2022

Mixed-integer Programming Methods for Modeling and Optimization of Cascading Processes in Complex Networked Systems

Cheng-Lung Chen
University of Central Florida

MIXED-INTEGER PROGRAMMING METHODS FOR MODELING AND OPTIMIZATION OF CASCADING PROCESSES IN COMPLEX NETWORKED SYSTEMS

by

CHENG-LUNG CHEN
M.S. The Ohio State University, 2017
M.S. Arizona State University, 2015
B.B.A Soochow University, Taiwan, 2009

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2022

Major Professor: Vladimir Boginski

ABSTRACT

Dynamics and growth of many natural and man-made systems can be represented by large-scale complex networks. Entity interactions and community interconnections within complex networks increase the level of difficulty for the investigation on structural network properties such as robustness, vulnerability and resilience. In this dissertation, we develop methodologies based on mixed-integer programming techniques to solve challenging optimization problems that model cascading processes in complex networked systems. In particular, we seek to provide decision making recommendations for problems related to different types of cascading processes in networks commonly considered in a variety of applications: *interdependent infrastructure networks* and *social networks*. In the first part, we propose a novel optimization model to enhance the resilience against cascading failure by mitigation and restoration in interdependent networks. We derive a polynomial class of valid inequalities from the cascading constraints and reformulate the substructure that describes capacity restriction to guarantee integral solutions. The computational experiments illustrate that our strengthened formulation outperforms the default setting of a commercial solver on all tested instances. Next, we study the least cost influence maximization problem that arises in social network analytics. We investigate the polyhedral properties of a substructure that is a relaxation of the mixed 0-1 knapsack polyhedron. We give three exponential class of facet-defining inequalities from this substructure and an exact polynomial time separation algorithm for the inequalities. In addition, we propose another new class of strong valid inequalities that dominates the cycle elimination constraints. Through the computational experiments, we demonstrate that a delayed cut generation algorithm that exploits these inequalities is very effective to solve the problem under different settings of network size, density and connectivity.

ACKNOWLEDGMENTS

This dissertation would certainly not exist without the continuous guidance and support from my advisor Dr. Vladimir Boginski. He has been extremely patient, generous, and supportive to me academically, professionally, and personally as well as helping me completing this dissertation. He always responses my concerns and questions with positive attitude following thoughtful and constructive feedback in our regular meetings. Learning from him, I was able to overcome the challenge and complete the projects that described in this dissertation. I sincerely appreciate his offer of research assistantship and conference travel grant during the past five years, which allow me to focus on my research projects, attend numerous conferences and workshops to present our work, and expand my professional network. Additionally, his nomination and recommendation letters helped me to win several awards, including the Gerald R. Langston Endowed Scholarship, the David T. & Jane M. Donaldson Memorial Scholarship, and the Bonder Award from Seth Bonder Foundation for INFORMS Doctoral Student Colloquium 2021. I could not ask for a better advisor. I would like to extend my gratitude to professors Qipeng Phil Zheng, Adan Vela and Müge Yayla-Küllü for serving as my dissertation committee. Their constructive suggestions and insights to the scope of my research help me to improve this dissertation as well as my professional growth. Special thanks to Dr. Zheng who introduced Applied Operations Research (AOR) Lab to me and helped me to get on the track during my first year at UCF.

My journey from a business management undergraduate student to an operations research Ph.D. took several detours and I often refer it as a longest path problem. I would like to take time to acknowledge those who have helped and inspired me along the way. My first operations research course is taught by professor Jinshyang Roan in a spreadsheet modeling class at Soochow University, Taiwan. I was immediately fascinated by the magic of making better decision by mathematical optimization. I thank him for encouraging me to

pursue graduate studies in operations research in United States even though I was considered by others not having sufficient math background to do so. My first academic research experience is supervised by professors Muhong Zhang and Srimathy Mohan at Arizona State University. I thank Muhong and Srimathy for offering this opportunity to work with them. This experience not only enlightens my interest in mixed-integer programming problems but also serves the origin of my career in operations research. I further obtain in-depth knowledge and establish solid theoretical foundation for mixed-integer programming and stochastic optimization from professors Simge Küçükyavuz and Güzin Bayraksan at Ohio State University. Both are wonderful teachers and devoted researchers in their own expertise, and I have learned a great deal from them. Moreover, I would like to thank Dr. Eduardo Pasiliao for providing me the opportunity to participate the research project at Mathematical Modeling and Optimization Institute of Eglin Air Force Research Lab. I am motivated by the fact that he is a dedicated researcher and continually has a lot of groundbreaking research ideas. While at AFRL, I had the chance to meet with Dr. Alexander Veremyev and Dr. Alexander Semenov for some stimulating mathematical and technical discussions, which transforms to the content of my first journal publication. I appreciate their help. I would also like to thank Dr. Hongbo Sun from Mitsubishi Electric Research Laboratories who recruited me as his research intern when I just lost my summer internship from a company due to COVID-19. I've obtained many industrial-scale knowledges of power grid and electricity market from him, and I benefited greatly from writing research papers with him.

My time at UCF was made enjoyable because of my wonderful AOR lab cohorts, including Shimaa Al-Quradaghi, Mengnan Chen, Marwen Elkamel, Zhecheng Qiang, Guanxiang Yun and Viacheslav Zhygulin for creating a productive environment of intellectual growth that full of peer support, suggestions, and cultural exchange. I also appreciate all the administrative support from Liz Stalvey, Ricardo Scuotto, Claudia Sardon, Shirley Wang, Luke Fuller and Samuel Miranda from Department of Industrial Engineering and Management

Systems.

I would like to acknowledge the financial support I received for completing this dissertation, including National Science Foundation award EFMA 1441231, Air Force Research Laboratory award FA8651-16-2-0009, the Marian and Gary Whitehouse IEMS Foundation from Department of Industrial Engineering and Management Systems, Graduate Presentation Fellowship from UCF graduate studies and Conference Registration & Travel (CRT) fund from UCF Student Government Association.

Finally, I want to thank the support from my dearest family and friends from Taiwan, especially my two very supportive sisters Ju-Lin and Ching-Hui. Both my grandparents and parents did not have chance to receive opportunities for better education. Ju-Lin is the first person in our family who came to United States to pursue her undergraduate study. She has shared many of her insightful opinions to help me survive and get involved in United States. Words cannot express how grateful I am to have a wonderful sister like her. Despite thousand miles away, my family always believe in me without hesitation whenever I have doubt about myself.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Complex networked systems are ubiquitous in the modern world as they arise in many aspects of human activities and natural systems. For instance, physical *infrastructure systems*, which provide necessary services to households, include electrical power grid, water and gas distribution networks. Infrastructure systems also include various types of transportation/supply networks (e.g., airports, harbors, highways, railroads), communication networks (e.g., internet, phone), etc. These networks are often *interdependent*, which makes them susceptible to cascading failures, where a failure in one network may cause failures in other networks. Furthermore, *social networks* are another important type of complex networked systems that have been receiving increased attention in the recent decades. In particular, this is associated with the growth of social media network platforms, which nowadays serve as the means of spreading/propagation of news, opinions, and influence in a cascading fashion.

These large-scale networked systems admit abstract mathematical representations as graphs, where nodes represent the elements of a system and a set of connecting edges represent the existence of a relation or interaction between those elements. The collective behavior exhibited by large-scale networks may be different from individual behavior of single units (nodes), which increases the difficulty of analyzing their properties and decision-making for optimal operations.

One particular factor that affect the analytical complexity of different large-scale networked systems is the *interdependence* between them, where these networks engage in complex interactions and information exchange via a flexible communication infrastructure, result in nodes in one network require support from nodes in another network. A very common example for interdependence in diverse infrastructures can be seen in the coupled components among water distribution, electrical power and transportation systems, where the generation of electricity requires water for cooling and fuel replenishment supplied from

transportation system; on the other hand, the power station provides electricity for pumping and control in water system and signaling in transportation system.

Despite the emergence of *network science* tools that aim to study the architecture and topological characteristics of complex networks, those tools are not always sufficient to uncover the complete picture of non-trivial structures and properties of the underlying architecture. The nature of interconnections between complex networks leads to the phenomena of *cascading processes*, which is crucial to the dynamics of system resilience and vulnerability. Many researchers have been investigating the impact and behavior of these percolation-like processes in various types of networks. For example, those concerned with public health policy in the epidemiology field are interested in determining how fast and how small of an outbreak that could lead to pandemic. Similarly, many studies on information spreading in social media have been engaged in estimating the coverage and propagation time of rumors, fake news or political propaganda across social media. There are also researchers who are concerned with robustness and vulnerability of infrastructure systems, where the cause of components failure is either random breakdown or targeted attacks.

In this dissertation, we investigate optimal decision-making problems arising in complex networked systems. Rather than just describing topological and structural characteristics of networks, we are interested in *optimization* problems that are critical to the functionality of complex networks under the influence of cascading processes, ensuring they are resilient to failures/attacks, reliable to operate and scalable to support changing needs and population demographics. We develop mathematical representations of these problems that contain logical conditions and time dynamics of cascading processes under the framework of mixed-integer programming. To develop exact and efficient algorithms for solving the corresponding mixed-integer programming problems, we study their polyhedral structures and propose valid inequalities, strong formulations and cutting plane based exact algorithms to strengthen the relaxation, speed up the computational time and obtain smaller or even

zero optimality gap.

In the remainder of this section, we provide a brief review of basic terminologies in graph theory and mixed-integer programming with emphasis on polyhedral theory. We then introduce the research scope and outline of this dissertation.

1.1 Graphs and Networks

A graph is defined in mathematical terms as follows:

Definition 1. *A graph, more specifically an undirected graph G consists of a set of vertices (nodes) $V = \{v_1, \ldots, v_n\}$ and a set of edges E, for which we write $G = (V, E)$. Each edge $e \in E$ is said to join two vertices, which are called its end points. If e joins $v_1, v_2 \in V$, we write $e = (v_1, v_2)$. Vertex v_1 and v_2 in this case are said to be adjacent. Edge e is said to be incident with vertices v_1 and v_2, respectively.*

In various cases, G is either vertex-weighted or edge-weighted, hence a function $f : V \to \mathbb{R}$ or $g : E \to \mathbb{R}$ is specified.

Definition 2. *The number of edges incident on a vertex v is called the degree of the vertex. We denote the degree of vertex v to be $d_G(v)$. A vertex v is called isolated if $d_G(v) = 0$, or pendant if $d_G(v) = 1$. Note that the summation of degrees in an undirected graph is always an even number, that is, $\sum_{v \in V} d_G(v) = 2|E|$.*

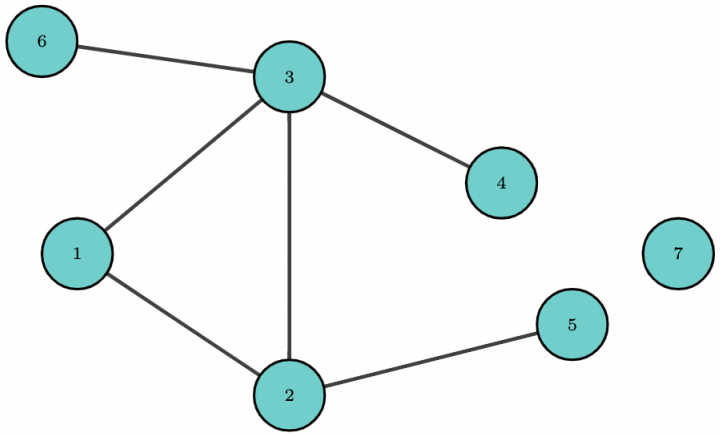


Figure 1.1: An undirected graph with pendant vertexes 4,5,6 and isolated vertex 7.

In some cases, the precise order of two vertices connected by an edge is important. We often refer to edge with direction as arc. Typical examples include the one-way road in transportation network, routing in vehicle scheduling network, journal article citation networks and food webs between different species in nature. Formally, the directed graph can be defined in the following way:

Definition 3. *A directed graph $D = (V, A)$ consists of a set of vertices $V = \{v_1, \ldots, v_n\}$ and a set of arcs A, where elements of A are distinct ordered pairs of distinct elements of V, and are called arcs or directed links.*

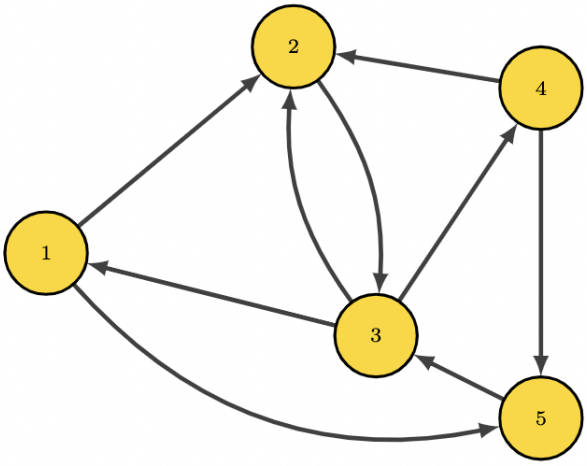


Figure 1.2: A directed graph.

In many practical situations, it is often necessary to describe the neighbors of a vertex. In graph-theoretical terms, the neighbors of a vertex v are formed by the vertices that are adjacent to v, or, in other words, those vertices to which v has been joined by means of an edge. We give precise formal mathematical notations as follows.

Definition 4. *For any graph $G = (V, E)$ and a vertex $v \in V$, the neighbor set N_v of v is the set of vertices adjacent to v except v itself, that is $N_v = \{u \in V : (u, v) \in E\}$.*

Definition 5. *A graph $G' = (V', E')$ is called a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. We say that G' is the subgraph of G induced by V'.*

Definition 6. *A path of G is a sequence of consecutive edges $e_1, e_2, \ldots, e_m \in E$ of the form: $e_1 = (v_1, v_2)$, $e_2 = (v_2, v_3)$, $\ldots$, $e_m = (v_m, v_{m+1})$. The path is a cycle if $v_1 = v_{m+1}$. The path/cycle is called elementary if an edge is never used twice, i.e., $e_i \neq e_j \, \forall i \neq j$. A path/cycle is called simple if it never visits the same vertex twice except for $v_1 = v_{m+1}$ in the case of a cycle.*

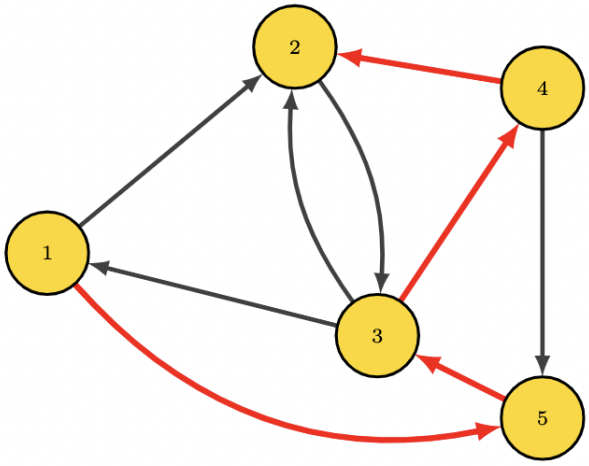


Figure 1.3: A directed path 1-5-3-4-2 on a directed graph.

The optimal solution of many graph/network optimization problems is indeed a subgraph of the original network with certain properties. For example, in the application of routing problem or influence propagation within a network, the optimal solution is acyclic, i.e., no cycles are allowed so that the direction is clearly identified by the optimal path.

1.2 Mixed-integer Programming

A generic mixed-integer (linear) programming (MIP) is a mathematical optimization model of the form

$$z^* = \min\{c^\intercal x : Ax \geq b, x \in \mathbb{R}_+^{n-k} \times \mathbb{Z}_+^k\} \tag{1.1}$$

where $c \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$ and we let $\mathbb{Q}, \mathbb{R}, \mathbb{Z}, \mathbb{B}$ represent the set of rational numbers, real numbers, integers and binaries, respectively. The set $P := \{x \in \mathbb{R}_+^{n-k} \times \mathbb{Z}_+^k : Ax \geq b\}$ is called the feasible set of MIP. The goal is to find a set of mixed-integer points $x^* \in \mathbb{R}_+^{n-k} \times \mathbb{Z}_+^k$ lie in the feasible set P that minimize the linear function $c^\intercal x^*$. MIP

provides an broad framework that a very large number of decision making problems can modeled in this form, including production planning, telecommunication network design, routing and staffing in airline scheduling and combinatorial auctions, just to name a few. Mixed-integer programming problems are in general $\mathcal{NP}$-hard. The difficulty of solving them to optimality increases exponentially as the problem size grows. It is common to consider a *relaxation* problem of MIP in order to design an efficient algorithm. By relaxing the integrality restriction on some variables, we obtain the *linear programming (LP) relaxation* of MIP in the form

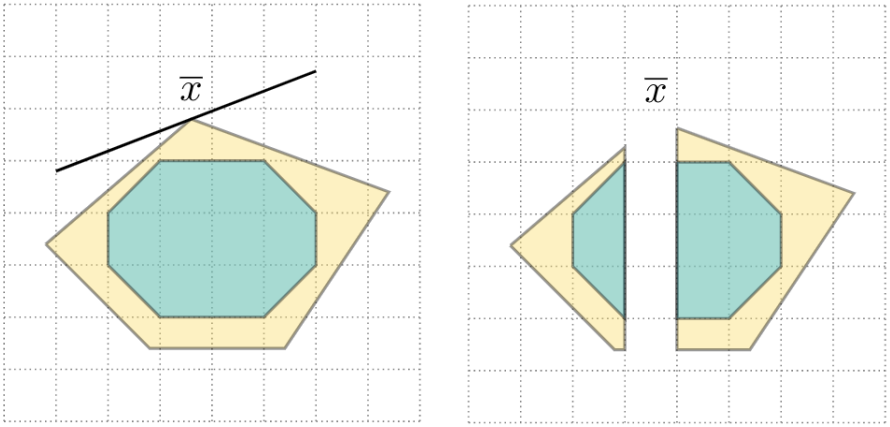
$$z_0^* = \min\{c^\intercal x : x \in P_0\} \tag{1.2}$$

where $P_0 := \{x \in \mathbb{R}^n_+ : Ax \geq b\}$. The LP relaxation is easier to solve than MIP since linear programming is polynomial solvable. In addition, the objective function value z_0 provides a lower bound for z^* as $P \subseteq P_0$ and typically this containment is strict. A fundamental idea behind most methods to solve MIP is the *branch-and-bound* algorithm, which repeatedly solve the LP relaxation, obtain the lower bound and decide to further branch the feasible region if integer feasible solution is not found. Another method relies on relaxation to solve MIP is the *cutting plane method*. The combination of these two methods results in the *branch-and-cut algorithm*, which has been implemented in the core of state-of-the-art MIP solvers nowadays. With the advance development in computing power, many difficult MIP problems can be solved to near optimal or even optimal in a reasonable amount of time.

1.2.1 Branch-and-Bound Algorithm

The branch-and-bound algorithm is an implicitly enumeration that utilizes the concept of divide-and-conquer. The algorithm begins with solving the LP relaxation and obtain a solution $\overline{x}$. If the solution is fractional, i.e., $\overline{x} \neq \mathbb{R}^{n-k}_+ \times \mathbb{Z}^k_+$, then we update the lower

bound information with the objective function value. Next, we select a particular $\overline{x}_j$ for $j \in \{1, \ldots, k\}$ to create two additional subproblems (nodes) by adding $x_j \leq \lfloor \overline{x}_j \rfloor$ and $x_j \geq \lceil \overline{x}_j \rceil$ into the LP relaxation. This process is called branching and it is executed at every node. We illustrate this concept in Figure 1.4. Nodes are pruned when a feasible solution $\overline{x} \in \mathbb{R}_+^{n-k} \times \mathbb{Z}_+^k$ is found and the best upper bound is replaced with the objective function value, or the lower bound exceeds current upper bound, or when the solution is infeasible. The algorithm takes finite but exponentially many iterations to terminate when all the available subproblems are explored.



(a) Initial problem (b) Adding $x_1 \leq \lfloor \overline{x}_1 \rfloor$ and $x_1 \geq \lceil \overline{x}_1 \rceil$

Figure 1.4: Divide the feasible region in branch-and-bound

1.2.2 Cutting Plane Algorithm

We first explore the geometry of polyhedra and use it to develop the concept of cutting planes. The polyhedron Q is a set

$$Q = \{x \in \mathbb{R}_+^n : a_i x \leq b_i, i \in [1, r]\} \tag{1.3}$$

for some finite r, where $[1, r]$ denotes the set $\{1, \ldots, r\}$. A geometric representation of Q can be regarded as the intersections of r half-spaces defined by the inequalities $a_i x \leq b_i$ The dimension of Q is the minimum dimension of affine subspace containing Q. In other word, the dimension of Q is p if there are at most $p + 1$ affinely independent points contained in Q. We say Q is full-dimensional if $p = n$.

The minimal description of polyhedron Q is defined as the smallest convex set containing Q, or the convex hull of Q, denoted by $\text{conv}\,(Q)$. Furthermore, $\text{conv}\,(Q)$ is a rational polyhedron, the extreme points of $\text{conv}\,(Q)$ and Q coincide.

We know optimization a linear function over a mixed-integer set is equivalent to optimizing its convex hull from the fundamental theorem of mixed-integer programming. When $\text{conv}\,(Q)$ is known, then the optimization problem of MIP over the polyhedron Q is reduced to solving the linear programming problem. Nevertheless, obtaining an explicit minimal description of $\text{conv}\,(Q)$ is a challenging task as the number of inequalities defining $\text{conv}\,(Q)$ can be very large. One possible way to eliminate fractional solution and acquire a set close to $\text{conv}\,(Q)$ is to generate a small set of *valid inequalities* progressively.

Definition 7. *An inequality $\pi^\intercal x \leq \pi_0$ is a valid inequality for Q if $\pi^\intercal x \leq \pi_0$ for all $x \in Q$.*

We need to give a precise definition on the *strength* of valid inequalities as it is possible that there exists exponentially many valid inequalities for a polyhedron.

