DETERMINING THE MOST VULNERABLE COMPONENTS IN TRANSPORTATION NETWORKS

by

Aylin Öncü

B.S., Industrial Engineering, Istanbul Technical University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Indusrial Engineering

Boğaziçi University

2019

ACKNOWLEDGEMENTS

I would like to express my special thanks to my thesis supervisor Prof. Necati Aras and co-supervisor Assist. Prof. Hande Küçükaydın for their continuous guidance, constructive suggestions, invaluable support and understanding throughout this study. It has been an intense learning process that has developed me a lot and I would like to thank them for that especially.

I would like to thank Prof. İ. Kuban Altınel, Prof. Z. Caner Taşkın and Assoc. Prof. Hakan Akyüz for taking time to examine my thesis and taking part in my thesis committee.

I would like to extend my thanks to Gökalp Erbeyoğlu, Bahadır Pamuk and Çiğdem Karademir for their help in coding.

I also wish to thank Yasemin Aylin, Özlem and Ruçhan for their valuable friendships. I am very happy to meet them in my master study.

Furthermore, I gratefully acknowledge the support of this study by TÜBİTAK through project 117M593.

Last but not least, I am very grateful to my precious family and Tunç for their endless encouragement and support.

ABSTRACT

DETERMINING THE MOST VULNERABLE COMPONENTS IN TRANSPORTATION NETWORKS

Transportation network, electricity distribution network, supply chain network, telecommunication network are the main critical infrastructure networks used to provide products and services to customers. Man-made intentional attacks may cause serious problems in these critical infrastructure networks, especially in large cities. For example, terrorist activities are man-made intentional attacks. As a result of terrorist attacks, a metro station may become unusable for transportation, an airport may be unable to provide service, a bridge may be closed to vehicle traffic, and the communication may be prevented due to the damaged telephone lines. When such attacks are carried out by an intelligent agent, disruptions may be even bigger.

In this study, it is aimed to determine the components that are affected most by disruptions in a transportation network. These most affected components are the most vulnerable components in the network. Therefore, a two-level mathematical model has been established that can periodically identify the most vulnerable components. In the developed model, the virtual attacker is the leader who wants to cause the most disruption in the transportation network (minimizing the amount of flow on the network), and the system operator is the follower who wants to operate the transportation network optimally after the disruption (to maximize the amount of flow on the network). The components that can be attacked in the model are stations and it is considered that a station is completely shut down after an interdiction. A tabu search heuristic is used as the solution method of the proposed model. The developed tabu search method has been tested on randomly generated different sized networks and its performance is assessed in comparison with a complete enumeration and a greedy approach.

ÖZET

ULAŞIM AĞLARINDA EN KIRILGAN BİLEŞENLERİN BELİRLENMESİ

Ulaşım ağı, elektrik dağıtım ağı, tedarik zinciri ağı, telekomünikasyon ağı müşterilere ürün ve hizmet sağlamada kullanılan başlıca kritik altyapı ağlarıdır. İnsan kaynaklı kasıtlı yapılan saldırılar, özellikle büyük şehirlerdeki bu kritik altyapı ağlarında ciddi problemlere sebep olabilir. Bunlara örnek olarak, terörist faaliyetler sonucunda bir metro istasyonun kullanılamaz hale gelerek ulaşımın engellenmesi, bir havaalanının hizmete kapanması, bir köprünün araç trafiğine kapatılması ve telefon hatlarının zarar görerek iletişimin engellenmesi verilebilir. Bu tarz saldırılar akıllı bir etmen tarafından gerçekleştirildiğinde ise hasarlar daha da büyük olabilmektedir.

Bu çalışmada, bir ulaşım ağındaki aksamalardan en çok etkilenen bileşenlerin saptanması amaçlanmaktadır. En çok etkilenen bileşenler ağdaki en kırılgan bileşenleri belirlemektedir. Bu nedenle, periyodik olarak ağdaki en kırılgan bileşenleri belirleyebilecek iki düzeyli bir matematiksel model kurulmuştur. Önerilen modelde ulaşım ağında en fazla aksamaya yol açmak (ağ üzerindeki akış miktarını enküçüklemek) isteyen sanal saldırgan öncü, saldırı sonrası ulaşım ağını eniyi şekilde işletmek (ağ üzerindeki akış miktarını enbüyüklemek) isteyen sistem operatörü ise takipçi konumundadır. Modeldeki ulaşım ağının saldırılabilen bileşenleri istasyonlar olarak ele alınmakta ve bir saldırı sonrasında bir istasyonun tamamen hizmete kapatıldığı düşünülmektedir. Geliştirilen model için çözüm yöntemi olarak bir tabu arama sezgiseli kullanılmıştır. Önerilen tabu arama yöntemi, rastgele oluşturulmuş farklı büyüklükteki ağlar üzerinde test edilmiş ve tabu arama yönteminin performansı, geliştirilen iki farklı çözüm yaklaşımı ile karşılaştırılarak değerlendirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_{ij}	1 if there is node i to node j in the first shortest path solution; 0 otherwise
B	set of source nodes in the network
c_{curr}	total cost of the current solution
c_{neigh}	total cost of a neighbor solution
d_{it}	capacity of node i in period t
f_{it}	amount of flow demand at node i in period t
$f_{i,t+l}$	amount of flow demand at node i in a recovery period
f_{wt}	flow demand between node pair w in period t
g_{ij}	distance from node i to node j
h	tabu tenure
h'	current size of the tabu list
K_w	set of admissible paths between node pair w
L	set of recovery periods, i.e. $L = \{0, ..., \bar{L}\}$
m	number of nodes
N	set of nodes in the network
o_{it}	amount of resource required for complete interdiction of node i in period t
P_{it}	1 if node i is operating in period t; 0 otherwise
r	budget of attacker
S	set of destination nodes in the network
T	set of periods
U_i	order of visiting node i in the second shortest path
W	set of node pairs which have passenger flow demand between source node b and destination node s
X_{it}	1 if node i is interdicted in period t; 0 otherwise
Y_{ij}	1 if the arc between node i and node j is in the selected path; 0 otherwise
$z_{\mathrm{TS}}^{\mathrm{avg}}$	average objective value of three runs performed in the tabu search heuristic

