem can be shown to be equivalent to

$$\max_{\boldsymbol{\psi}\in\Psi,\,\rho} \quad \rho \tag{8}$$

$$\text{s.t.} \quad \psi_{ij} \leq 1 - \xi_{ij} \quad \forall ij \in \mathcal{A} \tag{9}$$

$$\rho \leq \sum_{ij\in\mathcal{A}} c_{ij}x_{ij} + M \sum_{ij\in\mathcal{A}} \psi_{ij} \mathbb{1}_{>0}(x_{ij}) \quad \forall \mathbf{x} \in S \tag{10}$$

where S is the set of all feasible flows $\mathbf{x}$ in Stage 3. In constraints (10), M is a very large constant and $\mathbb{1}_{>0}(x)$ is a binary indicator function equal to 1 if $x > 0$ and 0 otherwise, which ensures that ρ is bounded above only by defender objectivesfor which the response flow $\mathbf{x}$ is feasible for the current attack vector $\boldsymbol{\psi}$.

This is a single MILP rather than a bilevel program, but it includes an intractable number of constraints due to the large number of possible Stage 3 flows in set S. In practice this can be solved using a *cutting plane algorithm* by relaxing constraints (10) to include only a subset of S.

Cutting Plane Algorithm for Stage 2–3 Bilevel Subproblem

$S \leftarrow \emptyset$ // relaxed constraint set
$\rho_2 \leftarrow -\infty$ // best Stage 2 objective value
$\rho_3 \leftarrow \infty$ // best Stage 3 objective value
$\boldsymbol{\psi} \in \Psi$ // any random initial attack
repeat
  Solve the Stage 3 problem for the current attack $\boldsymbol{\psi}$
  Set ρ_3 and $\mathbf{x}$ as its optimal objective and solution, respectively
  $S \leftarrow S \cup \{\mathbf{x}\}$ // add cut
  Solve the Stage 2 problem for the new constraint set S
  Set ρ_2 and $\boldsymbol{\psi}$ as its optimal objective and solution, respectively
until $|\rho_2 - \rho_3| \leq \epsilon$

The objectives from both stages in the main loop bound the true maximum value of (8), so the algorithm can be terminated when the optimality gap falls below a desired tolerance ϵ, returning the optimal attack $\boldsymbol{\psi}$ for the current defense $\boldsymbol{\xi}$.

Dual Algorithm and Linear Relaxation

Previous work shows that the *linear program (LP) relaxation* of the binary input dependence model produces quantitatively similar results while being significantly simpler to solve exactly [1]. The Stage 2–3 bilevel subproblem of the LP relaxation can be solved in a single step using the *dualize and combine* method, replacing the Stage 3 response problem with its own dual and combining both levels into a single maximization. This yields

$$\max_{\substack{\boldsymbol{\psi}\in\Psi \\ \boldsymbol{\mu},\boldsymbol{\eta}\geq\mathbf{0},\,\boldsymbol{\lambda}}} \sum_{i\in\mathcal{N}} b_i\lambda_i - \sum_{ij\in\mathcal{A}} u_{ij}\eta_{ij}$$

$$\text{s.t.} \quad \psi_{ij} \leq 1 - \xi_{ij} \quad \forall ij \in \mathcal{A}$$

$$c_{ij} + M\psi_{ij} - \lambda_i + \lambda_j + \eta_{ij} \geq 0 \quad \forall ij \in \bar{\mathcal{A}}$$

$$c_{ij} + M\psi_{ij} - \lambda_i + \lambda_j - \frac{u_{kl}}{u_{ij}}\mu_{ij}^{kl} + \eta_{ij} \geq 0 \quad \forall ij : (ij, kl) \in \mathcal{I}$$

$$c_{ij} + M\psi_{ij} - \lambda_i + \lambda_j + \mu_{kl}^{ij} + \eta_{ij} \geq 0 \quad \forall ij : (kl, ij) \in \mathcal{I}$$

where $\bar{\mathcal{A}} \subseteq \mathcal{A}$ is the set of arcs not involved in any interdependencies and $\boldsymbol{\mu}$, $\boldsymbol{\eta}$, and $\boldsymbol{\lambda}$ are the dual variables for the Stage 3 subproblem. This is a single MILP that need be solved only once, unlike the cutting plane algorithm which requires many iterative solutions.

Cutting Plane Algorithm for the Upper Level

The overall trilevel program can then be solved by applying a cutting plane algorithm analogous to that shown above. The relaxed constraint set is defined using a subset of observed attacks $\boldsymbol{\psi}$ which is added to in each iteration. The algorithm terminates and returns the optimal defense $\boldsymbol{\xi}$ when the optimality gap falls below a desired tolerance.

Computational Trials

The above solution algorithms were implemented in Python [4], calling CPLEX to solve the various LP and MILP subproblems involved in each step. Small test networks were generated using the NETGEN random flow network generator [2], augmented to select random arcs leading into demand nodes to act as parents for child arcs chosen randomly from elsewhere in the network. Networks were filtered to ensure that no feasible attack could disconnect the network. Each network had 24 nodes and 192 arcs. The defender was allowed to defend 10 arcs while the attacker was allowed to destroy 4.

The trilevel optimization model was then (approximately) solved on the test network in three ways:

- **MILP-CP:** binary input dependence solved via cutting plane
- **LP-CP:** LP relaxation solved via cutting plane
- **LP-D:** LP relaxation solved via dual algorithm

All three algorithms typically produced nearly identical optimal defense vectors $\boldsymbol{\xi}$, with the linear relaxations resulting in optimality gaps with means of less than 0.02% over all trials. Table 1 shows the computation times for the solution algorithms.

Int. Density		Solution Method		
		MILP-CP	LP-CP	LP-D
25%	Mean	2660.50	2496.67	861.52
	Std. Dev.	1268.06	1303.24	574.10
	n	19	19	19
75%	Mean	1792.62	1707.48	735.27
	Std. Dev.	1979.51	2031.60	893.55
	n	22	22	22

Table 1: Solution times (seconds) for each solution algorithm. *Int. Density* describes the fraction of demand nodes that act as the source of an input dependence.

Discussion and Future Work

These preliminary results show that there is potential for the use of the linear interdependence model in approximation algorithms for problems based on the binary dependence model. The solution algorithms based on linear input dependence appear to produce nearly identical results to those of the much more computationally expensive binary input dependence model.

The main work remaining for this study will consist of conducting more and more varied computational trials. Tests will also be run on networks with different topologies and with different distributions of interdependencies. In addition, it would be valuable to compare the optimal defenses with defenses obtained from simpler heuristics to quantify the value of an exact solution.

References

[1] H. Kaul and A. Rumpf. A linear input dependence model for interdependent networks. *European Journal of Operational Research*, 302(2):781–797, 2022. doi:10.1016/j.ejor.2022.01.020

[2] D. Klingman, A. Napier, and J. Stutz. NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems. *Management Science*, 20(5):814–821, 1974. doi:10.1287/mnsc.20.5.814

[3] E. E. Lee II, J. E. Mitchell, and W. A. Wallace. Restoration of services in interdependent infrastructure systems: A network flows approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1303–1317, 2007. doi:10.1109/TSMCC.2007.905859

[4] A. Rumpf. Trilevel Network Interdiction Game with Linear Interdependencies. GitHub repository, 2020. URL https://github.com/adam-rumpf/trilevel-mcnfli