Definition 8. *If $\pi^\intercal x \leq \pi_0$ and $\eta^\intercal x \leq \eta_0$ are two valid inequalities for Q, then we say $\pi^\intercal x \leq \pi_0$ dominates or is stronger than $\eta^\intercal x \leq \eta_0$ if $Q \cap \{x \in \mathbb{R}^n_+ : \pi^\intercal x \leq \pi_0\} \subset Q \cap \{x \in \mathbb{R}^n_+ : \eta^\intercal x \leq \eta_0\}$.*

An alternative way to verify the dominance relationship between valid inequalities is to check if there exists $\gamma > 0$ such that $\pi \geq \gamma\eta$, $\pi_0 \leq \gamma\eta_0$ and $(\pi, \pi_0) \neq \gamma(\eta, \eta_0)$.

Definition 9. *The inequalities with dimension $n - 1$ that describe $\text{conv}\,(Q)$ are called facet-defining inequalities.*

Given a facet-defining inequality $\pi^\intercal x \leq \pi_0$, the corresponding facet is the set

$$H = P \cap \{x \in \mathbb{R}^n_+ : \pi^\intercal x = \pi_0\}. \tag{1.4}$$

The method of combining the steps of iteratively solving LP relaxation and adding valid inequalities is referred to as cutting plane method. Given a fractional solution $\overline{x}$ from LP relaxation, we identify a valid inequality $\pi^\intercal x \leq \pi_0$ that cuts off this solution, i.e., $\pi^\intercal \overline{x} > \pi_0$ by introducing this inequality back to the LP relaxation. We continue this process until an integer feasible solution is found or no valid inequalities can be identified.

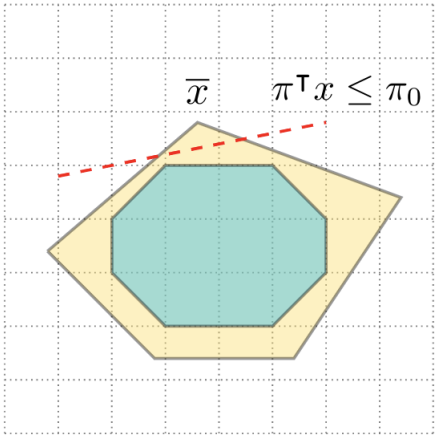


Figure 1.5: $\pi^\intercal x \leq \pi_0$ cut off $\overline{x}$

The most crucial question in designing cutting plane algorithm is how to identify a strong valid inequality that separates the current fractional solution $\overline{x}$ from the feasible region with maximum distance. This question is referred to as the *separation problem*. Consider a family of valid inequalities $\mathcal{F}$, the separation problem given a fractional solution $\overline{x}$ answers the following two questions:

1. there exists a valid inequality $\pi^\intercal x \leq \pi_0$ from $\mathcal{F}$ violated by the solution $\overline{x}$ with distance

$\xi = \pi^\intercal \overline{x} - \pi_0$,

2. all inequalities in $\mathcal{F}$ are satisfied by this solution $\overline{x}$ without violation.

Following the theorem of equivalence of optimization and separation, the separation problem is polynomial-time solvable if and only if the optimization problem is polynomial-time solvable. In addition, if there are finite number of valid inequalities in $\mathcal{F}$ or the inequalities in $\mathcal{F}$ plus original constraints are enough for the description of conv (Q), then the cutting plane method will terminate in finite steps. We summarize the cutting plane method in the following Algorithm 1.

Algorithm 1: Cutting Plane Algorithm for MIP

initialization: solve LP relaxation and get $x = \arg\min\{c^\intercal x : x \in P_0\}$;

while $x \neq P$ **do**

  solve the separation problem ;

  if $\pi^\intercal x \leq \pi^0$ *for all* $(\pi, \pi_0) \in \mathcal{F}$ **then**

    $x^* \Leftarrow x$, **break** ;

  else

    Separation gives (π, π_0) such that $\pi^\intercal x > \pi_0$;

    $P_0 \Leftarrow P_0 \cap \{x \in \mathbb{R}^n_+ : \pi^\intercal x \leq \pi_0\}$;

    solve $x = \arg\min\{c^\intercal x : x \in P_0\}$;

  end

end

Return x^* and $c^\intercal x^*$;

1.2.3 Branch-and-Cut Algorithm

The branch-and-cut algorithm provides a desirable scheme to alleviate the computational inefficiency of both branch-and-bound algorithm and cutting plane method, where the former

is suffered from the exponentially many nodes needed to visit in the branch-and-bound tree and the latter has exponentially many valid inequalities to identify via a $\mathcal{NP}$-hard separation problem. In branch-and-cut, the separation problem is *dynamically* executed to identify a whether a valid inequality cut off this fractional solution at every node. In practice, the separation is often done by heuristics in order to reduce the solution time. The benefit of this combination is to possibly reduce the size and nodes and improve the bound, with the expense of finding valid inequalities together to speed up the overall computational time for solving a MIP.

1.3 Research Scope and Outline

In this dissertation, we first study cascading processes that induces malfunction of components from two different interdependent systems. Particularly, we study how to optimally mitigate the malicious cascading failure and restore failed components in two-layered interdependent networks. We conduct polyhedral analysis on one substructure of the formulation and derive valid inequalities to strengthen the model. For another substructure, we give a convex hull where all extreme points are integral and replace the substructure with this strong formulation. We show that our proposed mitigation and restoration strategies are beneficial to lessen the scale of cascading failure. Our mathematical model is also very flexible to be extended to multiple layered interdependent networks, different types of network operations and partial capacity reduction. In addition, our computational results demonstrate the polyhedral enhancement is very effective in solving small to medium sized interdependent networks with interdependent links density from 10% up to 40%.

Next, we investigate a static least cost influence maximization problem in social network where the influence spreading is motivated by partial incentives given to a subset of users initially. The propagation mechanism follows the linear threshold model, where a user

will propagate the information if and only if the summation of peer influence plus the incentives is exceed the user's threshold. We study the node propagation substructure and exploit the valid inequalities from mixed 0-1 knapsack polyhedron hidden in this substructure. We give three new class of valid inequalities and exact polynomial time separation algorithms for them. We show that, the valid inequalities from mixed 0-1 knapsack polyhedron can also be separated efficiently from our proposed separation algorithm. In addition, we investigate the polyhedral properties for the problem when cycles are exist. We show that the problem can be solved in polynomial time on a simple cycle. For arbitrary graphs, we propose another new class of strong valid inequalities the dominates cycle elimination constraints. Our computational experiments illustrate the efficacy of the proposed valid inequalities via a delayed cut algorithm. For the cas of equal influence weights, we present a convex hull description for the problem with equal influence weights and 100% adoption rate on a tree in the natural space of incentive, arc propagation and activation variables. We prove that such description requires only polynomial number variables and constraints. Finally, we demonstrate the effectiveness of proposed valid inequalities through extensive computational experiments by comparing the delayed cut generation algorithm with different settings and an alternative formulation of the problem.

The remainder of this dissertation is organized as follows. In chapter 2, we study the failure mitigation and restoration in interdependent networks via mixed-integer optimization. In chapter 3, we investigate the polyhedron of the static least cost influence maximization problem in social networks. Finally, in chapter 4, we conclude this dissertation and discuss possible future work extensions.

CHAPTER 2: FAILURE MITIGATION AND RESTORATION IN INTERDEPENDENT NETWORKS VIA MIXED-INTEGER OPTIMIZATION

This chapter is based on [12]. In this chapter, we propose a new optimization model for determining optimal mitigation and restoration strategies for coupled interdependent networks in the context of preserving and/or restoring the maximum flow through the entire networked system, subject to cascading node failures that may be caused by disruptions of a subset of "seed nodes" at an initial time step. Previous related studies mainly focused on "static" strategies to mitigate cascading failures. However, our model allows one to identify "dynamic" strategies for step-by-step failure propagation, given initial seed node disruptions. Moreover, the proposed model accounts for backup arc capacity and node fortification to mitigate the impact of further failure cascades on network performance. The objective is to restore network performance during a finite recovery planning horizon at total minimal cost. We formulate this problem by mixed-integer optimization, and derive valid inequalities using the substructure of the problem. We report a summary of computational experiments to demonstrate the strength and effectiveness of the inequalities when compared to solving the problem with a commercial optimization solver.

2.1 Introduction

Network optimization problems that consider certain performance characteristic(s) of a given "single-layer" network have been studied extensively over the past decades. However, many complex systems consist of several distinct networks, or "layers", that interact with each other, thus forming multi-layer *interdependent* networks. In many applications, these interdependencies represent physical or virtual connections between network layers necessary

for the operation or delivery of information/commodities which ensure the functionality of the entire networked system. Coupled (two-layer) interdependent networks often arise in the domain of critical infrastructure systems, which provide necessary services such as water supply, transportation, communication, energy, etc. For instance, power grid networks provide electricity to every node in a communication (i.e., SCADA) network. The nodes in a communication network, on the other hand, send control signals back to the power grid. Other examples of coupled interdependent networks include hybrid social-physical networks [9], power-water distribution networks [61], and law enforcement [7]. Each type of a network has its distinct features and characteristics. Their closely connected and inseparable interactions make the analysis of general network properties, such as robustness, reliability, and vulnerability, more challenging than in the case of a single-layer network.

The phenomena of *cascading failures* has received increasing attention, especially in interdependent infrastructure literature, since these low-probability high-impact events affect the vulnerability of interdependent networks and produce substantial social and economic impacts. Generally, cascading failures originate from a small fraction of "activated" (disrupted or attacked) seed nodes which then propagate to different network layers via interdependent links over time. In the context of a social network, the seed nodes are the key opinion leaders which try to spread certain information in order to achieve spin control at a desired level among different groups. In critical infrastructure systems, these initial activated seed nodes could be regarded as damaged system components after either a random disruptive event (i.e., severe weather) or a targeted deliberate attack.

To illustrate that cascading failures triggered by seed nodes could propagate very rapidly and lead to catastrophic consequences, we list several examples of well-known large-scale blackouts around the world. One commonly shared cause of transmission line loss, overheating and regional power outrage is tree flashover. Cases like this include The 1996 Western North America blackouts [17], The 2003 U.S.-Canadian blackout[42], and the 2003

Italy blackout [57] . Another possible reason is deliberate cyberattack. For instance, the 2015 Ukrainian blackout was due to unauthorized intrusion into the SCADA system of the electricity distribution company [32]. Finally, the 2016 South Australia blackout was attributed to severe storms that damaged transmission towers and brought about 850,000 customers to lose their power supply [4]. For other examples of cascading blackouts in electric power grid, we refer the reader to [56].

Considering the fact that cascading failures significantly deteriorate network functionality, various studies of interdependent networks with cascading failures have focused on investigating the robustness of such networks (e.g., when random nodes or arcs are removed and/or directional changes occur between interdependent links) primarily through simulation experiments (see [16, 8, 28, 62, 19, 45, 47]).

One research direction in addressing adverse impacts of disruptive events on infrastructure networks is to consider restoration activities. A prompt planning for restoration can potentially prevent major outages and maintain stable operation of infrastructure systems. There exists a large body of literature on network restoration of single-layer networks in humanitarian operations (see [10] for a recent comprehensive review).

Similarly to single-layer networks, it is necessary to consider a time-expanded model for restoration in interdependent networks as the resources for restoration are generally scarce. A useful model would potentially guide the decision maker with respect to the scheduling of restoration activities. Lee et al. [33] propose a multi-commodity flow model that incorporates five types of infrastructure interrelationship. Specifically, they model interdependencies between the power system, telecommunication, and subway system of New York City using aforementioned definition. Without considering the time-index and restoration decision together in their model, the cascading failure is ignored and the restoration activities can not be determined from the optimal output of the model. The model can only be used as a guideline for restoration service after known disruption occurred on net-

work components. Sharkey et al. [53] introduce an integrated design and scheduling model to demonstrate the value of information sharing for restoring services provided by infrastructure systems with operational interdependencies. Gonzalez et al. [22] propose an interdependent network design problem for reconstructing partially destroyed infrastructure systems. Their model provides details on formulating functional, physical, and location interdependencies. However, none of these models have considered the impact of cascading failure to restoration activities. Nguyen and Sharkey [46] present a novel interdiction-based approach to determine the order and minimum number of components inspections required of a single infrastructure network after disruptive events, assuming customer self-report outrage data are available. Although their model does not provide restoration scheduling decision, it could be potentially improve the accuracy and efficiency of restoration for both single and multi-layer interdependent networks.

Research on developing *mathematical optimization* models (with guaranteed optimality of a solution) for interdependent networks with cascading failures have not been extensively addressed in the literature. Shen [54] considers two-stage stochastic programs for a single network and multiple interdependent networks subjected to random arc disruption. The mitigation is done by selectively disconnecting failed components instead of beforehand activities such as hardening components before disruption. Parandehgheibi et al. [48] design a two-phase control algorithm for a power grid and a communication network in order to redistribute power flow against cascading failures. However, they only consider initial node removal when implementing the algorithm, and it is unclear whether the algorithm will work if computing resources are limited and cascading failure spreads rapidly over time.Only a limited number of studies have included cascading failures into network optimization problems. Veremyev et al. [59] incorporate the cascading failure propagation processes into a mathematical optimization model. They sought to find the minimum-cardinality subset of "critical" nodes whose initial activation would isolate all nodes in both layers of an inter-

dependent network. Gillen et al. [20] propose a critical arc detection problem on a social network where the cascading influence propagation is based on linear threshold model. Their model seeks to identify which arcs are critical in the process of influence spreading.

Although researchers and practitioners are aware of the potential consequences of cascading failures, simulation results do not always provide information on the optimal strategies for managing interdependent networks under disruptive events. Optimization-based approaches are intended to provide such information; however, previously developed optimization models usually treat failed nodes/links as a "static" input. Cascading failures represent an ongoing process after they are triggered by a natural disaster or an adversarial attack. Therefore, effective strategies to increase the resilience of interdependent networks should consider "dynamic" mitigation actions in order to reduce the impact of cascading failures, as well as to establish a recovery plan to restore the network performance in a timely manner.

In this chapter, we introduce a new optimization model that takes into account node failures and not only captures the dynamics of failure propagation, but also simultaneously considers mitigation and restoration for interdependent networks. This work seeks to alleviate the impact of disruptions in interdependent networks via optimal mitigation and restoration strategies. To explicitly model how dynamic failure propagation exacerbates the network performance, we adopt the network flow model. This representation is without loss of generality as interdependent infrastructures, such as water, power, gas, and signal transmission can be regarded as networks with materials moving from source nodes to sink nodes that traverse capacitated arcs. For the ease of representation, we use the maximum flow problem as a "basis" for our models, since the maximum flow problem is often used in the context of operations of infrastructure, information, and other types of networked systems. Almoghathawi and Barker [3] also model interdependent infrastructure networks as maximum flow problem in each layer. They propose two component importance measure-

ments to examine the resilience of networks after disruption and restoration. They give four separate optimization models under different disruptive settings in order to identify critical components that affect resilience the most. However, they do not consider mitigation or the effect of cascading failure to network resilience in the model directly. Ahangar et al. [2] adopt multi-commodity flows to model the interdependencies between infrastructure systems. They show that their model is compact and can easily incorporate the type of interdependencies defined in [33]. However, they do not consider cascading failures, mitigation or restoration, as the objective is to maximize overall functionality of network operations.

To model mitigation and restoration, we assume that the decision-maker has a limited budget to execute mitigation and restoration activities. Specifically, mitigation consists of node fortifications and/or increasing arc backup capacity. After nodes are "fortified", they are assumed to be immune against disruption or unfavorable influences and thus will prevent future cascades. Related ideas about fortification against cascade propagation can be found in [21], where a single network critical node detection problem with arc weight uncertainty is proposed; however, that work does not consider capacity expansion or restoration. Arc capacity expansion allows the flow of networks to be redistributed so as to bypass failed components. These additional backup arc capacity can facilitate the possibility of flow redistribution through alternative routes. Restoration enables the node to change status from failed to functional. Note that during the recovery stage, our model also gives a recommendation for the node recovery sequence. As failure propagation is still going on while recovery activities are performed, it is critical to prioritize the recovery of nodes that would potentially cause further cascade propagation if not recovered.

To summarize, the main contributions of this chapter are:

- We propose the first mathematical optimization problem on coupled interdependent networks that simultaneously considers the effect of mitigation and restoration in the

presence of dynamic cascading failures. Moreover, we discuss how our model can potentially be extended to other related settings, including multi-layer networks and capacity degradation.

- We conduct polyhedral analysis from the two substructures of the mixed-integer optimization model and derive valid inequalities to strengthen the formulation.

- We compare the cost differences based on combinations of mitigation and restoration options and show the benefit of our proposed strategy to increase the resilience of interdependent networks. We also illustrate the efficacy of our valid inequalities through computational results.

2.2 Problem Statement

In this section, we consider a formal setup of an optimization problem formulated for a two-layer (coupled) interdependent network with considerations outlined in the previous section.

Throughout the chapter, let the finite recovery planning period be denoted by T, and also let $[i, j] := \{t \in \mathbb{Z} : i \leq t \leq j\}$. Consider two directed graphs, $G_1 = (N_1, A_1)$ and $G_2 = (N_2, A_2)$, where N_ℓ is the set of nodes and A_ℓ is the set of directed arcs for $\ell \in [1, 2]$. We define sets E_{12} and E_{21} as the interdependent link sets and $(i, j) \in E_{12}$ indicates the operation of a node i in N_1 depending upon the input of node j in N_2. A similar definition is given for the set E_{21}. Let the costs of fixed-charge on the usage of backup arc capacity be f_{ij}^ℓ, the node fortification be g_i^ℓ, and the node restoration be h_i^ℓ for some network layers $\ell \in [1, 2]$, nodes $i \in N_\ell$, and arcs $(i, j) \in A_\ell$.

2.2.1 Assumptions and Decision Dynamics

To develop a formal mathematical optimization model, we make the following assumptions regarding failure mechanisms and how they affect network functionality/operation. Of course,

Table 2.1: Definitions of indices, parameters and decision variables.

Indices	
i, j	Indexes used for the nodes
t	Time period index
ℓ	Network layer index
Sets	
N_1	Set of nodes in the first network layer
N_2	Set of nodes in the second network layer
A_1	Set of arcs in the first network layer
A_2	Set of arcs in the second network layer
E_{12}	Set of interdependent arcs where node i in layer 1 depends on the input of node j in layer 2
E_{21}	Set of interdependent arcs where node i in layer 2 depends on the input of node i in layer 1
Parameters	
T	Total number of time period for recovery planning
f_{ij}^{ℓ}	Fixed-charge cost of using backup capacity of arc (i, j) in layer ℓ
g_i^{ℓ}	Cost of fortification on node i in layer ℓ
h_i^{ℓ}	Cost of restoration on node i in layer ℓ
δ^{ℓ}	Restoration threshold of network layer ℓ
r_i^{ℓ}	Initial activation (failure) on node node i in layer ℓ
B_w^{ℓ}	Budget of fortification of network layer ℓ
B_z^{ℓ}	Budget of restoration in each time period of network layer ℓ
C_{ij}^{ℓ}	Capacity of arc (i, j) in network layer ℓ
$\overline{C}_{ij}^{\ell}$	Backup capacity of arc (i, j) in network layer ℓ
o_{ℓ}	Origin in network layer ℓ
d_{ℓ}	Destination in network layer ℓ
n^{ℓ}	Constant takes value of $\frac{1}{\lvert N_{\ell}\rvert+1}$
Decision Variables	
v_{ij}^{ℓ}	1 if backup capacity of arc (i, j) in layer ℓ is used, 0 otherwise
w_i^{ℓ}	1 if node i in layer ℓ is fortified, 0 otherwise
x_{ijt}^{ℓ}	flow amount on arc (i, j) in network layer ℓ at time t
y_{it}^{ℓ}	1 if node i in layer ℓ failed at time t, 0 otherwise
z_{it}^{ℓ}	1 if node i in layer ℓ is restored at time t, 0 otherwise

these are not the only possible assumptions that can be made for such a model; however, since we are formulating an optimization problem that takes into account node failures and consequent cascading failures dynamics, we believe that the assumptions listed below are reasonable for formulating the respective mixed-integer optimization problem while making it possible to conduct further rigorous analysis of its properties.

Assumption 1. *(A1) We consider node failures only, and also that nodes that belong to different layers have a "one-to-many" interdependence relationship; that is, if a node $j \in N_2$ (or $j \in N_1$) fails at time t, then all of its dependent nodes $i \in N_1$ (or $i \in N_2$) fail at time $t+1$. This definition is the same as the one used in [59], where it is referred to as "Type 1" interdependence.*

Assumption 2. *(A2) When node i fails, the arc capacity associated with the outgoing arcs of node i is reduced to 0.*

Assumption 3. *(A3) The original network performance (e.g., maximum flow values, without disruptions) is known to the decision-maker and is used as the restoration threshold.*

Assumption 4. *(A4) Restoration of a node requires one time period, hence, the node will be functional and flow can be restored at time period $t+1$ if restoration is made at time t.*

The decision dynamics of our proposed model are illustrated in Figure 2.1. The vertical axis represents the network flow level, while the the horizontal axis represents the time period. The total flow level drops when node failures are initiated. The backup capacity decision $\mathbf{v}$ and node fortification decision $\mathbf{w}$ can mitigate the damage caused by node failures. For the recovery planning, after node failures have occurred on the networks, the restoration decisions $\mathbf{z}$ represent a gradual repair of the nodes, thus bringing the flow level back to the designated threshold.