$z_{\mathrm{TS}}^{\mathrm{best}}$	the best objective value obtained by the tabu search heuristic
z_{GH}	objective value obtained by greedy heuristic
z^*	optimal objective value provided by complete enumeration
$Z_{bestneigh}$	objective value of the best neighbor
Z_{incum}	objective value of the incumbent solution
Z_{neigh}	objective value of a neighbor solution
Z_{wt}^k	amount of flow between node pair w on path k in period t
α_w^{ki}	1 if path k in node pair w includes node i; 0 otherwise
β_{itl}	ratio

LIST OF ACRONYMS/ABBREVIATIONS

BIP	Binary Integer Problem
BMPNIP	Bilevel Multi-period Network Interdiction Problem
CEM	Complete Enumeration Method
CPU	Central Processing Unit
GH	Greedy Heuristic
TSH	Tabu Search Heuristic

1. INTRODUCTION

Critical infrastructure networks are designed to provide any product or service and they consist of components such as facilities, stations, connection lines and so on. There may be a partial interdiction or a complete interdiction in a critical infrastructure network. When there is a partial interdiction, network components become partially inoperable and serve under their capacity. When there is a complete interdiction, network components become completely inoperable and out of service. Both interdiction situations cause a significant decrease in the quality of the service. For example, a damage in a transportation network may result in shutdown of a station or a transportation line for access, a disruption in an electric power network may bring serious reductions in the capacity of electricity distribution and an interruption in a supply chain network or in a telecommunication network may lead to important problems while providing service to customers.

These critical infrastructure disruptions can be divided into two main groups: unintentional causes and intentional causes. In the first group, there are natural disasters such as flood, hurricane, earthquake, tsunami, volcanic eruption, avalanche etc. Earthquake and tsunami in Indonesia, flooding and mudslides in Japan, dust storms in India and hurricane in Florence are some of natural disasters that occurred in 2018. Random man-made causes which occur because of unintentional human or system error are also located in the first group. For example, in Maryland, a CSX freight train burned as a result of an accident in 2001 and internet infrastructure was damaged because of that unintentional event [1]. The world's most recent and unintentional man-made disruption was the BP Deepwater Horizon oil spill that took place in the Gulf of Mexico in 2010.

The second group contains intentional causes created by humans. These may be terrorist activities that can lead to loss of life or property, as well as cyber-attacks caused by hackers against technological systems. Examples of terrorist activities are metro bombings in France, England and Russia, train bombings in Spain, suicide at-

tacks in Turkey. Such terrorist attacks lead to great disruptions and costs other than life losses. When the World Trade Center was attacked in 2001, communication infrastructure was greatly damaged, including about 300,000 lines, more than 3.5 million data circuits, and almost a dozen cellular towers [1]. 2004 Madrid train bombings not only killed 191 people, but also caused serious damage to the houses as well as to the railway infrastructure, resulting in a loss of 212 million euros [2].

An analysis of the bombings in London shows that the attacked stations were not random choices. Although there were about 3 million combinations to select three stations in the metro network, the attack was made close to the best option among these combinations [3]. Therefore, it can be said that the maximum damage or disruption usually occurs by intentional causes rather than random causes.

The vulnerability analysis of a critical infrastructure network shows the network sensitivity to attack. The vulnerability of a network is measured by how much disruption will occur in the network when a situation that causes any damage is encountered. The operational resilience of a network is measured by the amount or the quality of the service that the components operating within the network can provide after a disruption. The service quality of critical infrastructure networks with low operational resilience shows the vulnerability of these networks. On the contrary, the relatively less affected networks in service providing have higher operational stability and lower vulnerability. Smith [4] indicates that the most vulnerable components in a complex critical infrastructure network can be found effectively with the help of interdiction models and these models can be used together with the fortification models to demonstrate how to increase the resilience of the network. These models are based on the assumption that there is a virtual attacker who wants to cause the greatest damage to the critical infrastructure network. Therefore, it is necessary to think from the perspective of an intelligent attacker in order to predict the worst disruption and to measure the resilience of a critical infrastructure.