2.2.2 Network Performance Constraints

Next, we describe the details of the formulation. We assess network performance by defining the flow variables $x^{\ell}_{ijt} \geq 0$ between the origin and destination (O-D) path for each graph. Here we assume only one O-D path in each network and that x^{ℓ}_{ijt} represents the flow on arc

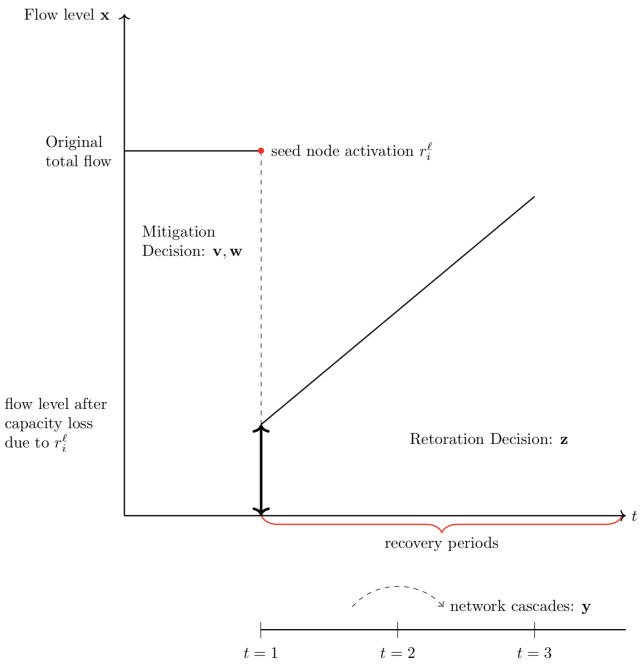


Figure 2.1: Illustration of decision dynamics in the proposed model.

(i, j) at time t of network layer ℓ. The physical flow must satisfy the flow conservation law,

$$\sum_{j:(i,j)\in A_\ell} x_{ijt}^\ell - \sum_{j:(j,i)\in A_\ell} x_{ijt}^\ell = 0 \quad \ell \in [1,2],$$
$$i \in N_\ell \setminus \{o_\ell, d_\ell\}, \quad t \in [1,T], \tag{2.1}$$

where o_ℓ and d_ℓ represented the origin and destination node of network layer ℓ. It is reasonable to assume that the decision maker has knowledge of the network's behavior (e.g., maximum flow values without node disruptions) and knows the required recovery level to restore network performance. Since we adopt maximum flow to evaluate performance, we define a non-negative parameter δ^ℓ to be the restoration threshold. The constraint on the

target performance in terms of flows can be stated as

$$\sum_{j:(i,d_\ell)\in A_\ell} x^\ell_{ijT} \geq \delta^\ell \quad \ell \in [1,2]. \tag{2.2}$$

2.2.3 Cascading Failure, Mitigation and Restoration Constraints

To model the loss and restoration of network performance, we introduce the dynamic indicator of cascading failure and recovery into the model. Let $y^\ell_{it} \in \{0,1\}$ equal 1 if node i in layer l is in a "failure" status at time step t, otherwise, let it equal 0. Also let $z^\ell_{it} \in \{0,1\}$ be 1 if the function of node i of network layer ℓ is recovered in time period t, 0 otherwise. For the purpose of mitigation, let $v^\ell_{ij} \in \{0,1\}$ equal 1 if the backup capacity of arc (i,j) in network layer ℓ is used, otherwise, let it equal 0. Let $w^\ell_i \in \{0,1\}$ equal 1 if node i in network layer ℓ is fortified and immune to failure, otherwise let it equal 0.The dynamics of failure propagation in each node can be described through the following constraints. First, we state the propagation constraints in layer 1, for $i \in N_1$, $t \in [1, T-1]$, we have

$$y^1_{i,t+1} \geq K_2(y^1_{it} - z^1_{it}) + K_2 \sum_{(j,i)\in E_{21}} (y^2_{jt} - z^2_{jt}) - w^1_i, \tag{2.3}$$

$$y^1_{i,t+1} \leq y^1_{it} - z^1_{it} + \sum_{(j,i)\in E_{21}} (y^2_{jt} - z^2_{jt}), \tag{2.4}$$

where K_2 is a scaling constant which takes the value of $\frac{1}{|N_2|+1}$. Constraints (2.3) and (2.4) repeatedly model the failure propagation for every time period. The constant K_2 ensures that the summation of the right-hand side is at most 1 when node i has failed already at time t, or if any of the nodes it depends on from layer 2 failed at time t while having no restoration or fortification. However, if a node is fortified, the failure propagation will be interdicted through all time periods. By the same logic, for $i \in N_2$, $t \in [1, T-1]$, the

propagation constraints in layer 2 can be stated as

$$y^2_{i,t+1} \geq K_1(y^2_{it} - z^2_{it}) + K_1 \sum_{(j,i)\in E_{12}} (y^1_{jt} - z^1_{jt}) - w^2_i, \tag{2.5}$$

$$y^2_{i,t+1} \leq y^2_{it} - z^2_{it} + \sum_{(j,i)\in E_{12}} (y^1_{jt} - z^1_{jt}), \tag{2.6}$$

where $K_1 = \frac{1}{|N_1|+1}$. To model node fortification in all the time periods it is effective in, we include the node immune constraint as follows:

$$y^\ell_{it} \leq 1 - w^\ell_i \quad \ell \in [1,2], \quad i \in N_\ell, \quad t \in [1,T]. \tag{2.7}$$

For each node i, the initial condition can be accounted for by adding the following constraints. Let parameter r^ℓ_i equal 1 if the "seed node" i is initially "activated" (or, failed), otherwise, let it equal 0. Seed nodes can be viewed as a subset of nodes that initiate the cascading process at time zero. The initial seed activating constraints are described by

$$y^\ell_{i1} \geq r^\ell_i(1 - w^\ell_i) \qquad \ell \in [1,2], \quad i \in N_\ell, \tag{2.8}$$

$$y^\ell_{i1} \leq r^\ell_i \qquad \ell \in [1,2], \quad i \in N_\ell. \tag{2.9}$$

2.2.4 Capacity Constraints

With proper definitions of network dynamics characterized by fortification, failure, and restoration indicators, we now characterize how the flow is affected by the interaction of mitigation, disruption, and restoration via modeling the capacity change in each arc caused by disruption. Because of (A3), the outgoing flow is blocked from node i once a node is failed. Let C^ℓ_{ij} and $\overline{C}_{ij}$ denote the capacity and backup capacity amount on each arc (i,j) of network layer ℓ. Hence, for $\ell \in [1,2]$, $(i,j) \in A_\ell$, $t \in [2,T]$, the relationship between physical

flow and capacity is described by

$$x_{ijt}^{\ell} \leq \left(C_{ij}^{\ell} + \overline{C}_{ij}^{\ell} v_{ij}^{\ell}\right)\left(1 - y_{i,t-1}^{\ell} + z_{i,t-1}^{\ell}\right). \tag{2.10}$$

Once a node has failed, the capacity is lost until recovery is made. Note that due to (A4), capacity is restored in the period $t+1$ only if recovery is done in t. The nonlinearity in constraints (2.8) can be linearized via McCormick linearization [38] with auxiliary variables $\phi_{ijt}^{\ell} \geq 0$, for $\ell \in [1,2]$, $(i,j) \in A_{\ell}$, $t \in [2,T]$, we have

$$x_{ijt}^{\ell} \leq \phi_{ijt}^{\ell}, \tag{2.11a}$$

$$\phi_{ijt}^{\ell} \leq \left(C_{ij}^{\ell} + \overline{C}_{ij}^{\ell} v_{ij}^{\ell}\right), \tag{2.11b}$$

$$\phi_{ijt}^{\ell} \leq \left(C_{ij}^{\ell} + \overline{C}_{ij}^{\ell}\right)\left(1 - y_{i,t-1}^{\ell} + z_{i,t-1}^{\ell}\right), \tag{2.11c}$$

$$\phi_{ijt}^{\ell} \geq \left(C_{ij}^{\ell} + \overline{C}_{ij}^{\ell} v_{ij}^{\ell}\right) - \left(C_{ij}^{\ell} + \overline{C}_{ij}^{\ell}\right)\left(y_{i,t-1}^{\ell} - z_{i,t-1}^{\ell}\right). \tag{2.11d}$$

2.2.5 Logical Conditions and Budget Constraints

In addition, we include several logical conditions between failure and recovery as well as the budget limitation to execute node fortification and recovery in the following constraints:

$$\sum_{i \in N_{\ell}} w_i^{\ell} \leq B_w^{\ell} \qquad \ell \in [1,2], \tag{2.12}$$

$$\sum_{i \in N_{\ell}} z_{it}^{\ell} \leq B_z^{\ell} \qquad \ell \in [1,2], \quad t \in [1,T], \tag{2.13}$$

$$z_{it}^{\ell} \leq y_{it}^{\ell} \qquad \ell \in [1,2], \quad i \in N_{\ell} \setminus \{o_{\ell}, d_{\ell}\}, \quad t \in [1,T], \tag{2.14}$$

$$z_{it}^{\ell} = y_{it}^{\ell} \qquad \ell \in [1,2], \quad i \in \{o_{\ell}, d_{\ell}\}, \quad t \in [1,T], \tag{2.15}$$

$$y_{i,t+1}^{\ell} + z_{it}^{\ell} \leq 1 \qquad \ell \in [1,2], \quad i \in N_{\ell}, \quad t \in [1,T-1]. \tag{2.16}$$

The budget limitation of fortification and restoration is described in constraints (2.12) and (2.13) where B_w^ℓ and B_z^ℓ are the numbers of nodes can be fortified and recovered in each time period, respectively. Constraints (2.14) represent that the recovery of a node is possible only if it has failed already. Because the origin and destination are the required nodes to create a positive flow in the maximum flow settings, we ensure their recovery in constraints (2.15) if any of them have failed. Finally, constraints (2.16) ensure that failure will not be propagated to time step $t+1$ once a node is restored at time step t. With the objective of minimizing the total cost of capacity expansion, while increasing node fortification and recovery, the mixed-integer programming formulation for the considered problem of mitigation and restoration on two-layered interdependent networks (**2-INMR**) is given by

$$\textbf{2-INMR:} \quad \min \Big\{ \mathbf{f}^\intercal \mathbf{v} + \mathbf{g}^\intercal \mathbf{w} + \mathbf{h}^\intercal \mathbf{z} : (2.1) - (2.9), \quad (2.11a) - (2.11d), \\ (2.12) - (2.16), \quad \mathbf{x} \geq 0, \quad \mathbf{v}, \mathbf{w}, \mathbf{y}, \mathbf{z} \in \{0,1\} \Big\}.$$

2.2.6 An Example of Mitigation and Restoration Against Cascading Failures

To illustrate the considered cascading failure mechanisms, a small-scale example of **2-INMR** is given in Figures 2.2 - 2.5, in which two identical networks G_1 and G_2, where $N_1 = N_2 = \{1,2,3,4,5\}$ and $A_1 = A_2 = \{(1,2),(1,3),(2,3),(2,4),(2,5),(3,2),(3,5),(4,5)\}$, are shown, along with the interdependent links set $E_{12} = \{(5,1),(5,2),(5,3)\}$ and $E_{21} = \{(4,1),(5,5)\}$. The origin and destination in each network are 1 and 5. In Figure 2.2, the black arcs represent the optimal positive flow amount under normal node operations.

Suppose in Figure 2.3 the initial failed nodes are $\{2,4,5\} \in G_1$ and $\{2\} \in G_2$ at $t = 1$. Immediately as a result of these node failures, arcs $(2,4)$ and $(4,5)$ in layer 1 and arcs $(2,3),(2,4)$ and $(4,5)$ in layer 2 would have zero capacity, which blocks the flow and reduces

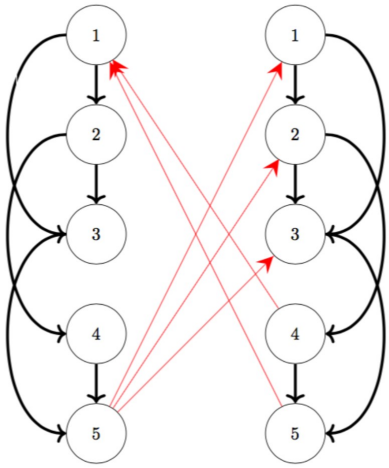


Figure 2.2: Optimal flow distribution on a two-layer interdependent network.

the flow level. Without mitigation and restoration, the consequence is shown in Figure 2.4, where failed nodes block the flow at the next time step $t+1$ and result in partial flow in G_1 and zero flow in G_2 due to the failure propagation from node 5 in layer 1 to node 1, 2, and 3 in layer 2 in one time period.

To demonstrate potential benefits of using our proposed strategies with mitigation and restoration, as depicted in Figure 2.5, we consider a small test case with four time periods and let $B_w^1 = B_w^2 = 1$ and $z_{it}^1 = z_{it}^2 = 1$. We let $C_{ij}^\ell = \overline{C}_{ij}^\ell = f_{ij}^\ell$ and randomly generate the arc capacity parameters. The parameters associated with arcs $\{(1,2),(1,3),(2,3),(2,4),(2,5),(3,2),(3,5),(4,5)\}$ are $\{143, 104, 82, 137, 109, 132, 131, 145\}$, respectively. We also randomly generate fortification and restoration costs and assume that restoration cost is approximately 25% higher than fortification cost. The fortification costs associated with each node are $\{139, 132, 124, 143, 174\}$ and the restoration costs are $\{173, 165, 155, 178, 217\}$. The original maximum flow in both graphs are equal and thus the restoration threshold is $\delta^1 = \delta_2 = 247$.

In Figure 2.5, network layer G_1 is able to restore the flow by fortifying node 5 (yellow

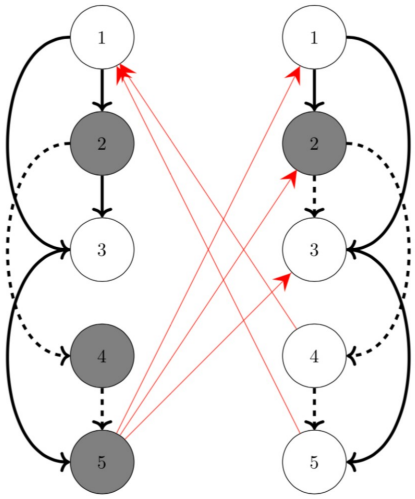


Figure 2.3: Disruptions of flows caused by failures of the "seed nodes" (in grey) at the intitial time step.

node), recovering node 2, and utilizing the backup capacity of arc $(2, 5)$ to mitigate the damage caused by node 4. In network layer G_2, simply fortifying node 2 is sufficient. The total cost is 580 units. We compare optimal costs based on different combinations of decisions; these decisions are whether to allow restoration only (R), restoration and backup capacity (R+B), or restoration, backup capacity, and fortification (R+B+F). Table 2.2 shows the cost comparison with all four scenarios and the node recovery sequence for each scenario. The costs of using restoration only is 20% higher than using restoration, backup capacity, and fortification. In the subsequent Table 2.3, we also report the optimal decisions corresponding to each scenario. The solution suggests that using the backup capacity of arc $(2, 5)$ is critical to mitigate damage on node 4 in layer 1.

Remark 1. *The upper bound of the cost can be estimated by multiplying the number of*

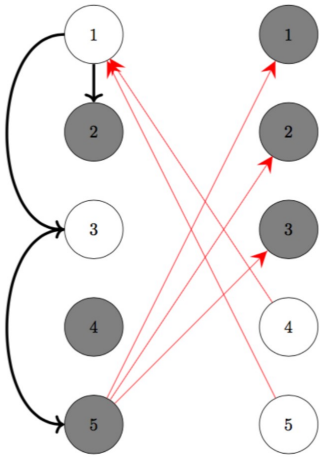


Figure 2.4: Flow distribution after initial node failures (note that additional two nodes fail as a result of their dependence on a node that failed at the initial time step).

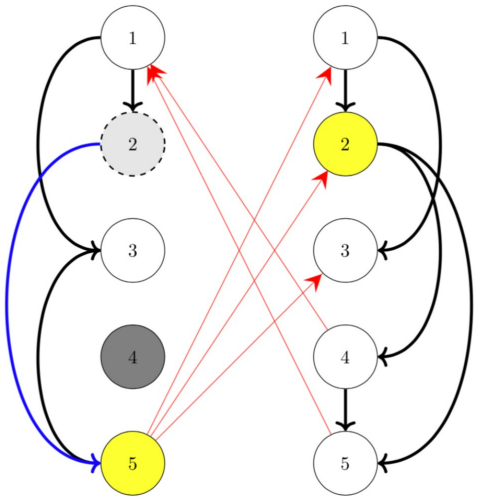


Figure 2.5: Mitigation and restoration against cascading failures.

initial failed nodes and restoration costs, that is,

$$\sum_{\ell \in [1,2]} \sum_{i \in N_\ell} r_i^\ell h_i^\ell.$$

Table 2.2: Cost comparison based on different scenarios.

Scenario	Node Recovery Sequence				Cost
	$t = 1$	$t = 2$	$t = 3$	$t = 4$	
R	$N_1(5), N_2(2)$	$N_1(2)$	$N_1(4)$	—	725
R + B	$N_1(5), N_2(2)$	—	$N_1(2)$	—	656
R + F	$N_1(4)$	—	$N_1(2)$	—	649
R + B + F	—	$N_1(2)$	—	—	580

Table 2.3: Optimal decision in different scenarios.

Scenario	Failed Node	Backup Arc Used	Node Fortified
R	–	–	–
R+B	$N_1(4)$	$(2, 5) \in A_1$	–
R+F	–	–	$N_1(5), N_2(2)$
R+B+F	$N_1(4)$	$(2, 5) \in A_1$	$N_1(5), N_2(2)$

This is the worst-case cost when neither fortification nor backup arc capacity are available to mitigate the losses caused by cascading failures.

2.2.7 Possible Extensions of the Model

We close this section by discussing several possible extensions of the proposed model in order to illustrate its flexibility in other related settings.

First, we show how to develop our model for *multi-layer interdependent networks*. Consider m directed graphs $G_\ell = (N_\ell, A_\ell)$ where $\ell \in [1, m]$. For every pair of interdependent link $(i, j) \in E_{pq}$, it now represents the operations of node i in N_p depends upon the input of node j in N_q, where $p \in [1, m]$, $q \in [1, m]$ and $p \neq q$. Clearly, the number of sets E_{pq} is bounded by 2-permutations of m, if all networks are interdependent with each other. Since the constraints in section 2.2.2, 2.2.3, 2.2.4, and 2.2.5 are indexed by the layer index ℓ, these

constraints can be extended for multi-layer networks naturally. For example, the multi-layer version of cascading constraints (2.3) and (2.4) in section 2.2.3 between layers p and q can be formulated as

$$y^q_{i,t+1} \geq K_p(y^q_{it} - z^q_{it}) + K_p \sum_{(j,i)\in E_{pq}} (y^p_{jt} - z^p_{jt}) - w^q_i, \tag{2.17}$$

$$y^q_{i,t+1} \leq y^q_{it} - z^q_{it} + \sum_{(j,i)\in E_{pq}} (y^p_{jt} - z^p_{jt}). \tag{2.18}$$

Next, to model other types of network operations (flows) in each layer, one can simply modify the constraints in section 2.2.2. For instance, suppose that for layer 1 we consider a multi-commodity flow problem with K commodities and their time-dependent demand D_{kt} at time period t. Then for $k \in [1, K]$ and $t \in [1, T]$, the network operations constraints in layer 1 can be formulated as

$$\sum_{j:(i,j)\in A_1} x^1_{ijkt} - \sum_{j:(j,i)\in A_1} x^1_{ijkt} = \begin{cases} 0 & i \in N_1 \setminus \{o_1, d_1\}, \\ D_{kt} & i = o_1, \\ -D_{kt} & i = d_1. \end{cases} \tag{2.19}$$

Note that the capacity level on each arc is still subject to node failures as described in Section 2.2.3. The evaluation of network performance for this layer is to see how much cost is required to allocate on fortification, backup capacity and restoration in order to satisfy the demand for each commodity.

Finally, we consider the situation where the arc capacities are not necessarily 100% lost when nodes fail. Given a predetermined loss rate parameter γ with $0 \leq \gamma \leq 1$, the

partial loss capacity constraints due to node fail can be formulated as

$$x^{\ell}_{ijt} \leq \left(C^{\ell}_{ij} + \overline{C}^{\ell}_{ij} v^{\ell}_{ij}\right) \left[1 - \gamma \left(y^{\ell}_{i,t-1} - z^{\ell}_{i,t-1}\right)\right]. \tag{2.20}$$

If node i is failed at time $t-1$ ($y^{\ell}_{i,t-1} = 1$) and there is no backup capacity ($v^{\ell}_{ij} = 0$) or restoration ($z^{\ell}_{i,t-1} = 0$), then the upper bound of flow is reduced to $(1-\gamma)C^{\ell}_{ij}$. The predetermined loss rate can also be tailored to each node depending on the type of network components.

2.3 Valid Inequalities

Mixed-integer optimization problems are generally known to be computationally challenging. In addition, a proper mathematical formulation is critical for the computational performance. Polyhedral methods such as deriving valid inequalities or strong formulations are often used to improve the computational time and solution quality. Therefore, in this section, we provide valid inequalities for the **2-INMR** problem formulation defined above.

2.3.1 Strengthened Cascades Inequalities

The big-M type constants K_1 and K_2 in constraints (2.3) - (2.6) create many fractional points; therefore, we attempt to remove them by introducing valid inequalities and thus tighten the constraints. We let $\mathscr{P}$ be the set of feasible solutions to the polytope consisting of (2.3) - (2.7). Let $\Gamma^1_i := \{j \mid (j,i) \in E_{21}\} \cup \{i\}$ for $i \in N_1$. In other words, Γ^1_i is the set which contains all the nodes $j \in N_2$ of which $i \in N_1$ depends on through interdependent arc set E_{21} and itself. A similar definition is made for Γ^2_i as well.

Proposition 1. *For $i \in N_1$, $\quad t \in [1, T-1]$, $\quad k \in \Gamma^1_i$, the strengthened cascades inequality*

in layer 1

$$y^1_{i,t+1} + w^1_i \geq y^\ell_{kt} - z^\ell_{kt} \tag{2.21}$$

is valid for $\mathscr{P}$*, where* $\ell = 1$ *if* $k = i$*,* $\ell = 2$ *if* $k \neq i$*.*

Proof. For $k = i$, the inequalities reduce to $y^1_{i,t+1} + w^1_i \geq y^1_{it} - z^1_{it}$ which is valid. For $k \neq i$, the inequalities reduce to the disaggregate versions of inequalities (2.3), saying any node in $j \in N_2$ which fails will cause $i \in N_1$ to fail, which is exactly the definition of Type 1 interdependence mentioned in Assumption 1 above. □

Corollary 1. *For* $i \in N_2, \quad t \in [1, T-1], \quad k \in \Gamma^2_i$*, the cascade inequality in layer 2*

$$y^2_{i,t+1} + w^2_i \geq y^\ell_{kt} - z^\ell_{kt} \tag{2.22}$$

is valid for $\mathscr{P}$*, where* $\ell = 2$ *if* $k = i$*,* $\ell = 1$ *if* $k \neq i$*.*

Example 1. *Consider two node sets* $N_1 = \{1, 2, 3\}$ *and* $N_2 = \{4, 5\}$ *and let* $E_{21} = \{(4, 2), (5, 1), (5, 2)\}$*. The strengthened cascade inequalities in layer 1 are*

$$y^1_{1,t+1} + w^1_1 \geq y^1_{1t} - z^1_{1t}, \tag{2.23}$$

$$y^1_{1,t+1} + w^1_1 \geq y^2_{5t} - z^2_{5t}, \tag{2.24}$$

$$y^1_{2,t+1} + w^1_2 \geq y^1_{2t} - z^1_{2t}, \tag{2.25}$$

$$y^1_{2,t+1} + w^1_2 \geq y^2_{4t} - z^2_{4t}, \tag{2.26}$$

$$y^1_{2,t+1} + w^1_2 \geq y^2_{5t} - z^2_{5t}. \tag{2.27}$$

2.3.2 Strengthened Capacity Inequalities

Observe that (2.11b) - (2.11d) consist of a polytope. We are interested in finding a tighter description of these constraints.

Proposition 2. *Let $\mathscr{Q}_1$ denote the feasible solution set to constraints* (2.11b) - (2.11d) *and define feasible set $\mathscr{Q}_2$ by the following constraints:*

$$\phi^\ell_{ijt} \leq C^\ell_{ij} + \overline{C}^\ell_{ij} - \left(C^\ell_{ij} + \overline{C}^\ell_{ij}\right)\left(y^\ell_{i,t-1} - z^\ell_{i,t-1}\right), \tag{2.28a}$$

$$\phi^\ell_{ijt} \leq C^\ell_{ij} + \overline{C}^\ell_{ij} v^\ell_{ij} - C^\ell_{ij}\left(y^\ell_{i,t-1} - z^\ell_{i,t-1}\right), \tag{2.28b}$$

$$\phi^\ell_{ijt} \geq C^\ell_{ij} + \overline{C}^\ell_{ij} v^\ell_{ij} - \left(C^\ell_{ij} + \overline{C}^\ell_{ij}\right)\left(y^\ell_{i,t-1} - z^\ell_{i,t-1}\right), \tag{2.28c}$$

$$\phi^\ell_{ijt} \geq C^\ell_{ij}\left(1 - y^\ell_{i,t-1} + z^\ell_{i,t-1}\right), \tag{2.28d}$$

then $\mathscr{Q}_2 \subseteq \mathscr{Q}_1$.

Proof. We characterize all the extreme points of a polytope denoted by $\mathscr{Q}_2$ in Table 2.4 by dropping all the indexes. Observe that $\mathscr{Q}_2$ is a polytope of one network component at a single time period. It is not difficult to verify that these extreme points are $\in \mathscr{Q}_1$ as well. Note that as long as C and $\overline{C}$ are integers, constraints (2.28a) - (2.28d) will give integer extreme points. Now consider fractional points $(\hat{\phi}, \hat{v}, \hat{y}, \hat{z}) = (0, 0, \frac{C}{C+\overline{C}}, 0), (C, 0, \frac{\overline{C}}{C+\overline{C}}, 0), (0, 0, 1, \frac{\overline{C}}{C+\overline{C}})$ and $(C, 0, 1, \frac{C}{C+\overline{C}})$, these points belong to $\mathscr{Q}_1$ but they are cut off by constraints (2.28a) - (2.28d). □

2.4 Computations

In this section, we report our computational experiments on the interdependent networks mitigation and restoration problem under different settings.

Table 2.4: Extreme points of a single polytope of $\mathscr{Q}_2$.

ϕ	v	y	z
0	0	1	0
0	1	1	0
C	0	0	0
C	0	1	1
$C+\overline{C}$	1	0	0
$C+\overline{C}$	1	1	1

2.4.1 Test Instances Generation and Software/Hardware Description

For illustrative purposes, we consider two networks where $|N_1| = |N_2|$ and $|A_1| = |A_2|$, with the same network topology. In each combination of node and arc sets, the interdependent arc sets $|E_{12}|$ and $|E_{21}|$ are randomly generated according to the targeted density 10%, 20%, 30%, and 40%. We use the `NetworkX` package of Python 2.7 to generate a Erdos-Renyi graph with the connecting probabilities between each node to be 0.1. We generate 5 instances with 10, 20, 30, 40, and 50 nodes in each layer, denoted by 10-10, 20-20, ..., 50-50, respectively.

In each network layer, there exists exactly one origin-destination path, and the maximum flow δ^1 and δ^2 (without disruptions/failures) are assumed to be known already. Arc capacities C_{ij}^{ℓ} are generated based on the discrete uniform distribution in interval $[80, 150]$, and we let the backup capacity amount $\overline{C}_{ij}^{\ell}$ and fixed-charge cost f_{ij}^{ℓ} equal the generated capacity. For the fortification cost and restoration cost, we assume the latter is 25% more expensive; the costs are generated based on a discrete uniform distribution in the interval $[120, 180]$.

In order to ensure that a feasible solution exists, we let the planning horizon T equal the number of nodes. The fortification and restoration budgets are assumed to be 40% of the nodes in each layer. The initial seed nodes failures (or, "activation") is generated based on the discrete uniform distribution in the interval $[0, 1]$.

All experiments are conducted by using a single thread on a PC Intel Core i7 processor with a CPU at 3.40 GHz and a RAM space of 8GB. The problem is coded in Python 2.7 and Gurobi 8.1 is used as the mixed-integer optimization solver.

2.4.2 Benefits of Proposed Mitigation and Restoration Strategy

We first report the cost savings of different scenarios based on the instances generated using the above assumptions. Similar to the small illustrative example in Section 2.2.6, we compute the exact optimal cost of the number of nodes on each side when equaled to 10, 20 and 30 respectively, with an interdependent arc density range from 10% to 40%. The results are shown is Table 2.5. In each category, the last row of Table 2.5 indicates the cost savings by calculating the maximum cost minus the minimum cost divided by maximum cost of that column, multiplied by 100%. We further depict the optimal cost of four different scenarios as the function of interdependent arc density for the instances with $N_1 = N_2 = 10$, $N_1 = N_2 = 20$ and $N_1 = N_2 = 30$ in Figure 2.6, 2.7 and 2.8, respectively.

Clearly, the cost increases as the density of interdependent arcs increases, and using restoration only produces the highest costs compared to other scenarios. The minimum costs for all tested data from using restoration, backup capacity, and node fortification together shows that using all three strategies produces compelling results in terms of cost saving. Note that depending on the network topology, it is possible that backup arc capacity may be unable to mitigate the damage since the node must survive for it to be used. When fortification is incorporated, it is possible to circumvent failed nodes by launching backup capacity from surviving nodes, thus reducing the recovery cost.

2.4.3 Solution Quality at Root Node

Next, we test the strength of the "natural" formulation given in Section 2.2 (denoted by NF) and the one with valid inequalities derived in Section 2.3. We add all of the cascade

Table 2.5: Cost comparison for generated instances.

		Cost with Different $\|E\|$ Density			
Nodes	Scenario	10%	20%	30%	40%
10-10	R	1814	2374	2354	2836
	R+B	1484	2374	2354	2836
	R+F	1225	1520	1520	1705
	R+B+F	1221	1405	1520	1705
	Saving	32.69%	40.82%	35.43%	39.88%
20-20	R	2901	3097	3097	3097
	R+B	2884	3097	3097	3097
	R+F	2148	2482	2482	2482
	R+B+F	2082	2482	2482	2482
	Saving	28.23%	19.86%	19.86%	19.86%
30-30	R	6062	8520	10652	10652
	R+B	6062	8520	10652	10652
	R+F	2036	2444	2482	2482
	R+B+F	1829	2292	2292	2292
	Saving	69.83%	73.10%	78.48%	78.48%

inequalities into the natural formulation as the number of these inequalities is polynomial, then we replace constraints (2.11b) - (2.11d) by (2.28a) - (2.28d) and denote this formulation as SF.

In Table 2.6, we compare the LP relaxation gap at the root node for both formulations. We do not limit the computational time, since the goal is to get the true optimal solution so that the root gap can be calculated. The root gap is calculated by $\frac{\texttt{ZUB}-\texttt{ZLB}}{\texttt{ZUB}} \times 100\%$ where ZUB is the optimal solution and ZLB is the LP relaxation on root node. From the last column we can observe that the improvement on LP relaxation is very significant, which means the valid inequalities are very strong. The improvement for all the instances are above 54% with the

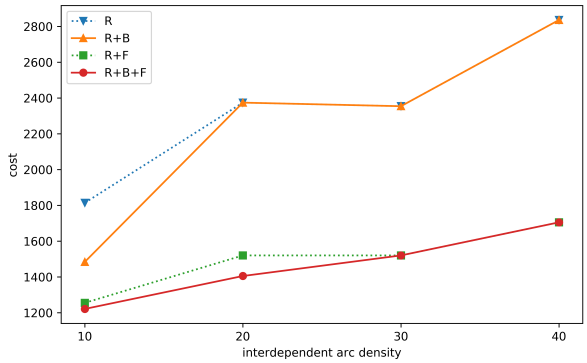


Figure 2.6: Optimal cost comparison when $N_1 = N_2 = 10$.

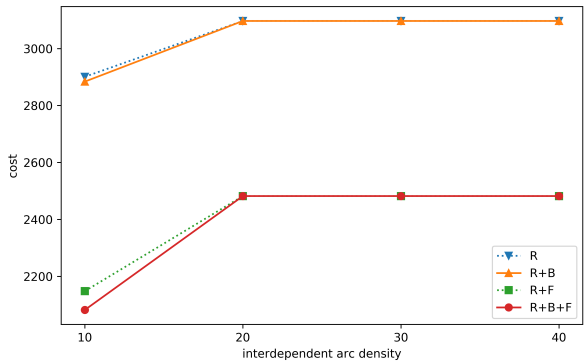


Figure 2.7: Optimal cost comparison when $N_1 = N_2 = 20$.

maximum gap improvement 86.36%. Reducing the root relaxation gap improves the lower bound and thus gives a computational advantage for solving the optimization problem.

2.4.4 Benefits of Strengthened Formulation

Finally, we would like to demonstrate the effectiveness of the proposed inequalities when it comes to optimization. The computational time is limited to 3600 seconds. If a problem is not solved to optimality when the time limit is reached, we report the ending gap, which is

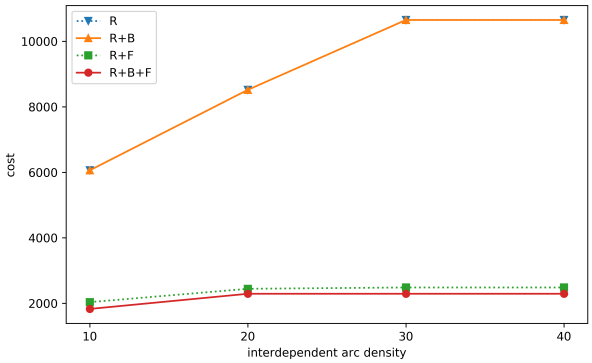


Figure 2.8: Optimal cost comparison when $N_1 = N_2 = 30$.

calculated by $\frac{\texttt{ZUB}-\texttt{ZLB}}{\texttt{ZUB}} \times 100\%$. Note that ZUB is the incumbent integer solution while ZLB is the best lower bound generated by Gurobi. The notation "–" indicates that Gurobi is not able to find any integer solution within an hour but a lower bound is generated. We report the total branch-and-bound nodes explored, the number of Gurobi cuts generated, and the CPU seconds in subsequent columns these are summarized in Table 2.7.

For the first three data categories, the effectiveness of the valid inequalities is superior. Gurobi cannot close the gap when the number of nodes is 20 and the density of interdependent arcs are above 10% for NF, while with the valid inequalities all the instances in the first three categories can be solved in less than 161 seconds. However, when the number of nodes increases to 40 and 50, the problem becomes very challenging to solve. When the number of nodes equals 40, the ending gap of SF within an hour is less than 6%. Moreover, Gurobi is unable to find an integer solution for an instance with 20% and 40% interdependent arcs density while the ending gap is above 60% for interdependent arcs density 10% and 30%. When the number of nodes equals 50, Gurobi is again unable to find an integer incumbent solution for NF with 10%, 20%, 30% interdependent arcs density respectively. The only exception when SF does not outperform NF within an hour is the instance with 50 nodes and

Table 2.6: Comparison on Root Relaxation.

					Root Gap (%)		
Nodes	Arcs	Density	$\|E_{12}\|$	$\|E_{21}\|$	NF	SF	Improve
10-10	74	10%	8	10	44.42	9.03	79.67
		20%	22	23	63.38	8.65	86.36
		30%	25	25	65.62	14.41	78.04
		40%	42	41	78.02	21.38	72.60
20-20	36	10%	37	41	80.71	33.31	58.73
		20%	71	87	82.94	31.91	61.53
		30%	123	114	83.54	33.09	60.39
		40%	164	148	83.73	35.98	57.03
30-30	103	10%	99	80	82.01	23.90	70.76
		20%	179	196	85.93	27.53	67.96
		30%	259	265	85.62	28.88	66.27
		40%	359	353	84.71	27.21	67.89
40-40	184	10%	143	171	90.04	35.58	60.49
		20%	340	309	90.13	37.86	58.00
		30%	483	496	90.02	41.14	54.3
		40%	622	654	90.21	41.27	54.25
50-50	247	10%	247	242	88.98	30.57	65.64
		20%	517	498	88.33	31.96	63.82
		30%	760	742	87.98	32.73	62.80
		40%	1021	990	87.73	32.72	62.71

40% density interdependent arcs, where we denote * in the column. From our experiment, it took Gurobi 4147.11 seconds to solve the root relaxation for SF, however, the root relaxation gap is significantly smaller than NF (33% versus 88%), which results in an overall shorter computational time.

Table 2.7: Effectiveness of the Formulations.

					B&B Nodes		Gurobi Cuts		CPU Time (EGap)	
Nodes	Arcs	Density	$\|E_{12}\|$	$\|E_{21}\|$	NF	SF	NF	SF	NF	SF
10-10	74	10%	8	10	115	1	698	3	1.05	0.35
		20%	22	23	195	1	1843	0	3.44	0.37
		30%	25	25	45	1	1438	43	3.36	0.87
		40%	42	41	6294	1	2108	24	124.63	0.41
20-20	36	10%	37	41	15812	868	2460	243	934.35	9.73
		20%	71	87	9587	2309	4037	199	(11.97%)	29.11
		30%	123	114	7585	828	4387	149	(26.47%)	22.96
		40%	164	148	6833	1675	4884	178	(27.6%)	47.30
30-30	103	10%	99	80	4334	315	8415	69	(15.47%)	39.67
		20%	179	196	1652	1463	8489	272	(34.83%)	160.81
		30%	259	265	459	261	8815	211	(54.60%)	104.09
		40%	359	353	3501	303	8222	178	(41.43%)	131.08
40-40	184	10%	143	171	129	17778	15684	589	(59.38%)	(1.72%)
		20%	340	309	453	10202	15967	579	—	(2.56%)
		30%	483	496	312	10202	18078	307	(84.53%)	(2.85%)
		40%	622	654	774	3957	16812	318	—	(5.18%)
50-50	247	10%	247	242	1	2598	28586	318	—	(5.53%)
		20%	517	498	1	1	27692	362	—	(13.98%)
		30%	760	742	1	526	27164	288	—	(8.72%)
		40%	1021	990	754	0	26829	0	(91.10%)	∗

2.5 Conclusion

In this chapter, we propose a novel optimization model for mitigation and restoration against a dynamic cascade of node failures over two-layer interdependent networks. We consider the maximum flow problem as it is often used in various settings of studying information, infrastructure, and other types of networked systems. From the computational results on the

considered illustrative examples, we demonstrate that significant cost savings can be achieved by considering mitigation and restoration simultaneously to prevent cascading failure and restore network performance in a designated time horizon. Moreover, in the presence of dynamic cascading failures, our model is able to prioritize the recovery of the damaged components to avert further failure cascade.

We present two classes of valid inequalities derived from the cascading constraints and capacity constraints, both of which are substructures of the original formulation. We incorporate these valid inequalities into the formulation directly in order to tighten the bound. Our computational experiments show that these inequalities are enough to solve instances from $N = 10$ up to $N = 30$ on each side of the interdependent network with an interdependence density range from 10% to 40% within a short amount of time, while default Gurobi settings are unable to solve most of the instances when valid inequalities are not added to the formulation. To identify some potential model extensions, we discussed the possibility of extending our model to several different settings, including multi-layer interdependent networks, multi-commodity flows, and capacity degradation, by changing certain parts of the formulation of **2-INMR**.

CHAPTER 3: A POLYHEDRAL STUDY OF LEAST COST INFLUENCE MAXIMIZATION IN SOCIAL NETWORKS

This chapter is partially based on [11]. The least cost influence maximization problem aims to determine minimum cost of partial (e.g., monetary) incentives initially given to the influential spreaders on a social network, so that these early adopters exert influence toward their neighbors and prompt influence propagation to reach a desired penetration rate by the end of cascading processes. We first conduct polyhedral analysis on a substructure that describes influence propagation assuming influence weights are unequal, linear and additively separable. Two classes of facet-defining inequalities based on a mixed 0-1 knapsack set contained in this substructure are proposed. We characterize another exponential class of valid and facet-defining inequalities utilizing the concept of minimum influencing subset. We show that these inequalities can be separated in polynomial time efficiently. Furthermore, a polynomial-time dynamic programming recursion is presented to solve this problem on a simple cycle graph. For arbitrary graphs, we propose a new exponential class of valid inequalities that dominates the cycle elimination constraints and an efficient separation algorithm for them. A compact convex hull description for a special case is presented. We illustrate the effectiveness of these inequalities via a delayed cut generation algorithm in the computational experiments.

3.1 Introduction

Intricate connections between entities in many natural and man-made systems form large complex networks. Of particular interest in network science is to gain insight into the dynamic of cascading processes in complex networks. For instance, the spreading of new behaviors, opinions, technologies, conventions, and gossips from person to person through a social network may play an important role in designing competitive marketing strategies.

Indeed, the collection of social ties among consumers can be exploited to select pivotal early adopters for initiation and anticipate the time and cost required for the information propagation. Therefore, a key research question has been focused on how to efficiently identity a set of users to disseminate a certain information within the network, result in an increasing trend in studying social influence and information propagation in social networks (see, e.g., [14, 31]). As it may be expected, the spreading of social influence is commonly studied on network graphs that consist of nodes and links representing users and their connections, as well as a framework that describes how the information propagates among various intermediate users over time. Granovetter [23] first presented the threshold model to simulate the collective force exerted by a group on each of its members for predicting innovation and rumor diffusion, voting trends, and migration. A distinct threshold value is assigned to each user representing the proportion of neighbors who make a decision before a particular user makes such decision. Many extensions built on the threshold model have been proposed to encompass different circumstances. Among them, the linear threshold model and independent cascade model are the two most popular and well-studied one. Kempe et al. [30] study the linear threshold and independent cascade models under the influence maximization (IM) problem. The goal of IM is to activate as many nodes as possible by the end of propagation process given selected influential nodes within budget. Specifically, they show that the expected influence spread is a monotone submodular function, which can be approximated by greedy algorithm with performance guarantee $1 - \frac{1}{e} - \epsilon$. On the other hand, the target set selection (TSS) introduced by [13] aims to find the minimum number of users required initially and activate the entire network through the propagation process. These two types of problems are fundamental in social network analysis and have many variants. Here we refer the reader to a survey of hardness and solution approaches for TSS [6], a review on approximation algorithms for IM [34], and a comprehensive review on introduction to types of social networks, properties, evaluation metrics and known method to solve IM [49]. Recently,

Azaouzi et al. [5] presented a comprehensive review on models and methods of group-level IM and IM under privacy protection.

In this chapter, we study an optimization problem arising in social network analytics, referred to as the *least cost influence maximization (LCIM)* problem [18, 25, 26]. The propagation process in LCIM is based on the linear threshold model, where each user is assigned a real-valued threshold and each link between users is assigned a weight to represent influence level. If the summation of influence weights from neighbors exceeds one's threshold value, then the individual is *activated*, meaning that the information is adopted or the person changes their opinion to align with friends (neighbor nodes) in the social network. In LCIM, the concept of influence weights is extended with the idea of giving partial incentives such as free samples or coupons to individual as a motivation for spreading information. The assumption of the linear threshold model allows the incentives and influence weights in the activation processes to be linearly additive, which immediately admits a mixed-integer optimization formulation for the influence propagation. Our goal is to develop an exact computational method based on the polyhedral structure. We assume that all the parameters are deterministic, all the nodes are inactive initially and nodes remain activated once the influence weights exceeds the threshold. A similar assumption on using deterministic parameters in linear threshold model can be found in [27]. From a practical point of view, the assumption of deterministic linear threshold depends on the accuracy of estimation of influence factors and threshold parameters. Machine learning and data mining techniques may enable one to obtain accurate predictions on those parameters from massive amounts of data available nowadays. Note that when incentive is allowed and the influence weights and thresholds are considered known in the problem, the influence propagation is no longer submodular, thus, a greedy algorithm is not applicable for arbitrary graphs. However, the problem is still challenging to solve due to the combinatorial nature. The minimum subset of nodes required under dynamic activation is also referred to as the dynamic monopoly, where

nodes become activated if at least half of the neighbors are activated in the previous time period. Hence, an extra index of time is incorporated into the integer programming formulation to describe step-by-step activation of nodes, see [40, 41, 55]. We do not consider time dynamics in our problem despite the fact that the influence propagation occurs in discrete steps. Moreover, the induced optimal influence propagation graph for LCIM in the static network setting given by the mixed-integer programming formulation is known to be acyclic [18].

Even though IM, TSS and LCIM problems share certain similarities, most of the previous studies on IM or TSS mainly focus on developing degree-based or centrality-based metaheuristics and approximation algorithms. Existing studies on using exact mixed-integer optimization techniques for LCIM under deterministic data settings are relatively limited. The computational complexity of LCIM is established in [26]. In particular, they show that LCIM is NP-hard on arbitrary graphs and bipartite graphs for both equal and unequal influence when 100% penetration rate is not required. They also give a greedy algorithm and a total unimodular formulation for LCIM with equal influence on a tree and 100% penetration rate. Günneç et al. [25] develop a branch-and-cut algorithm using this total unimodular formulation for LCIM on arbitrary graphs. Fischetti et al. [18] present a novel set covering formulation for a generalized LCIM, where the activation function can be adjusted to be nonlinear in order to capture the situation of diminishing marginal influences or over-proportional effect from peers. They propose strengthened generalized propagation inequalities and a price-cut-and-branch algorithm to deal with the exponential number of variables and constraints. Recent developments on using exact mixed-integer optimization methods for stochastic influence maximization problem include a delayed constraint generation algorithm for a two-stage stochastic influence maximization problem [60] and a branch-cut-and-price algorithms for robust a influence maximization, where node thresholds and arc influence factors are subject to budget uncertainty [43].

3.1.1 Notation and problem definition

For convenience, we use the notation $[a, b]$ to denote the set $\{a, a+1, \ldots, b\}$ for $a \leq b$, and $[a, b] = \varnothing$ for $a > b$. For a set $\mathcal{R} \subseteq \mathbb{R}^n$, we use $\text{conv}\,(\mathcal{R})$ to denote its convex hull of solutions. Formally, an oriented network (e.g., a social network) is represented by a graph $G = (V, E)$ with the set of nodes $V := \{1, \ldots, n\}$ corresponding to people or users and the set of bidirectional arcs $E \subseteq \{(i, j) \in V \times V : (j, i) \in V \times V, i \neq j\}$ with cardinality m corresponding to connections and influence directions between people in the network. Hence, each node has identical number of predecessors and successors, denoted by v_i: $v_i = |N(i)|$, where $N(i) := \{j \in V : (j, i) \in E\}$. Each node $i \in V$ has a threshold value h_i measuring the "difficulty level" of an individual to be "activated". Each arc $(i, j) \in E$ is associated with an influence weight d_{ij}. The coverage (penetration) rate is denoted by a, where $0 < a \leq 1$ is assumed. We also assume that d_{ij} and h_i are positive integers such that $\max\{d_{ji} : j \in N(i)\} \leq h_i$ and $\sum_{j \in N(i)} d_{ji} > h_i$ for all $i \in V$ such that $|N(i)| \geq 2$ throughout this chapter to omit trivial cases. For any $d_{ji} > h_i$ and $|N(i)| = 1$ for some $i \in V$, we pre-process the data to set $d_{ji} = h_i$. All nodes are assumed inactive initially and nodes remain active once influences from neighbors and incentives reach or exceed the threshold. For each node $i \in V$, let nonnegative continuous variables x_i be the amount of partial incentives given to user i, binary variables y_{ij} indicate whether influence is exerted from node i to j nor not, and binary variables z_i indicate whether node i is activated or not. The arc-based formulation of static

LCIM is given by

$$
\begin{aligned}
\text{(LCIM)} \min_{x,y,z} \quad & \sum_{i \in V} x_i \\
\text{s.t. } & x_i + \sum_{j \in N(i)} d_{ji} y_{ji} \geq h_i z_i \quad \forall i \in V && \text{(3.1a)} \\
& y_{ij} + y_{ji} \leq z_i \quad \forall (i,j) \in E && \text{(3.1b)} \\
& \sum_{i \in V} z_i \geq b && \text{(3.1c)} \\
& \sum_{(i,j) \in C} y_{ij} \leq \sum_{i \in V(C) \setminus \{k\}} z_i \quad \forall k \in V(C), \forall \text{ cycles } C \subseteq E && \text{(3.1d)} \\
& x \in \mathbb{R}^n_+ \\
& y \in \mathbb{B}^m, z \in \mathbb{B}^n.
\end{aligned}
$$

Node propagation constraints (3.1a) follow the linear threshold model by evaluating the total incoming influence from neighbor plus the incentives given to a node. Constraints (3.1b) state that for every two nodes with bidirectional arcs, the influence exertion is allowed in one way only. The cardinality constraint (3.1c) describes the desired number of activated nodes given a predetermined penetration rate a, where $b = \lceil an \rceil$ and $1 \leq b \leq n$. Constraints (3.1d) are generalized cycle elimination constraints (GCEC) where $V(C) = \{i \in V : (i,j) \in C\}$. They are necessary to produce the acyclic optimal influence propagation graph. Note that the arc-based formulation proposed by [1] is different from this chapter as the influence weights in their model are coming solely from their neighbors without incentives. Fischetti et al. [18] adopt this arc-based formulation on arbitrary graphs for computational performance comparison, but possible values of incentives are discretized by a set of an exponential number of binary variables.

3.1.2 Main contributions

The key contributions of this chapter are summarized as follows. We conduct a polyhedral study for the hidden mixed 0-1 knapsack substructure in (3.1a) of LCIM. We propose three classes of strong valid inequalities, namely, the continuous cover, continuous packing and the minimum influencing subset inequalities from this substructure and specify the conditions under which they are facet-defining. The coefficient of these inequalities can be adjusted to equal influence assumption directly. We have improved the run time of the separation algorithm for the the continuous cover and continuous packing inequalities compared with our preliminary results presented in [11]. In addition, we provide a polynomial-size complete linear description of the polyhedron of LCIM on a tree when equal influence weights for every node and 100% coverage are assumed. For LCIM on arbitrary bidirectional network graphs, we derive a novel class of strong valid inequalities called the the (U, C) inequalities and show they dominate the general cycle elimination constraints. Finally, we augment the preliminary computations in [11] with different formulations, cutting plane strategies, density and scale of networks in extensive computational experiments.

3.1.3 Outline

The remainder of this chapter is organized as follows. In Section 3.2, we derive strong valid inequalities from the mixed 0-1 knapsack substructure and give an exact polynomial time separation algorithm for these inequalities. In Section 3.3, we discuss an $O(n)$ time dynamic programming recursion for LCIM on a simple cycle. We propose the (U, C) inequalities for arbitrary bidirectional graphs and develop a polynomial time separation algorithm via solving the longest path problem on a directed acyclic graph. We present the convex hull description of LCIM on a special case in Section 3.4. Finally, we illustrate the effectiveness of our proposed valid inequalities in the computational experiments in Section 3.5 and conclude

with Section 3.6.

3.2 Valid inequalities based on mixed 0-1 knapsack polyhedron

To develop a strong formulation for LCIM, we study the polyhedral structure of constraints (3.1a). For $i \in V$, let

$$\mathcal{P}_i = \left\{ (x_i, y, z_i) \in \mathbb{R}_+ \times \mathbb{B}^{v_i+1} : x_i + \sum_{j \in N(i)} d_{ji} y_{ji} \geq h_i z_i \right\}.$$

The set $\mathcal{P}_i$ is a mixing set with a binary variable on the right-hand side value. For any inequality that is facet-defining for $\text{conv}\,(\mathcal{P}_i)$, it is facet-defining for $\text{conv}\,(\cap_{i \in V} \mathcal{P}_i)$ as well. Therefore, we now consider a single node propagation by dropping the subscript i and obtain the following set

$$\mathcal{P} = \left\{ (x, y, z) \in \mathbb{R}_+ \times \mathbb{B}^{v+1} : x + \sum_{j \in N} d_j y_j \geq hz \right\}.$$

Observe that the set $\mathcal{P}$ contains a mixed 0-1 knapsack structure. Let set $\overline{\mathcal{P}}$ obtained from $\mathcal{P}$ by setting $\overline{y}_j = 1 - y_j$, $j \in N$ and $z = 1$, then we have a mixed 0-1 knapsack set $\overline{\mathcal{P}}$ with weight d_j for each item $j \in N$ and the capacity of knapsack $\sum_{j \in N} d_j - h$ plus an unbounded continuous variable x in the following

$$\overline{\mathcal{P}} = \left\{ (x, \overline{y}, z) \in \mathbb{R}_+ \times \mathbb{B}^{v} \times \{1\} : \sum_{j \in N} d_j \overline{y}_j \leq \left(\sum_{j \in N} d_j - hz \right) + x \right\}.$$

The mixed 0-1 knapsack set $\overline{\mathcal{P}}$ is a special case of traditional 0-1 knapsack problem where the knapsack size is expanded with additional capacity. Observe that $\dim(\mathcal{P})$ is full-dimensional and contains the origin. There are trivial facets for $\text{conv}\,(\mathcal{P})$ and their verification is straightforward.

Proposition 3. *The following facet-defining inequalities for* $\text{conv}\,(\mathcal{P})$ *are trivial.*

(i) The inequality $x + \sum_{j\in N} d_j y_j \geq hz$ *is facet-defining for* $\text{conv}\,(\mathcal{P})$ *if* $d_j \leq h$ *for all* $j \in N$.

(ii) The inequality $x \geq 0$ *is facet-defining for* $\text{conv}\,(\mathcal{P})$ *if* $d_j \leq h$ *for all* $j \in N$.

(iii) The inequality $y_j \geq 0$ *is facet-defining for* $\text{conv}\,(\mathcal{P})$.

(iv) The inequality $y_j \leq 1$ *is facet-defining for* $\text{conv}\,(\mathcal{P})$.

Marchand and Wolsey [37] propose two classes of valid inequalities for $\overline{\mathcal{P}}$ based on mixed-integer rounding and lifting function, namely, the continuous cover and continuous packing (reversed cover) inequalities. We follow a similar idea to strengthen the formulation for LCIM, with moderate modification as $\overline{\mathcal{P}} \subset \mathcal{P}$. Applications of continuous cover and continuous packing inequalities for $\overline{\mathcal{P}}$ can be seen in delay management for public transportation [15], job scheduling with uncertain multiple resources [29], discrete lot sizing [35] and single-item capacitated lot sizing [39]. They can also be extended to solve general mixed-integer optimization that contains mixed 0-1 knapsack set with bounded continuous variables, see [44, 51, 52].

3.2.1 Continuous cover and continuous packing inequalities

Consider a continuous cover $S := \{1, \ldots, s\} \subseteq N$ such that $h + \sum_{j\in S} d_j - \sum_{j\in N} d_j = \pi > 0$ and $h + \sum_{j\in S\setminus\{k\}} d_j - \sum_{j\in N} d_j < 0$ for any $k \in S$. Let $d_j \in S$ be in non-increasing order with $d_1 \geq \ldots \geq d_r > \pi \geq d_{r+1} \geq \ldots \geq d_s$, $D_j = \sum_{k=1}^{j} d_k$ for $j \leq r$ and $D_0 = 0$.

Proposition 4. *The continuous cover inequality*

$$x + \sum_{j\in S} \min\{\pi, d_j\} y_j + \sum_{j\in N\setminus S} \Phi(d_j) y_j \geq \left(\min_{j\in S}\{\pi, d_j\} + \sum_{j\in N\setminus S} \Phi(d_j) \right) z \tag{3.2}$$

where

$$\Phi(d) = \begin{cases} j\pi & D_j \leq d \leq D_{j+1} - \pi, \quad j \in [0, r-1] \\ j\pi + d - D_j & D_j - \pi \leq d \leq D_j, \quad j \in [1, r-1] \\ r\pi + d - D_r & D_r - \pi \leq d, \end{cases} \tag{3.3}$$

is valid for $\mathcal{P}$.

Similarly, consider a continuous packing $L := \{1, \ldots, l\} \subseteq N$ such that $\sum_{j \in L} d_j - h = \lambda > 0$ and $\sum_{j \in L \setminus \{k\}} d_j - h < 0$ for any $k \in L$. Let $d_j \in L$ be in non-increasing order with $d_1 \geq \ldots \geq d_r > \lambda \geq d_{r+1} \geq \ldots \geq d_l$, $D_j = \sum_{k=1}^{j} d_k$ for $j \leq r$ and $D_0 = 0$.

Proposition 5. *The continuous packing inequality*

$$x + \sum_{j \in L} \max\{0, d_j - \lambda\} y_j + \sum_{j \in N \setminus L} \Psi(d_j) y_j \geq \left(\sum_{j \in L} \max\{0, d_j - \lambda\} \right) z \tag{3.4}$$

where

$$\Psi(d) = \begin{cases} d - j\lambda & D_j \leq d \leq D_{j+1} - \lambda, \quad j \in [0, r-1] \\ D_j - j\lambda & D_j - \lambda \leq d \leq D_j, \quad j \in [1, r-1] \\ D_r - \lambda r & D_r - \lambda \leq d, \end{cases} \tag{3.5}$$

is valid for $\mathcal{P}$.

Here we omit the proofs as the validity of inequalities (3.2) and (3.4) and their facet-defining conditions for $\operatorname{conv}(\mathcal{P})$ directly follow from [37] when $z = 1$, while when $z = 0$, we must have $y_j = 0$ for all $j \in N$, both inequalities are trivially satisfied and facet-defining according to Proposition 3.

Example 2. Let $d = (7, 6, 5, 4)$ and $h = 8$, we list the facet-defining inequalities of inequality (3.2) and (3.4) in Table 3.1. For example, for $S = \{1, 2, 4\}$, we have $\pi = 3$ and $r = 3$. Then the lifting function Φ is given by

$$\Phi(d) = \begin{cases} 0 & 0 \leq d \leq 4 \\ 3 & 7 \leq d \leq 10 \\ 6 & 13 \leq d \leq 14 \\ d-4 & 4 \leq d \leq 7 \\ d-7 & 10 \leq d \leq 13 \\ d-8 & 14 \leq d \end{cases}$$

Hence, the coefficient of y_3 is $\Phi(d_3) = \Phi(5) = 5 - 4 = 1$, which generates

$$x + 3y_1 + 3y_2 + y_3 + 3y_4 \geq 4z.$$

Table 3.1: Continuous cover and continuous packing inequalities of Example 2

$x + 7y_1 + 6y_2 + 5y_3 + 4y_4 \geq 8z$	
set	facet-defining inequality
$S = \{2, 3, 4\}$, $L = \{1, 2\}$	$x + y_1 + y_2 + y_3 + y_4 \geq 2z$
$S = \{1, 3, 4\}$, $L = \{1, 2\}$	$x + 2y_1 + y_2 + 2y_3 + 2y_4 \geq 3z$
$S = \{1, 2, 4\}$, $L = \{1, 3\}$	$x + 3y_1 + 3y_2 + y_3 + 3y_4 \geq 4z$
$S = \{1, 2, 3\}$, $L = \{1, 4\}$	$x + 4y_1 + 4y_2 + 4y_3 + y_4 \geq 5z$
$S = \{1, 2, 4\}$, $L = \{2, 3\}$	$x + 4y_1 + 3y_2 + 2y_3 + 3y_4 \geq 5z$
$S = \{1, 2, 3\}$, $L = \{2, 4\}$	$x + 5y_1 + 4y_2 + 4y_3 + 2y_4 \geq 6z$
$S = \{1, 2, 3\}$, $L = \{3, 4\}$	$x + 6y_1 + 5y_2 + 4y_3 + 3y_4 \geq 7z$

Essentially, the continuous cover inequalities (3.2) and packing inequalities (3.4) are not sufficient to describe conv $(\mathcal{P})$, as the additional binary variable z creates new extreme points. Next, we introduce a new class of valid inequalities for $\mathcal{P}$ that utilizes the concept of minimal influencing set.

3.2.2 Minimum influencing subset inequalities

We use the definition of minimal influencing set from [18], which we include here for the reader's convenience.

Definition 10 ([18]). *Let $p_i \in [0, h_i - 1]$ be an incentive payment to node $i \in V$ and $M \subseteq N(i)$ be a set of active neighbors of node $i \in V$, such that $p_i + \sum_{j \in M} d_{ji} = h_i$. We say M is a minimal influencing subset for node $i \in V$ if and only if for a fixed incentive payment $\overline{p}_i$, it satisfies $\overline{p}_i + \sum_{j \in M} d_{ji} = h_i$ and $\overline{p}_i + \sum_{j \in M \setminus \{k\}} d_{ji} < h_i$ for any $k \in M$. In other words, a strict subset of M with the same incentive payment is not sufficient to activate node i.*

Proposition 6. *Let $M \subseteq N$ be a minimum influencing subset with an incentive payment $p = h - \sum_{j \in M} d_j$. The minimal influencing subset inequality*

$$x + \sum_{j \in N \setminus M} \min\{d_j, p\} y_j \geq pz \tag{3.6}$$

is valid for $\mathcal{P}$.

Proof. If $z = 0$ then inequality (3.6) is trivially satisfied. If $y_j = 0$ for all $j \in N \setminus M$, either $x = 0$ for $z = 0$ or $x = p$ for $z = 1$. Assume that none of these cases hold, given a $p > 0$, rewrite the left term of the inequality in the following form:

$$\begin{aligned} & x + \sum_{j \in N} d_j y_j \\ = \; & x + \sum_{j \in N \setminus M : d_j \leq p} d_j y_j + p \sum_{j \in N \setminus M : d_j > p} y_j + \sum_{j \in M} d_j y_j \geq h, \end{aligned}$$

which implies

$$x + \sum_{j \in N \setminus M : d_j \leq p} d_j y_j + p \sum_{j \in N \setminus M : d_j > p} y_j \geq h - \sum_{j \in M} d_j y_j \geq h - \sum_{j \in M} d_j = p.$$

□

Proposition 7. *Inequality* (3.6) *is facet-defining for* $\operatorname{conv}(\mathcal{P})$ *if and only if $p > 0$. Moreover, for a given $i \in V$ and a set $N(i)$, for each $M \subseteq N(i)$ such that $h_i - \sum_{j \in M} d_{ji} = p_i > 0$, the minimal influencing subset inequality*

$$x_i + \sum_{j \in N(i) \setminus M} \min\{d_{ji}, p_i\} y_{ji} \geq p_i z_i \tag{3.7}$$

is facet-defining for $\operatorname{conv}(\cap_{i \in V} \mathcal{P}_i)$.

Proof. Recall that $p \in [0, h-1]$ by definition. If $p = 0$, then inequality (3.6) reduces to $x \geq 0$; therefore, $p > 0$ is necessary. To show the sufficiency that inequality (3.6) is facet-defining for $\text{conv}(\mathcal{P})$, we exhibit $v + 1$ linearly independent points $(x, \mathbf{y}, z)$ on the face defined by inequality (3.6). Let $e_j \in \mathbb{B}^v$ be the unit vector corresponding to y_j for $j \in N$. Consider the two feasible points $(p, \sum_{j \in M} e_j, 1)$ and $(0, \sum_{j \in M} e_j, 0)$, then, consider the $v - 1$ feasible points $(0, \sum_{j \in M}(e_j + e_k), 1)$ for $k \in L \setminus M$. It is straightforward to verify that these $v + 1$ points are linearly independent and satisfy inequality (3.6) at equality. To prove the second part of this proposition, let $\eta_i \in \mathbb{B}^{2n+m}, \mu_{ij} \in \mathbb{B}^{2n+m}$, and $\zeta_i \in \mathbb{B}^{2n+m}$ for $i \in V, j \in N(i)$ be the unit vectors corresponding to variables x_i, y_{ij}, and z_i, respectively. The component of η_i is 1 if it has the same position with x_i in the feasible solution; all other components of η_i are 0. Similar setting is made to μ_{ij} for y_{ij} and ζ_i for z_i, respectively. For $i \in V$, consider the n points $p_i\eta_i + \sum_{j \in M} \mu_{ji} + \zeta_i$, also, consider the n points $\sum_{j \in M} \mu_{ji}$. For $i \in V$ and $k \in L \setminus M$, consider the $m - 1$ points $\sum_{j \in M}(\mu_{ji} + \mu_{ki}) + \zeta_i$. These $2n + m - 1$ points are linearly independent and satisfy inequality (3.7) at equality, which completes the proof. □

Example 1 (Continued)**.** The facet-defining inequalities of (3.6) for Example 2 are listed in Table 3.2.

Table 3.2: Minimal influencing subset inequalities of Example 2

$x + 7y_1 + 6y_2 + 5y_3 + 4y_4 \geq 8z$	
set	facet-defining inequality
$M = \{1\}$	$x + y_2 + y_3 + y_4 \geq z$
$M = \{2\}$	$x + 2y_1 + 2y_3 + 2y_4 \geq 2z$
$M = \{3\}$	$x + 3y_1 + 3y_2 + 3y_4 \geq 3z$
$M = \{4\}$	$x + 4y_1 + 4y_2 + 4y_3 \geq 4z$

Although inequalities (3.2), (3.4) and (3.6) define a large number of facets for conv $(\mathcal{P})$, they are not sufficient to completely describe conv $(\mathcal{P})$ in its original space of variables. Particularly, the following inequality is valid and facet-defining for Example 2 but can not be obtained through either inequalities (3.2), (3.4) or (3.6):

$$x + 3y_1 + 2y_2 + 2y_3 + 2y_4 \geq 4z.$$

3.2.3 Separation of minimal influencing subset inequalities

In this section, we give an exact polynomial time separation algorithm for finding the most violated minimal influencing subset inequality. From inequality (3.7), we observe that finding the most violated inequality for a given fractional solution $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in \mathbb{R}_+^{2n+m}$ consists of choosing a set $M \subseteq N(i)$ such that $p_i z_i - \sum_{j \in N(i) \setminus M} \min\{d_{ji}, p_i\} y_{ji}$ is maximized. Let $\hat{v} := \max\{v_i : i \in V\}$.

Proposition 8. *Given a fractional solution $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in \mathbb{R}_+^{2n+m}$ from solving LCIM, there exists an $O(n\hat{v} \log \hat{v})$ time separation algorithm for inequality* (3.7).

Proof. A violated inequality (3.7) can be found if

$$p_i \left(z_i^* - \sum_{j \in N(i) \setminus M : d_{ji} > p_i} y_{ji}^* \right) - \sum_{j \in N(i) \setminus M : d_{ji} \leq p_i} d_{ji} y_{ji}^* > x_i^*,$$

which implies that it suffices to consider y_{ji}^* for some $j \in N(i)$ such that $z_i^* - \sum_{j \in N(i)} y_{ji}^* > 0$ and $p_i > 0$. To do so, we sort y_{ji}^* in non-decreasing order for $j \in N(i)$ with indices $j_1, j_2, \ldots, j_v$ such that $y_{j_1 i}^* \leq y_{j_2 i}^* \leq \ldots \leq y_{j_v i}^*$. For $j_1 \leq j_k \leq j_v$, we sum up first k elements, then we check if $z_i^* - \sum_{\ell=1}^{k} y_{j_\ell i}^* > 0$ and $p_i' = h_i - \sum_{\ell=k+1}^{v} d_{j_\ell i} > 0$, until $z_i^* - \sum_{\ell=1}^{k+1} y_{j_\ell i}^* < 0$. These k elements constitute the subset M and $N(i) \setminus M$ simultaneously and ensure $z_i^* - \sum_{j \in N(i) \setminus M} y_{ji}^* > 0$ and $p_i > 0$ in order to generate a violated cut. The set M corresponds to the most

violated cut can be determined by evaluating $\max\{0, p'_i(z^*_i - \sum_{\ell=1}^{k} y^*_{j_\ell i}) : k \in [1, v]\}$. If $\max\{0, p'_i(z^*_i - \sum_{\ell=1}^{k} y^*_{j_\ell i}) : k \in [1, v]\} = 0$, then there are no violated cuts. The sorting process runs in $O(\hat{v} \log \hat{v})$ time and the evaluation takes $O(\hat{v})$, since we have to check for every node $i \in V$, overall the separation algorithm runs in $O(n\hat{v} \log \hat{v})$ time. □

3.2.4 Separation of continuous cover and continuous packing inequalities

Up to this point, we presented an exact polynomial time separation algorithm for inequalities (3.7). Next, we show that a violated continuous cover inequality for $\text{conv}\,(\cap_{i \in V} \mathcal{P}_i)$ can be identified by exploiting the result of Proposition 8. Then we can obtain a violated continuous packing inequality in polynomial time after finding a violated continuous cover inequality. We first formally establish the relationship between S, L and M in the following lemma.

Lemma 1. *If $p = h - \sum_{j \in M} d_j > 0$ and there exists $k \in N \setminus M$ such that $\sum_{j \in M \cup \{k\}} d_j > h$ and $\sum_{j \in M \cup \{k\} \setminus \{\ell\}} d_j < h$ for any $\ell \in M$, then $p = \pi$, $S = N \setminus M$, $\sum_{j \in M \cup \{k\}} d_j - h = \lambda$ and $L = M \cup \{k\}$.*

Proof. By rearranging the terms in the definition of p,

$$\begin{aligned} p &= h - \sum_{j \in M} d_j > 0 \\ &= h + \sum_{j \in N \setminus M} d_j - \sum_{j \in N} d_j > 0 \\ &= h + \sum_{j \in S} d_j - \sum_{j \in N} d_j > 0. \end{aligned}$$

Hence, we can see that S is equivalent to $N \setminus M$ and $p = \pi$ if $p > 0$. Next, if there exists an element $k \in N \setminus M$ such that $\sum_{j \in M \cup \{k\}} d_j > h$, immediately we have $M \cup \{k\} = L$ by definition and $\sum_{j \in M \cup \{k\}} d_j - h = \lambda$. Since $N \setminus M \cup \{M \cup \{k\}\} = N$, we have $S \cup L = N$, $S \cap L = \{k\}$, and $L \setminus \{k\} = N \setminus S = M$. □

Following Lemma 1, we present an efficient separation procedure to determine violated continuous cover and continuous packing inequalities by using the information of the set M. Note that here we add an index i to inequalities (3.2) similar to (3.7) for all $i \in V$.

Proposition 9. *There exists a violated continuous cover inequality if a violated inequality* (3.7) *is identified.*

Proof. Recall that inequality (3.7) is violated if

$$p_i \left(z_i^* - \sum_{j \in N(i) \setminus M : d_{ji} > p_i} y_{ji}^* \right) - \sum_{j \in N(i) \setminus M : d_{ji} \leq p_i} d_{ji} y_{ji}^* > x_i^*,$$

or equivalently by Lemma 1,

$$\pi_i z_i^* - \pi_i \sum_{j \in S : d_{ji} > \pi_i} y_{ji}^* - \sum_{j \in S : d_{ji} \leq \pi_i} d_{ji} y_{ji}^* > x_i^*.$$

Now, a continuous cover inequality for a fixed node $i \in V$ is violated if

$$\min_{j \in S} \{\pi_i, d_{ji}\} z_i^* + \sum_{j \in N(i) \setminus S} \Phi(d_{ji})(z_i^* - y_{ji}^*) - \sum_{j \in S} \min\{\pi_i, d_{ji}\} y_{ji}^* > x_i^*.$$

Suppose $d_{ji} > \pi_i$ for all $j \in S$, then the left term of the continuous cover inequality can be further written as

$$\pi_i z_i^* + \sum_{j \in N(i) \setminus S} \Phi(d_{ji})(z_i^* - y_{ji}^*) - \pi_i \sum_{j \in S : d_{ji} > \pi_i} y_{ji}^* - \sum_{j \in S : d_{ji} \leq \pi_i} d_{ji} y_{ji}^*.$$

Since $(z_i^* - y_{ji}^*) \geq 0$ holds and the lifting function Φ is nonnegative, the rest of the terms already violate the current solution $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$, we then obtain a violated continuous cover inequality with π_i being the minimum among $\min\{\pi_i, d_{ji} : j \in S\}$. This suggests that, when a violated inequality (3.7) is found, it suffices to generate a violated continuous cover

inequality concurrently. □

On the other hand, a violated continuous packing inequality can not be obtained directly from separating inequality (3.7). However, when a violated continuous cover inequality is found, we can check every element in S to see if there exists an element k such that $\sum_{j\in N\setminus S\cup\{k\}} d_{ji} > h_i$. For every k satisfying this condition, the packing set L is then determined. A violated continuous packing inequality is found if

$$\sum_{j\in L} \max\{0, d_{ji} - \lambda_i\}(z_i^* - y_{ji}^*) - \sum_{j\in N(i)\setminus L} \Psi(d_{ji}) y_{ji}^* > x^*$$

holds. Furthermore, suppose $\hat{S} = \max\{|S| : S \subseteq N(i), i \in V\}$, the process of checking elements in S takes $O(\hat{S})$ time and the function Ψ can be constructed in $O(\hat{v}\log\hat{v})$ time using binary search proposed in [44].

3.3 Valid inequalities for LCIM with cycles

In this section, we expand the study of conv $(\cap_{i\in V}\mathcal{P}_i)$ to incorporate the remaining constraints in LCIM on an arbitrary bidirectional graph that contains cycles. The polyhedron that describes the intersection of these constraints is

$$\mathcal{Q} = \left\{(x, y, z) \in \mathbb{R}_+^n \times \mathbb{B}^{n+m} : (3.1a) - (3.1d)\right\}.$$

To simplify the notation, we let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be the coefficients associated with variables $\mathbf{y}$ and $\mathbf{z}$ corresponding to continuous cover and continuous packing inequalities. In other words, we express them in the following form:

$$x_i + \sum_{j\in N(i)} \alpha_{ji} y_{ji} \geq \beta_i z_i, \tag{3.8}$$

and it is facet-defining for conv $(\mathcal{Q})$. Due to the straightforward structure of a cycle, the minimum incentive of the influence propagation can be easily characterized. Raghavan and Zhang [50] give an $O(n)$ time algorithm for the weighted target set selection problem on a cycle. We also present a polynomial time dynamic programming to solve LCIM on a simple cycle. Then we give a class of exponential number of valid inequalities that forms acyclic influence propagation by exploiting inequality (3.8) as the base inequality. We also demonstrate that the separation for this class of valid inequalities can be done in polynomial time for arbitrary bidirectional graphs that contain cycles.

3.3.1 Dynamic programming recursion for LCIM on a simple cycle

Without loss of generality, we assume $|V(C)| = |V| = n$ in this section. Hence, for a smple cycle graph, we have $V = V(C) = n$, $E = C$ and $|E| = 2n$ with $v_i = 2$ for all $i \in V$. We still use $V(C)$ for the set of all nodes and C for set of all bidirectional arcs to focus on the discussion of LCIM on a simple cycle for consistency. Observe that due to the cycle structure, the influence propagation occurs on an induced path of a cycle as it is one-way and consecutive after a particular node is activated by paying full incentive to it. Given the cardinality requirement $b \leq n - 1$, the cost of activating other nodes on this path is equivalent to the threshold of the inactivated node minus the influence weights exerted from the activated predecessor on the side. For $b = n$, the cost of activating the last node is zero as it receives influence exertion from both its predecessor and the firstly activated node. Therefore, we only need to evaluate the cases for $b \leq n - 1$.

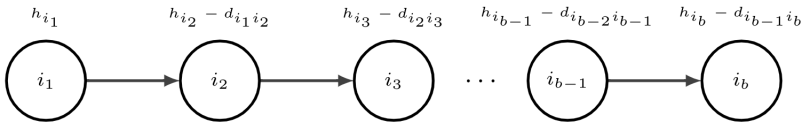


Figure 3.1: An induced subgraph of a cycle is an one-way path

Given $b \leq n-1$, for $i \in V(C)$, construct a node set $\overrightarrow{V}_{ib}$ that contains node i with forward arcs connecting nodes starting from i with total number of nodes equals to b. Let $i_1, i_2, \ldots, i_{b-1}, i_b$ be the indices in $\overrightarrow{V}_{ib}$. The corresponding set of forward arcs is then $\overrightarrow{C}_{ib} = \{(i_j, i_{j+1}) : j \in [1, b-1]\}$ as illustrated in Figure 3.1. We apply the similar logic for backward set of nodes $\overleftarrow{V}_{ib}$ and arcs $\overleftarrow{C}_{ib}$. Let $F(i,b)$ denote the minimum cost of activating b nodes on the cycle beginning with node i. For $1 \leq b \leq n-1$, the minimum cost of activating b nodes on a cycle is given by

$$\min_{i \in V(C)} F(i,b) \tag{3.9}$$

where

$$F(i,b) = \min \left\{ h_i + \sum_{(j,k) \in \overrightarrow{C}_{ib}} (h_k - d_{jk}), \quad h_i + \sum_{(j,k) \in \overleftarrow{C}_{ib}} (h_k - d_{jk}) \right\}. \tag{3.10}$$

Proposition 10. *The dynamic programming recursion given by* (3.9) *and* (3.10) *solves LCIM on a simple cycle in* $O(n)$ *time.*

Proof. The recursion (3.10) evaluates the minimum cost of activating b nodes on a path beginning with node i on both directions. As we only need to compare the minimum of (3.10) for every node, we obtain the optimal objective function of LCIM on a simple cycle

in $O(n)$ time. $\square$

3.3.2 Valid inequalities for influence propagation over a cycle

Since every network consists of trees and cycles as substructures, observe that for every cycle in the network, at least one node is either paid with full incentive or the activation requires influence exertion from nodes outside the cycle. Fischetti et al. [18] first recognized this observation and proposed a generalized propagation constraints in a different space of variables. Here we propose an exponential class of valid inequalities that captures this observation as well as ensures the influence propagation is acyclic for conv $(\mathcal{Q})$.

Proposition 11. *Given an inequality* (3.8) *and a cycle with set of nodes* $V(C)$ *and set of arcs* C*, for* $U \subseteq V(C)$*, the* (U, C) *inequality*

$$\sum_{i \in U} \gamma_i \left(x_i + \sum_{j \in N(i)} \alpha_{ji} y_{ji} - \beta_i z_i \right) \geq \delta(U) \left(1 - \sum_{(k,\ell) \in C : \ell \notin U} (z_\ell - y_{k\ell}) \right) \tag{3.11}$$

is valid for $\mathcal{Q}$*, where*

(i) $\omega_i = h_i - \beta_i + \sum_{j \in N(i): j \notin V(C)} (\alpha_{ji} - d_{ji})$,

(ii) $\delta(U) = \omega_i$ *if* $|U| = 1$,

(iii) $\delta(U)$ *computes the least common multiple of* ω_i *for* $i \in U$ *if* $|U| \geq 2$,

(iv) $\gamma_i = \frac{\delta(U)}{\omega_i}$.

Proof. Due to constraint (3.1c), there exists $z_i = 1$ for some $i \in V(C)$. Let $H = \{(k, \ell) \in C : \ell \notin U\}$. We partition H into two disjoint sets H_0 and H_1 such that $H = H_0 \cup H_1$ and $H_0 \cap H_1 = \varnothing$, where $H_0 = \{(k, \ell) \in H : k \notin U\}$ and $H_0 = \{(k, \ell) \in H : k \in U\}$. We distinguish two main cases:

Case 1. *We first consider* $z_i = 0$ *for all* $i \in U$ *and* $z_i = 1$ *for all* $i \in V(C) \setminus U$*. In this case, the left hand side of the inequality is equal to 0, whereas in the right hand side, we have* $k \in V(C) \setminus U$ *and* $\ell \in\in V(C) \setminus U$*, if there exists at least one* $y_{k\ell} = 1$*, we must have* $z_k = z_\ell = 1$*, which leads to* $|V(C) \setminus U| > |H_0|$*. Consequently,*

$$\delta(U) \left(1 - \sum_{i \in V(C) \setminus U} z_i + \sum_{(k,\ell) \in H_0} y_{k\ell} \right) < 0$$

holds, inequality (3.11) *is thus valid.*

Case 2. *Next we consider* $z_i = 1$ *for all* $i \in U$ *and* $z_i = 0$ *for all* $i \in V(C) \setminus U$*. In this case, we must have* $y_{k\ell} = 0$ *for all* $(k, \ell) \in H$ *as influence exertion towards nodes belong to* $V(C) \setminus U$ *is unnecessary. Then the inequality reduces to*

$$\sum_{i \in U} \gamma_i \left(x_i + \sum_{j \in N(i)} \alpha_{ji} y_{ji} - \beta_i \right) \geq \delta(U).$$

Regardless of whether $y_{ji} = 0$ *for all* $i \in N(i)$ *or* $y_{ji} = 1$ *for some* $j \in N(i)$*, there exists at least one* $i \in U$ *such that* $x_i = h_i$ *and* $y_{ji} = 0$*. In other words, at least one node is activated with full incentive payment in order to launch the propagation for nodes in* U*. For other nodes that receive partial or zero incentives, their corresponding term in the left hand side is zero. To compute the possible range of the left hand side, we rearrange the terms and obtain the following*

$$\begin{aligned} & \sum_{i \in U} \gamma_i (h_i - \beta_i) \\ = & \sum_{i \in U} \frac{\delta(U)}{(h_i - \beta_i)} (h_i - \beta_i) \\ = & \sum_{i \in U} \delta(U). \end{aligned}$$

Consequently, the left hand side is at most $|U|\delta(U)$ and at least $\delta(U)$ with one particular node with full incentive $x_i = h_i$, which is valid. This completes the proof.

□

Example 3. Consider a graph G illustrated in Figure 3.2. The number next to each arc is the influence weight d_{ij}, while the number inside the brackets next to each node is the threshold h_i.

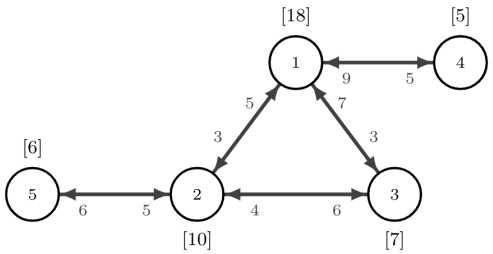


Figure 3.2: A social network with $n = 5$.

For node 2, take $L = \{1, 3, 5\}$ with $\lambda_2 = 3 + 4 + 5 - 10 = 2$, the corresponding continuous packing inequality (3.8) is

$$x_2 + y_{12} + 2y_{32} + 3y_{52} \geq 6z_2.$$

Similarly, for node 3, take $L = \{1, 2\}$ with $\lambda_3 = 6 + 3 - 7 = 2$, the corresponding continuous packing inequality (3.8) is

$$x_3 + y_{13} + 4y_{23} \geq 5z_3.$$

There are two cycles $\{(1, 2), (2, 3), (3, 1)\}$ and $\{(1, 3), (3, 2), (2, 1)\}$ in Figure 3.2. For $U = \{2, 3\}$ with cycle $C = \{(1, 2), (2, 3), (3, 1)\}$, we have $\omega_2 = 10 - 6 + 3 - 5 = 2$, $\omega_3 = 7 - 5 = 2$,

$\delta(\{2,3\}) = \texttt{lcm}(2,2) = 2$, then the (U, C) inequality is

$$(x_2 + y_{12} + 2y_{32} + 3y_{52} - 6z_2) + (x_3 + y_{13} + 4y_{23} - 5z_3) \geq 2(1 - z_1 + y_{31}).$$

Furthermore, consider the cycle $C = \{(1,3),(3,2),(2,1)\}$ from another direction, the (U, C) inequality is

$$(x_2 + y_{12} + 2y_{32} + 3y_{52} - 6z_2) + (x_3 + y_{13} + 4y_{23} - 5z_3) \geq 2(1 - z_1 + y_{21}).$$

Next, we study the strength of inequality (3.11). Note that conv $(\mathcal{Q})$ is full-dimensional and does not pass through the origin. Without loss of generality, we assume that $\delta(U) = 1$ for $U = \varnothing$.

Proposition 12. *Inequality* (3.11) *with* $U = \varnothing$ *dominates the generalized cycle elimination constraint.*

Proof. Consider a particular cycle C, the GCEC can be states as

$$\sum_{(i,j)\in C} (z_j - y_{ij}) \geq z_k,$$

where $k \in V(C)$ is an arbitrary choice of index among $V(C)$. For a (U, C) inequality (3.11) with $U = \varnothing$ on this cycle, we obtain

$$\sum_{(i,j)\in C} (z_j - y_{ij}) \geq 1.$$

Clearly, the GCEC is weaker then the (U, C) inequality unless $z_k^* = 1$. Furthermore, if we have $\sum_{(i,j)\in C}(z_j - y_{ij}) < z_k$, then $\sum_{(i,j)\in C}(z_j - y_{ij}) < 1$ must hold. This implies that there exist violated (U, C) inequalities for every violated cycle C identified. □

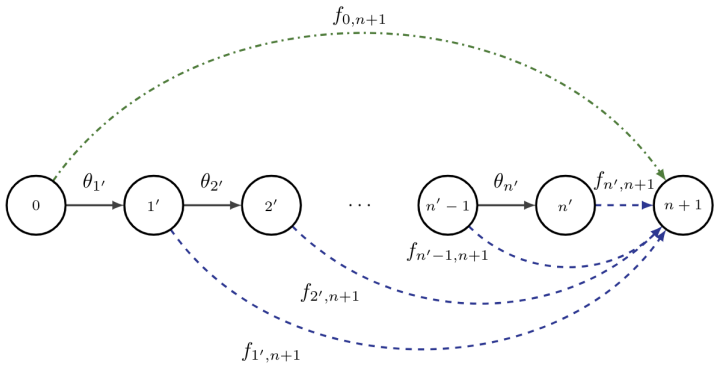


Figure 3.3: Graph $\mathcal{D}$ for separation of inequality (3.11)

3.3.3 Separation of (U, C) inequalities

Since the size of of inequalities (3.11) is exponential, we explore a separation scheme to find the most violated inequality corresponding to set U in polynomial time. We again assume that $|V(C)| = n$ in this section.

Proposition 13. *Separation problem for inequality* (3.11) *can be solved in* $O(n^3 \log n)$ *time.*

Proof. Inequality (3.11) is violated if

$$\delta(U)\left(1 - \sum_{(k,\ell)\in C:\ell\notin U}(z^*_\ell - y^*_{k\ell})\right) - \sum_{i\in U}\gamma_i\left(x^*_i + \sum_{j\in N(i)}\alpha_{ji}y^*_{ji} - \beta_i z^*_i\right) > 0. \tag{3.12}$$

For a given fractional point $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in \mathcal{Q}$, we determine sets $U \subseteq V(C)$ such that the left hand side of (3.12) is maximized. With the observation in Proposition 12, for every violated cycle detected, we construct a longest path problem on a directed acyclic network to solve the separation problem.

Consider a directed acyclic network $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ with a source vertex $0 \in \mathcal{V}$ and a sink vertex $n+1 \in \mathcal{V}$. Define $\theta_i = x^*_i + \sum_{j\in N(i)}\alpha_{ji}y^*_{ji} - \beta_i z^*_i$ for all $i \in [1, n]$. It is possible

that there exists multiple inequality (3.8) for a fixed i. We select the one with the minimum value of θ_i accordingly. Let index set $\{i' : i \in [1,n]\}$ such that $\theta_{1'} \leq \theta_{2'} \leq \ldots \leq \theta_{n'}$. Node i' is sorted according the the value of θ_i and each node $i' \in \mathcal{V}$ has a unique mapping to each node $i \in V(C)$. The node set $\mathcal{V}$ is then $\{0, n+1\} \cup \{i' : i \in [1,n]\}$. The arc set is $\mathcal{A} = \{(0,n+1)\} \cup \{(0,1')\} \cup \{(i',(i+1)') : i \in [1,n-1]\} \cup \{(i',n+1) : i \in [1,n]\}$.

Next, we assign length on each arc in $\mathcal{A}$. For arc $(0,n+1)$, we let

$$f_{0,n+1} = \max\left\{\omega_i\left(1 - \sum_{(k,\ell)\in C:\ell\neq i}(z_\ell - y_{k\ell})\right) - \theta_i : i \in V(C)\right\}.$$

Let $f_{0,1'} = \theta_{1'}$. For arcs $\{(i',(i+1)') : i \in [1,n-1]\}$, we set the length $f_{i',(i+1)'} = \theta_{(i+1)'}$. For arcs $\{(i',n+1) : i \in [1,n]\}$, we set the length

$$f_{i',n+1} =$$
$$\delta(\{\omega_j\}_{j=1'}^{i'})\left(1 - \sum_{(k,\ell)\in C:\ell\notin\{j\}_{j=1'}^{i'}}(z^*_\ell - y^*_{k\ell})\right) - \sum_{k=1'}^{i'}\left(\frac{\delta(\{\omega_j\}_{j=1'}^{i'})}{\omega_k} + 1\right)\theta_k.$$

This longest path problem depicted in Figure 3.3 can be solved by Dijkstra's algorithm. There exists a violated inequality (3.11) if and only if the longest path is strictly positive and the nodes on this path determine the elements in set U. The sorting process of θ_i takes $O(n \log n)$ time, the evaluation of $f_{0,n+1}$ takes $O(n)$ time, and the longest path on a directed acyclic graph takes $O(n)$ time as there are $n+2$ nodes and $2n+1$ arcs. Since we have to solve this problem for every violated cycle found by Dijkstra algorithm, the separation algorithm runs $O(n^3 \log n)$ time overall. □

Example 4. Consider solving LCIM with $b = 3$ on a graph depicted in Figure 3.2. The initial linear programming relaxation solution without enumerating any cycle elimination

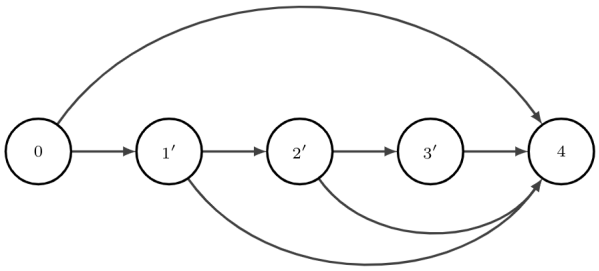


Figure 3.4: A DAG for separating inequality (3.11) in Example 4

constraints is

$$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*) = (0, 4.92, 0.6, 3, 0),$$
$$(z_1^*, z_2^*, z_3^*, z_4^*, z_5^*) = (0.6, 0.6, 0.6, 0.6, 0.6),$$
$$(y_{12}^*, y_{13}^*, y_{14}^*, y_{21}^*, y_{23}^*, y_{25}^*, y_{31}^*, y_{32}^*, y_{41}^*, y_{52}^*) = (0.36, 0, 0, 0.24, 0.6, 0.6, 0.6, 0, 0.6, 0),$$

and the objective function value is 8.52. The violated cycle is $\{(1,2),(2,3),(3,1)\}$ for this solution. Then we obtain $\theta_1 = -0.72$, $\theta_2 = 1.68$ and $\theta_3 = 0$ and sort them in non-decreasing order. Since $\theta_1 < \theta_3 < \theta_2$, the set $\{1', 2', 3'\}$ is corresponding to the original node set $\{1, 3, 2\}$. This leads to a directed acyclic network illustrated in Figure 3.4 with new node set $\mathcal{V} = \{0, 1', 2', 3', 4\}$. The length of $f_{0,1'}$, $f_{1',2'}$ and $f_{2',3'}$ are θ_1, θ_3 and θ_2, respectively. The lengths of the remaining arcs are

$$f_{0,4} = \max\{3(1 - 0.24) - (-0.72), 2 - 1.68, 2\} = 3,$$
$$f_{1',4} = 3(1 - 0.24) - (\frac{3}{3} + 1)(-0.72) = 3.72,$$
$$f_{2',4} = 6(1 - 0.24) - (\frac{6}{3} + 1)(-0.72) = 6.72,$$
$$f_{3',4} = 6 - (\frac{6}{3} + 1)(-0.72) - (\frac{6}{2} + 1)(1.68) = 1.44.$$

The longest path is $0 \to 1' \to 2' \to 4$, which determines $U = \{1, 3\}$ with maximum violation 6. We add the following violated inequality (3.11)

$$2(x_1 + 2y_{21} + 4y_{31} + 6y_{41} - 12z_1) + 3(x_3 + y_{13} + 4y_{23} - 5z_3) \geq 6(1 - z_2 + y_{12})$$

to cut off this fractional solution. We then obtain the new objective function value 10.2 and the solution sets

$$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*) = (0, 6, 2.2, 2, 0),$$
$$(z_1^*, z_2^*, z_3^*, z_4^*, z_5^*) = (0.6, 0.6, 0.6, 0.6, 0.6),$$
$$(y_{12}^*, y_{13}^*, y_{14}^*, y_{21}^*, y_{23}^*, y_{25}^*, y_{31}^*, y_{32}^*, y_{41}^*, y_{52}^*) = (0, 0, 0.2, 0.6, 0.6, 0.6, 0.6, 0, 0.4, 0),$$

which is very close to the true optimal objective function value 11 in this example.

3.4 LCIM under equal influence and 100% adoption

Since the optimal propagation subgraph of LCIM is acyclic, the solution of LCIM on a tree provides a valid lower bound for LCIM on a graph with cycles. Moreover, in practical applications, it is common to assume that both threshold and influence exertion are identical for every node, due to simplicity or lack of accurate estimation. For example, in the unanimous threshold model [13], $h_i = v_i$ for all $i \in V$ is assumed. This diffusion model is normally considered as the most influence resistant one, and it has applications in complex computer network security problems. In addition, the majority threshold model [58] assumes $h_i = \lceil \frac{v_i}{2} \rceil$ for all $i \in V$. Both information diffusion models assume that $d_{ij} = 1$ for all $(i, j) \in E$.

In this special case of LCIM where equal influence is assumed for all $i \in V$ and 100% coverage is required on a tree, we have $d_{ij} = d_i$ for all $i \in V$ and GCEC can be discarded. The LCIM formulation corresponding to equal influence and 100% coverage on a tree graph

is given by

$$\text{(LCIM-TE)} \min_{x,y} \quad \sum_{i \in V} x_i$$

$$\text{s.t. } x_i + d_i \sum_{j \in N(i)} y_{ji} \geq h_i \quad \forall i \in V \tag{3.13a}$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i,j) \in E : i < j \tag{3.13b}$$

$$x \in \mathbb{R}^n_+ \tag{3.13c}$$

$$y \in \mathbb{B}^m. \tag{3.13d}$$

Let $\mathcal{S}$ denote the set of feasible solutions to LCIM-TE on a tree graph and let $\mathcal{R}$ denote the set of feasible solutions to the linear programming relaxation of (3.13a) - (3.13d). Günneç et al. [26] prove that LCIM-TE is polynomial solvable on a tree graph. They propose a compact extended formulation with total unimodular constraints. However, the extended formulation can not be applied to unequal influence weights directly. We give the complete linear description of $\text{conv}\,(\mathcal{S})$ in the original space of variables with additional $O(n)$ constraints and show they are a special case of the continuous cover and continuous packing inequalities by adjusting the influence weights.

Proposition 14. *Let* $\sigma_i = \lceil \frac{h_i}{d_i} \rceil$ *and* $g_i = h_i - (\sigma_i - 1)d_i$ *for all* $i \in V$*, the inequality*

$$x_i + \min\{g_i, d_i\} \sum_{j \in N(i)} y_{ji} \geq g_i \sigma_i \tag{3.14}$$

is facet-defining for $\text{conv}\,(\mathcal{S})$ *if and only if* $g_i < d_i$ *and* $\sigma_i \geq 2$*. Furthermore, the complete linear description of* $\text{conv}\,(\mathcal{S})$ *is given by*

$$\text{conv}\,(\mathcal{S}) = \left\{ (x,y) \in \mathcal{R} : x_i + \min\{g_i, d_i\} \sum_{j \in N(i)} y_{ji} \geq g_i \sigma_i \quad \forall i \in V \right\}.$$

Proof. If $g_i = d_i$, then $h_i = \sigma_i d_i$ and inequality (3.14) coincides with (3.13a). Similarly, if $\sigma_i = 1$, we also have $g_i = h_i = d_i$ and inequality (3.14) is reduced to (3.13a). To prove the sufficiency, we demonstrate that inequality (3.14) is a special case of the continuous cover and continuous packing inequalities. Observe that σ_i is the minimum number that exceeds h_i if multiplied by d_i, which implies that $h_i - (\sigma_i - 1)d_i > 0$. Therefore, σ_i is the cardinality of set L corresponding to the continuous packing inequality with equal influence weights. We thus obtain $\lambda_i = d_i\sigma_i - h_i$ and $g_i = d_i - \lambda_i$. Equivalently,

$$\begin{aligned} g_i &= h_i - (\sigma_i - 1)d_i \\ &= h_i + [|N(i)| - \sigma_i + 1 - |N(i)|]\, d_i \\ &= \pi_i, \end{aligned}$$

hence $|N(i)| - \sigma_i + 1$ is the cardinality of set S in the continuous cover inequality. Following the result of Lemma 1 for the interchangeable relationship between sets S and L, g_i and $g_i\sigma_i$ coincide with the coefficients of the continuous cover and continuous packing inequalities, respectively.

For the second part of this Proposition, we assume $g_i < d_i$ and $g_i\sigma_i < h_i$ holds for all $i \in V$ without loss of generality. Observe that for $i \in V$, the possible values of $x_i = \max\{0, h_i - (\sigma_i - w)d_i\}$ for $w \in [0, \sigma_i]$, where $\sigma_i - w$ is an implicit upper bound of number of activated neighbors for node i, namely, $\sum_{j\in N(i)} y_{ji} \leq \sigma_i - w$. We prove that for any choice of $w \in [0, \sigma_i]$, we must have integral $(\mathbf{x}, \mathbf{y})$ in the following three cases:

Case 1. *Suppose $w = 0$ and $x_i = 0$. Inequality* (3.13a) *is reduced to $\sum_{j\in N(i)} y_{ji} \geq \frac{h_i}{d_i}$, which is dominated by inequality* (3.14) *with $\sum_{j\in N(i)} y_{ji} \geq \sigma_i$. There exist at least σ_i activated neighbors that exert influence toward node i. Moreover, from the implicit upper bound $\sum_{j\in N(i)} y_{ji} \leq \sigma_i$, we have $\sum_{j\in N(i)} y_{ji} = \sigma_i$. Since $x_i = 0$ we must have $y_{ij} = 0$ and $y_{ji} = 1$*

such that $|\{j : j \in N(i)\}| = \sigma_i$ *due to constraints* (3.13b).

Case 2. *Suppose* $w = \sigma_i$ *and* $x_i = h_i$. *Inequality* (3.13a) *becomes* $\sum_{j \in N(i)} y_{ji} \geq 0$, *which dominates inequality* (3.14) *with* $\sum_{j \in N(i)} y_{ji} \geq \sigma_i - \frac{h_i}{g_i}$ *as the right hand side here is strictly negative. Similar to Case 1, we must have* $y_{ji} = 0$ *and* $y_{ij} = 1$ *for all* $j \in N(i)$ *due to the implicit upper bound* $\sum_{j \in N(i)} y_{ji} \leq 0$ *and constraints* (3.13b).

Case 3. *Suppose* $w \in [1, \sigma_i - 1]$. *First, let* $w = 1$, *then* $x_i = g_i = h_i - (\sigma_i - 1)d_i$. *We have* $\sum_{j \in N(i)} y_{ji} \geq \sigma_i - 1$ *in both inequality* (3.13a) *and inequality* (3.14). *Following Case 1 and Case 2, we have* $\sum_{j \in N(i)} y_{ji} = \sigma_i - 1$ *with* $y_{ji} = 1$ *and* $y_{ij} = 0$ *for some* $j \in N(i)$ *such that* $|\{j : j \in N(i)\}| = \sigma_i - 1$. *Next, let* $w = 2$ *and* $\sum_{j \in N(i)} y_{ji} \geq \sigma_i - 2$ *holds in inequality* (3.13a). *While in inequality* (3.14),

$$\begin{aligned} \sum_{j \in N(i)} y_{ji} &\geq \frac{g_i\sigma_i - h_i + d_i\sigma_i - 2d_i}{g_i} \\ &\geq \sigma_i - 1 - \frac{d_i}{g_i}. \end{aligned}$$

Since we assume $g_i < d_i$, *inequality* (3.13a) *dominates inequality* (3.14). *By mathematical induction, for* $w \in [1, \sigma_i - 1]$, *we conclude that inequality* (3.13a) $\sum_{j \in N(i)} y_{ji} \geq w$ *always dominates inequality* (3.14). *Furthermore, we must have* $\sum_{j \in N(i)} y_{ji} = w$ *from the implicit upper bound and the value of* y_{ij} *and* y_{ji} *are either 0 or 1 following Case 1 and Case 2. We have now demonstrated that inequality* (3.14) *is facet-defining and* $(\mathbf{x}, \mathbf{y})$ *are integral for any choice of* $w \in [0, \sigma_i]$, *thus the proof is completed.*

□

We close this section by noting that the (U, C) inequalities (3.11) and the separation algorithm in Section 3.3.3 can be directly applied to LCIM-TE on a graph with cycles.

Proposition 15. *The (U, C) inequality for equal influence weights of a cycle is given by*

$$\sum_{i \in U} \gamma_i \left(x_i + \alpha_i \sum_{j \in N(i)} y_{ji} - \beta_i \right) \geq \delta(U) \left(1 - |V(C)| + |U| + \sum_{(k,\ell) \in C : \ell \neq i} y_{k\ell} \right) \tag{3.15}$$

for conv $(\mathcal{S})$.

Proof. The result is deduced from Proposition 11 by fixing z_i to 1 for all $i \in V$ and substituting the coefficients accordingly. The definition of γ_i and $\delta(U)$ follows Proposition 11. Let $\omega_i = h_i - \beta_i + |\{j \in N(i) : j \notin V(C)\}|(\alpha_i - d_i)$ for all $i \in V(C)$. The right hand side of the inequality is equivalent to fix $z_i = 1$ for $i \notin U$. Finally, let $\alpha_i = \min\{g_i, d_i\}$ and $\beta_i = g_i \sigma_i$ as in inequality (3.14). □

3.5 Computational Experiments

In this section, we give a detailed description of the data generation and algorithm settings. We test the effectiveness of a delayed cut generation algorithm that incorporates the proposed valid inequalities in solving LCIM under different conditions. All the experiments were conducted on a single thread of a Windows 10 Enterprise server with Intel(R) Core i7-4770 CPU at 3.40 GHz x-64 based processor and 8GB of RAM using Python 3.8 and Gurobi 9.1.2 with default settings as the optimization solver. A 3600 seconds time limit was imposed for each experiment.

3.5.1 Data generation and algorithm settings

We follow the exact data generation scheme in [18], except for the fact that we generate bidirectional arcs between every two nodes. The small-world network topology for each instance is generated based on `watts_strogatz_graph` function in the `NetworkX` package of Python. The instances have the following properties: size of node set $n \in \{50, 75, 100\}$, average node

degree $v \in \{4, 8, 12, 16\}$, rewiring probability $q \in \{0.1, 0.3\}$ and we set penetration rate $a \in \{0.1, 0.25, 0.5, 0.75, 1\}$. Influence weight d_{ij} for all $(i, j) \in E$ are generated from discrete uniform distribution between 1 and 10. Let $\Delta_i = \sum_{j \in N(i)} d_{ji}$ and Υ_i be a random variable follows normal distribution $\mathcal{N}(0.7\Delta_i, \Delta_i/v_i)$ for all $i \in V$. We set $h_i = \lceil \max\{1, \min\{\Upsilon_i, \Delta_i\}\} \rceil$. For each setting, we generate three instances and report the average.

The effectiveness of two delayed cut generation algorithms and one alternative reformulation are compared in our study:

1. **DEF**: formulation LCIM given by (3.1a) - (3.1c),
2. **CB**: formulation LCIM with cut-and-branch enhancement, and
3. **LN**: layered-network formulation.

To implement the delayed cut generation, the GCEC (3.1d) is separated via lazy constraint callback only for integer solutions for **DEF** and **CB**. Grötschel et al. [24] give a shortest path algorithm for separating (3.1d) and we utilize the existing shortest path function `dijkstra_path` in the `NetworkX` package to perform such task. For the cut-and-branch enhancement in algorithm **CB**, we add the proposed inequalities via user-cut callback at the root node to tighten the linear programming relaxation of formulation LCIM.

We use the layered-network formulation proposed by [36] to replace (3.1d) in algorithm **LN** as this formulation gives a directed acyclic graph with the additional layer assignment variables l_i for all $i \in V$. We are interested in testing whether this cycle-free formulation is beneficial to solve LCIM compared with GCEC (3.1d). In addition, this formulation allows us to keep the mixed 0-1 knapsack substructure so the valid inequalities can be applied to it directly. In our preliminary experiments, we observe that this formulation does not produce better optimality gap for $a \in \{0.1, 0.25, 0.5, 0.75\}$ by relaxing constraints (3.16a) compared with **DEF** and **CB** (average optimality gap $> 90\%$), hence, we only report

the computation for $a = 1$ ($b = n$). The layered-network formulation used in algorithm **LN** is given by

$$\min \left\{ \sum_{i \in V} x_i : (x, y, z) \text{ satisfies (3.1a), (3.1c), (3.16a) } - \text{ (3.16c)} \right\},$$

where

$$y_{ij} + y_{ji} = 1 \quad \forall (i,j) \in E \tag{3.16a}$$

$$y_{ji} - (n-1) y_{ij} \leq l_j - l_i \quad \forall (i,j) \in E \tag{3.16b}$$

$$1 \leq l_i \leq n \quad \forall i \in V. \tag{3.16c}$$

3.5.2 Analysis of results

We summarize our computational results in TABLE 3.3, 3.4 and 3.5 under various settings of (n, m, v, q, a). We report the average number of branch-and-cut tree nodes explored in the column **Nodes**. The column **Cuts** shows the average number of Gurobi cuts added during the optimization process. In column **Time[Gap]*** we report the average solution time (in seconds) of the instances that are solved to optimality, and the average of the optimality gap of the instances that are not solved to optimality when reaching time limit (in brackets). Each asterisk sign indicates an unsolved instance and the gap is calculated by $100 \times (\texttt{ub} - \texttt{lb})/\texttt{lb}$ where `ub` and `lb` are the best integer feasible solution obtained and best lower bound generated by the algorithm within time limit, respectively.

We observe that the major factor that contributes to the unsolved instances with positive optimality gap is the average node degree rather than the number of nodes. This observation can be justified by comparing the instances $(n, v, m) = (100, 4, 400)$ and $(n, v, m) = (50, 8, 400)$, as the former is easier to solve than the later. A similar observation is also established in [18] in their set covering formulation using the price-cut-and-branch algo-

Table 3.3: Computational performance comparing MIP nodes, cuts, time and unsolved instances on network with $n = 50$.

$n = 50$			Nodes			Cuts			Time[Gap]*		
$v - m$	q	a	DEF	CB	LN	DEF	CB	LN	DEF	CB	LN
		0.1	18	12		12	12		0.30	0.31	
		0.25	121	53		25	15		0.45	0.40	
4-200	0.1	0.5	492	254		52	21		0.84	0.65	
		0.75	1562	1680		76	45		2.53	2.74	
		1	921	636	1041	70	27	233	1.15	0.78	17.84
		0.1	1	5		5	4		0.29	0.20	
		0.25	1	1		13	3		0.33	0.27	
4-200	0.3	0.5	33	9		28	10		0.41	0.39	
		0.75	129	99		45	18		0.61	0.56	
		1	207	1	140	47	16	83	0.57	0.42	5.02
		0.1	85	115		18	15		1.02	1.71	
		0.25	429	392		14	14		2.41	2.85	
8-400	0.1	0.5	6077	8831		57	65		19.08	27.31	
		0.75	457292	196845		189	169		1367.06[7.31]*	682.35	
		1	838587	658888	69506	339	341	2491	[8.22]***	[4.70]***	[8.56]***
		0.1	104	85		19	14		0.96	1.49	
		0.25	375	497		17	18		2.50	3.81	
8-400	0.3	0.5	3009	3592		35	40		15.88	31.34	
		0.75	323777	185520		148	170		2028.51[5.11]*	1486.88	
		1	418483	404013	57946	297	265	2372	[10.47]***	[7.13]***	[12.08]***

rithm. Despite the average number of nodes and cuts are the greatest in **LN** for $n = 50$, there exists no clear domination relationship in columns **Nodes** and **Cuts** between **DEF**, **CB** and **LN** for $n = 75$ or 100. Moreover, the average number of cuts added is not very large, which indicates that Gurobi cuts do not complement the optimization process. For the unsolved instances in Table 3.3 and 3.4, **CB** outperforms **DEF** and **LN** except for instances $(n, v, m, q, a) = (75, 12, 900, 0.1, 0.5)$ and $(75, 12, 900, 0.3, 0.5)$. In Table 3.5 where n=100, **CB** still outperforms **DEF** and **LN** for $v \in \{4, 8, 12\}$ except for one instance $(100, 12, 1200, 0.3, 0.5)$, where the optimality gap difference is 0.92%. The common setting $a = 0.5$ shared in these exceptions suggests that the symmetry created by the cardinality constraint requires additional improvement. We also notice that **LN** produces better opti-

Table 3.4: Computational performance comparing MIP nodes, cuts, time and unsolved instances on network with $n = 75$.

$n = 75$			Nodes			Cuts			Time[Gap]*		
$v - m$	q	a	DEF	CB	LN	DEF	CB	LN	DEF	CB	LN
4-300	0.1	0.1	91	17		11	8		0.57	0.31	
		0.25	833	742		46	22		1.87	1.35	
		0.5	1094	1543		82	48		2.11	3.17	
		0.75	3378	1880		119	69		5.92	4.17	
		1	1753	1384	1252	130	69	366	2.08	2.15	18.39
4-300	0.3	0.1	10	12		17	9		0.64	0.70	
		0.25	309	76		34	13		0.85	0.71	
		0.5	805	151		78	21		1.78	1.24	
		0.75	1537	604		80	37		3.50	2.66	
		1	976	965	1067	107	53	445	2.44	2.12	16.63
8-600	0.1	0.1	229	195		13	10		2.73	3.45	
		0.25	3178	912		26	20		18.82	7.81	
		0.5	216230	212500		192	78		227.75[4.77]*	857.06	
		0.75	564387	348760		248	169		[6.15]***	508.18[3.36]**	
		1	360594	379776	50255	368	324	2923	[9.50]***	[5.71]***	[9.64]***
8-600	0.3	0.1	214	127		13	5		2.10	2.78	
		0.25	1078	1224		19	15		10.54	10.27	
		0.5	9948	3776		68	75		129.29	113.47	
		0.75	161508	223027		184	194		[5.00]***	3489.72[1.99]**	
		1	191592	167297	41342	266	356	2701	[9.48]***	[6.38]***	[10.30]***
12-900	0.1	0.1	812	635		9	10		13.93	18.00	
		0.25	1803	2376		18	13		53.28	62.91	
		0.5	369964	229309		36	63		2803.78[2.11]*	[2.96]***	
		0.75	18614	31354		111	144		[13.68]***	[10.28]***	
		1	48188	33974	26375	228	230	3317	[16.49]***	[16.18]***	[17.85]***
12-900	0.3	0.1	1163	1275		10	2		15.08	30.70	
		0.25	3757	5019		42	19		115.19	224.71	
		0.5	27217	22714		35	48		1363.66[2.83]*	1827.81[9.39]*	
		0.75	11042	12625		58	120		[24.75]***	[15.69]***	
		1	20954	25899	23253	199	283	3071	[22.71]***	[17.21]***	[26.88]***

mality gap than **DEF** for instances $(100, 8, 800, 0.1, 1)$ and $(100, 12, 1200, 0.1, 1)$. However, **LN** suffers from slow improvement of both upper and lower bound and results in larger ending gap in general in our experiments. We begin to see the degraded performance of algorithm **CB** for instances with $v = 16$. A large number of valid inequalities added to the root node could be a possible reason that decelerates the optimization process. Five out of ten results generated by **CB** under this category are no better than **DEF**. Nevertheless,

Table 3.5: Computational performance comparing MIP nodes, cuts, time and unsolved instances on network with $n = 100$.

$n = 100$			Nodes			Cuts			Time[Gap]*		
$v - m$	q	a	DEF	CB	LN	DEF	CB	LN	DEF	CB	LN
		0.1	28	1		11	4		0.95	0.85	
		0.25	148	118		50	12		1.24	1.09	
4-400	0.1	0.5	1554	998		80	33		3.77	2.49	
		0.75	1705	834		128	61		6.58	3.62	
		1	669	1600	670	120	63	318	3.09	2.67	13.01
		0.1	15	11		15	3		0.53	0.36	
		0.25	83	123		22	7		1.27	1.07	
4-400	0.3	0.5	365	391		92	26		2.40	1.60	
		0.75	2045	697		117	53		8.55	2.92	
		1	904	213	994	124	42	383	3.84	1.52	18.65
		0.1	511	295		17	7		7.11	7.28	
		0.25	16138	9611		16	59		96.09	88.48	
8-800	0.1	0.5	364834	278716		297	157		1499.18[4.20]**	1247.35[2.37]*	
		0.75	175014	231825		282	306		[9.81]***	[7.10]***	
		1	121123	154716	46686	374	439	2988	[13.84]***	[8.39]***	[13.02]***
		0.1	434	345		16	12		5.74	7.27	
		0.25	5682	4509		19	21		56.08	72.36	
8-800	0.3	0.5	65328	25567		91	129		735.63[1.79]*	685.49	
		0.75	13119	37109		190	233		[14.13]***	[6.43]***	
		1	40550	68921	35379	261	407	2849	[15.19]***	[9.39]***	[17.02]***
		0.1	1866	1249		6	4		44.77	83.37	
		0.25	6425	6356		29	48		296.98	372.08	
12-1200	0.1	0.5	40341	41264		41	52		[11.17]***	[9.17]***	
		0.75	8292	7915		143	173		[19.69]***	[16.40]***	
		1	15506	16755	20873	187	388	4447	[29.78]***	[18.11]***	[20.74]***
		0.1	3984	2870		0	9		39.06	145.17	
		0.25	55111	33660		14	65		1187.31	2191.00	
12-1200	0.3	0.5	15457	11751		44	51		[21.00]***	[21.92]***	
		0.75	4426	5168		102	128		[30.63]***	[24.46]***	
		1	8704	10036	20128	174	241	3688	[21.11]***	[19.05]***	[21.31]***
		0.1	4585	5403		14	11		193.29	474.52	
		0.25	9590	11593		26	28		1244.07	2010.94	
16-1600	0.1	0.5	5384	4268		9	14		[22.36]***	[22.42]***	
		0.75	3360	3591		68	92		[32.71]***	[30.71]***	
		1	8464	6772	11907	148	158	3443	[22.47]***	[23.43]***	[24.10]***
		0.1	11982	13063		5	20		286.70	852.37	
		0.25	51785	23015		19	33		[19.46]***	[22.45]***	
16-1600	0.3	0.5	5271	4520		17	18		[34.36]***	[35.58]***	
		0.75	2636	2724		85	89		[42.24]***	[44.46]***	
		1	4441	4485	11802	78	91	3408	[25.78]***	[24.78]***	[25.79]***

the average optimality gap produced by **CB** for these five instances are 1.49% higher than **DEF**. To sum up, although the linear programming relaxation of the arc-based formulation

is very weak with zero objective function value in most cases, our proposed valid inequalities significantly improve the strength of the lower bound and effectively reduce or close the optimality gap.

3.6 Conclusion

We study the least cost influence maximization problem in social networks where the influence propagation behavior among users is captured by the deterministic linear threshold model. A typical application of this problem is to obtain an estimation of the partial incentives given to early product adopters in viral marketing while achieving a desired coverage rate by the end of information spreading. We focus on the case where influence weights exerted from peers are heterogeneous and derive several classes of valid inequalities from the hidden mixed 0-1 knapsack substructure in the mixed-integer programming formulation. Despite the fact that the set of feasible solutions is hard to convexify due to the knapsack constraints and the linear programming relaxation being very weak, our computational experiments show that the delayed cut generation algorithm exploiting these inequalities can effectively reduce the optimality gap. For the case with equal influence weights and 100% adoption on a tree, we characterize the complete linear description of the convex hull in it in the natural space of incentive, arc propagation and activation variables. The convex hull of the LCIM with equal influence weights and arbitrary adoption on a tree is still an open question and requires further investigation. We observe that the bottleneck of computational improvements are mostly in the instances where 100% adoption is not required. A promising future research direction is to identify more explicit valid inequalities from the intersections of the cardinality constraint that controls the penetration rate with the rest of the formulation.

CHAPTER 4: CONTRIBUTIONS AND FUTURE WORK

In this dissertation, we presented mathematical optimization models that describe different types of cascading processes in interdependent infrastructure networks and social networks with discrete decisions. We studied the polyhedral structure and propose exact solution algorithms based on mixed-integer programming techniques, including dynamic programming, cutting plane, strong formulation and delayed cut generation algorithm. In this concluding chapter, we summarize our contributions and discuss some of the promising areas of future research.

In Chapter 2, we proposed a novel mixed-integer optimization model for a two-layered interdependent networks mitigation and restoration problem. The dynamic of the cascading failure processes and the corresponding time and logical conditions are captured explicitly in the model. Our model provides a guidance for the trade-off decision between mitigation investment in advance and sequential post-disaster recovery actions, while minimizing the damage caused by cascading failure. We propose valid inequalities and strong formulations for the constraints of cascading process and capacity degradation, respectively. The computational results outperform the state-of-the-art commercial optimization solver in its default settings. It should be noted, in general, that mixed integer optimization problems often present significant computational challenges, which is also the case in the considered setup. The results from these computationally challenging instances indicate that exploring effective decomposition methods (for instance, Lagrangian relaxation) and other techniques for improving the computational performance is a promising future research direction.

In addition, from the perspective of modeling assumptions, other optimization models can be formulated and solved in related settings. Our model considers mitigation and restoration with respect to nodes only, since the failure propagation mechanism in our model is based on failed nodes between network layers. Another possible direction would be to con-

sider restoration and hardening of the interdependent links, which would be suitable under the assumptions that cascading failures propagate via link failures instead of (or in addition to) node failures. Such models would require introducing a different set of variables and constraints in order to properly model the propagation of cascading failures in these settings, and their properties may differ from the ones derived here for our model. Therefore, it would be interesting to formulate and study these optimization problems in the future work.

In Chapter 3, we study the least cost influence maximization problem in social networks with the static cascading influence propagation processes. The mixed-integer optimization formulation is challenging to solve as it contains a mixed 0-1 knapsack polyhedron, cardinality constraint and an exponential number of cycle elimination constraints. We derive strong valid inequalities from the intersections of the mixed 0-1 knapsack polyhedron and cycle elimination constraints and utilize them in a delayed cut generation algorithm. Our computational results show that they are very effective in solving the problem. While we present a complete linear description of the convex hull for the problem with equal influence weights and 100 % on a tree, the convex hull formulation for such case on arbitrary graphs is still an open question and requires further investigation. When 100% adoption is not required, the cardinality constraint also creates a significant computational burden. It would be interesting to find more explicit valid inequalities from the intersections of the cardlinaity constraint with the rest of the formulation.

An alternative way to get rid of the exponential number of cycle elimination constraint is to formulate the problem with time index. Such formulation can also be extended into multistage setting with uncertain influence weights and threshold directly. In our preliminary investigation, we observe the continuous cover and continuous packing inequalities are valid for the time index formulation. It would be interesting to further investigate other valid inequalities and develop a scalable decomposition algorithm to solve the multistage stochastic counterpart of this problem.

LIST OF REFERENCES

[1] Eyal Ackerman, Oren Ben-Zwi, and Guy Wolfovitz. Combinatorial model and bounds for target set selection. *Theoretical Computer Science*, 411(44-46):4017–4022, 2010.

[2] Negin Enayaty Ahangar, Kelly M Sullivan, and Sarah G Nurre. Modeling interdependencies in infrastructure systems using multi-layered network flows. *Computers & Operations Research*, page 104883, 2020.

[3] Yasser Almoghathawi and Kash Barker. Component importance measures for interdependent infrastructure network resilience. *Computers & Industrial Engineering*, 133:153–164, 2019.

[4] Australian Energy Market Operator. Black system south australia 28 september 2016. *Report of the Australian Energy Market Operator Limited (AEMO)*, 2017.

[5] Mehdi Azaouzi, Wassim Mnasri, and Lotfi Ben Romdhane. New trends in influence maximization models. *Computer Science Review*, 40:100393, 2021.

[6] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihar. A survey on influence maximization in a social network. *Knowledge and Information Systems*, 62(9):3417–3455, 2020.

[7] Nail Orkun Baycik, Thomas C Sharkey, and Chase E Rainwater. Interdicting layered physical and information flow networks. *IISE Transactions*, 50(4):316–331, 2018.

[8] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025, 2010.

[9] Massimo Cavallaro, Domenico Asprone, Vito Latora, Gaetano Manfredi, and Vincenzo Nicosia. Assessment of urban ecosystem resilience through hybrid social–physical complex networks. *Computer-Aided Civil and Infrastructure Engineering*, 29(8):608–625, 2014.

[10] Melih Çelik. Network restoration and recovery in humanitarian operations: framework, literature review, and research directions. *Surveys in Operations Research and Management Science*, 21(2):47–61, 2016.

[11] Cheng-Lung Chen, Eduardo L Pasiliao, and Vladimir Boginski. A cutting plane method for least cost influence maximization. In *International Conference on Computational Data and Social Networks*, pages 499–511. Springer, 2020.

[12] Cheng-Lung Chen, Qipeng Zheng, Alexander Veremyev, Eduardo L Pasiliao, and Vladimir Boginski. Failure mitigation and restoration in interdependent networks via mixed-integer optimization. *IEEE Transactions on Network Science and Engineering*, 2020.

[13] Ning Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009.

[14] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[15] Veronica Dal Sasso, Luigi De Giovanni, and Martine Labbé. Strengthened formulations and valid inequalities for single delay management in public transportation. *Transportation Science*, 53(5):1271–1286, 2019.

[16] Leonardo Dueñas-Osorio and Srivishnu Mohan Vemuru. Cascading failures in complex infrastructure systems. *Structural safety*, 31(2):157–167, 2009.

[17] EC Eakeley. 1996 system disturbances. *North America Electric Reliability Council*, 2002.

[18] Matteo Fischetti, Michael Kahr, Markus Leitner, Michele Monaci, and Mario Ruthmair. Least cost influence propagation in (social) networks. *Mathematical Programming*, 170(1):293–325, 2018.

[19] Jianxi Gao, Sergey V Buldyrev, H Eugene Stanley, and Shlomo Havlin. Networks formed from interdependent networks. *Nature physics*, 8(1):40, 2012.

[20] Colin P Gillen, Alexander Veremyev, Oleg A Prokopyev, and Eduardo L Pasiliao. Critical arcs detection in influence networks. *Networks*, 71(4):412–431, 2018.

[21] Collin P. Gillen, Alexander Veremyev, Oleg A. Prokopyev, and Eduardo L. Pasiliao. Fortification against cascade propagation under uncertainty. Technical report, University of Pittsburgh, 2018.

[22] Andrés D González, Leonardo Dueñas-Osorio, Mauricio Sánchez-Silva, and Andrés L Medaglia. The interdependent network design problem for optimal infrastructure system restoration. *Computer-Aided Civil and Infrastructure Engineering*, 31(5):334–350, 2016.

[23] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

[24] Martin Grötschel, Michael Jünger, and Gerhard Reinelt. On the acyclic subgraph polytope. *Mathematical Programming*, 33(1):28–42, 1985.

[25] Dilek Günneç, S Raghavan, and Rui Zhang. A branch-and-cut approach for the least cost influence problem on social networks. *Networks*, 76(1):84–105, 2020.

[26] Dilek Günneç, Subramanian Raghavan, and Rui Zhang. Least-cost influence maximization on social networks. *INFORMS Journal on Computing*, 32(2):289–302, 2020.

[27] Furkan Gursoy and Dilek Günneç. Influence maximization in social networks under deterministic linear threshold model. *Knowledge-Based Systems*, 161:111–123, 2018.

[28] Xuqing Huang, Jianxi Gao, Sergey V Buldyrev, Shlomo Havlin, and H Eugene Stanley. Robustness of interdependent networks under targeted attack. *Physical Review E*, 83(6):065101, 2011.

[29] Brian Keller and Güzin Bayraksan. Scheduling jobs sharing multiple resources under uncertainty: A stochastic programming approach. *Iie Transactions*, 42(1):16–30, 2009.

[30] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[31] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[32] Robert M. Lee, Michael J. Assante, and Tim Conway. Analysis of the cyber attack on the ukrainian power grid. *SANS Industrial Control Systems*, 388, 2016.

[33] Earl E Lee II, John E Mitchell, and William 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.

[34] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1852–1872, 2018.

[35] Marko Loparic, Hugues Marchand, and Laurence A Wolsey. Dynamic knapsack sets and capacitated lot-sizing. *Mathematical Programming*, 95(1):53–69, 2003.

[36] Hasan Manzour, Simge Küçükyavuz, Hao-Hsiang Wu, and Ali Shojaie. Integer programming for learning directed acyclic graphs from continuous data. *Informs Journal on Optimization*, 3(1):46–73, 2021.

[37] Hugues Marchand and Laurence A Wolsey. The 0-1 knapsack problem with a single continuous variable. *Mathematical Programming*, 85(1):15–33, 1999.

[38] Garth P McCormick. *Nonlinear programming; theory, algorithms, and applications*. John Wiley & Sons: New York, NY, 1983.

[39] Andrew J Miller, George L Nemhauser, and Martin WP Savelsbergh. On the capacitated lot-sizing and continuous 0–1 knapsack polyhedra. *European Journal of Operational Research*, 125(2):298–315, 2000.

[40] Babak Moazzez and Hossein Soltani. Integer programming approach to static monopolies in graphs. *Journal of Combinatorial Optimization*, 35(4):1009–1041, 2018.

[41] Babak Moazzez and Hossein Soltani. Facets of the dynamic monopoly polytope: Linear ordering formulation. *Discrete Optimization*, 40:100641, 2021.

[42] A Muir and J Lopatto. Final report on the august 14, 2003 blackout in the united states and canada. *US–Canada Power System Outage Task Force, Canada*, 2004.

[43] Giacomo Nannicini, Giorgio Sartor, Emiliano Traversi, and Roberto Wolfler Calvo. An exact algorithm for robust influence maximization. *Mathematical Programming*, 183(1):419–453, 2020.

[44] Amar K Narisetty, Jean-Philippe P Richard, and George L Nemhauser. Lifted tableaux inequalities for 0–1 mixed-integer programs: A computational study. *INFORMS Journal on Computing*, 23(3):416–424, 2011.

[45] Dung T Nguyen, Yilin Shen, and My T Thai. Detecting critical nodes in interdependent power networks for vulnerability assessment. *IEEE Transactions on Smart Grid*, 4(1):151–159, 2013.

[46] Huy Nguyen and Thomas C Sharkey. A computational approach to determine damage in infrastructure networks from outage reports. *Optimization Letters*, 11(4):753–770, 2017.

[47] Marzieh Parandehgheibi and Eytan Modiano. Robustness of interdependent networks: The case of communication networks and the power grid. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 2164–2169. IEEE, 2013.

[48] Marzieh Parandehgheibi, Eytan Modiano, and David Hay. Mitigating cascading failures in interdependent power grids and communication networks. In *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, pages 242–247. IEEE, 2014.

[49] Sancheng Peng, Yongmei Zhou, Lihong Cao, Shui Yu, Jianwei Niu, and Weijia Jia. Influence analysis in social networks: A survey. *Journal of Network and Computer Applications*, 106:17–32, 2018.

[50] S Raghavan and Rui Zhang. Weighted target set selection on trees and cycles. *Networks*, 77(4):587–609, 2021.

[51] J-PP Richard, Ismael R de Farias Jr, and George L Nemhauser. Lifted inequalities for 0-1 mixed integer programming: Basic theory and algorithms. *Mathematical programming*, 98(1):89–113, 2003.

[52] J-PP Richard, Ismael R de Farias Jr, and George L Nemhauser. Lifted inequalities

for 0-1 mixed integer programming: Superlinear lifting. *Mathematical programming*, 98(1):115–143, 2003.

[53] Thomas C Sharkey, Burak Cavdaroglu, Huy Nguyen, Jonathan Holman, John E Mitchell, and William A Wallace. Interdependent network restoration: On the value of information-sharing. *European Journal of Operational Research*, 244(1):309–321, 2015.

[54] Siqian Shen. Optimizing designs and operations of a single network or multiple interdependent infrastructures under stochastic arc disruption. *Computers & Operations Research*, 40(11):2677–2688, 2013.

[55] Hossein Soltani and Babak Moazzez. A polyhedral study of dynamic monopolies. *Annals of Operations Research*, 279(1):71–87, 2019.

[56] Kai Sun, Yunhe Hou, Wei Sun, and Junjian Qi. *Power System Control Under Cascading Failures: Understanding, Mitigation, and System Restoration*. John Wiley & Sons, 2019.

[57] UCTE Investigation Committee. Interim report of the investigation committee on the 28 september 2003 blackout in italy. *UCTE Report, October 27*, 27, 2003.

[58] Thomas W Valente. Social network thresholds in the diffusion of innovations. *Social networks*, 18(1):69–89, 1996.

[59] Alexander Veremyev, Alexey Sorokin, Vladimir Boginski, and Eduardo L Pasiliao. Minimum vertex cover problem for coupled interdependent networks with cascading failures. *European Journal of Operational Research*, 232(3):499–511, 2014.

[60] Hao-Hsiang Wu and Simge Küçükyavuz. A two-stage stochastic programming approach for influence maximization in social networks. *Computational Optimization and Applications*, 69(3):563–595, 2018.

[61] Yanlu Zhang, Naiding Yang, and Upmanu Lall. Modeling and simulation of the vulnerability of interdependent power-water infrastructure networks to cascading failures. *Journal of Systems Science and Systems Engineering*, 25(1):102–118, 2016.

[62] Enrico Zio and Giovanni Sansavini. Modeling interdependent network systems for identifying cascade-safe operating margins. *IEEE Transactions on Reliability*, 60(1):94–101, 2011.