Interdiction models can be formulated as multi-level programming models. Two-level (bilevel) mathematical models are the most commonly used multi-level program-

ming models in optimization problems. These models have two independent players within a hierarchical structure: upper-level decision maker (leader) and lower-level decision maker (follower). These players make their decisions sequentially with the aim of optimizing their own objective functions corresponding to a Stackelberg game [5]. First, the leader chooses its own strategy. The follower then solves its own problem considering the leader's strategy. The leader also takes into account the follower's reaction to optimize its own objective function. In other words, one of the constraints of the leader's optimization problem is the follower's optimization problem. Objective functions of the players are mostly in conflict with each other which means that while one player wishes to maximize a certain objective function, the other player may wish to minimize the same objective function. A general two-level optimization model can be shown as below [6]:

$$\min_{\mathbf{X}} F(x,y) \tag{1.1}$$

$$\text{s.t.} \quad G_i(x,y) \leq 0 \qquad \forall i \tag{1.2}$$

$$\min_{\mathbf{Y}} f(x,y) \tag{1.3}$$

$$\text{s.t.} \quad g_i(x,y) \leq 0 \qquad \forall i \tag{1.4}$$

x is the decision vector of the leader, whereas y is the decision vector of the follower. $F(x,y)$ is the objective function of the leader's optimization problem, while $f(x,y)$ is the objective function of the follower's optimization problem. $G_i(x,y) \leq 0$ is the ith constraint of the leader's problem, $g_i(x,y) \leq 0$ is the ith constraint of the follower's problem.

In this study, a bilevel multi-period network interdiction problem (BMPNIP) in a transportation network is addressed by taking the recovery time of interdicted components. The proposed bilevel model includes an integer mathematical model in the upper level and a mixed-integer model in the lower level. A tabu search algorithm is developed to solve the formulated bilevel problem. In addition to the tabu search heuristic (TSH), two shortest path models are used in a pre-processing step of the algorithm. These shortest path models determine the admissible paths between source and

destination nodes. The experiments are carried out on randomly generated instances. In order to understand the performance of TSH, a complete enumeration and a greedy approach are developed.

The rest of this study is organized as follows. In Chapter 2, the literature related to the interdiction models is reviewed. The problem together with its properties are introduced in Chapter 3. The proposed tabu search solution method and two different solution approaches are presented in Chapter 4. Chapter 5 provides computational results on randomly generated instances. Finally, conclusions related to this study are discussed in Chapter 6.

2. LITERATURE REVIEW

We survey the studies on determining of the most vulnerable components and on increasing the operational resilience of critical infrastructures and divide them into two groups: attacker-operator models and operator-attacker models.

2.1. Attacker-Operator Models

The problems including an intelligent attacker who wants to cause the most disruption in a critical infrastructure network and an operator who wants to operate the system effectively after the disruption are generally called attacker-operator problems.

Attacker-operator models can be further divided into two main groups as network interdiction and facility interdiction models. In both types of models, the attacker is the upper-level decision maker and the system operator is the lower-level decision maker.

(i) *Facility interdiction:* The target of the virtual attacker in a facility interdiction model is the facilities that serve customers in the critical infrastructure network. In these models, it is assumed that customers are assigned to undamaged facilities under capacity constraints.

(ii) *Network interdiction:* The components of the critical infrastructure networks are attacked in network interdiction models. When there are damaged components, the system operator in the lower level tries to either find the shortest path between two nodes of the network or maximize the total flow in the network.

Network interdiction models have begun to be examined much earlier in the literature. Therefore, they are more common in the literature. The first study on network interdiction belongs to Wollmer [7], where the author examines the minimization of the maximum possible flow between two nodes by removing a certain number of arcs in a network.

Then, Wood [8] considers a network interdiction problem with total and partial interdiction of arc capacities under resource budget and number of arcs to be interdicted constraints. In this paper, a drug career wants to maximize the flow of drug chemicals while the anti-drug effort of army wants to minimize the flow of drug chemicals by interdicting arcs on the possible paths using limited resources. It is also shown that the problem is NP-complete, even if one unit of resource is required to interdict an arc. Altner et al. [9] extends the study of Wood [8] by considering a cardinality maximum flow network interdiction problem by removing k arcs from the network.

Cormican et al. [10] study a network interdiction problem where interdiction occurs with a certain probability. The objective of the problem is to minimize the maximum expected flow in the network. They formulate two-stage stochastic integer models where network arcs are destroyed in accordance with the Bernoulli process with a probability of 1-p in the objective function of the attacker. In addition, the same stochastic network interdiction problem is investigated by Janjarassuk and Linderoth [11] and the authors develop a deterministic mixed-integer program utilizing duality and linearization techniques. On the other hand, Ramirez-Marquez and Rocco [12] consider a stochastic network interdiction problem in a capacitated network and aim to minimize the interdiction cost in their study.

The shortest path network interdiction problem, which is generally based on the undesired increase of arc lengths, is first defined by Fulkerson and Harding [13]. They search for the maximum shortest path under a budget constraint. Later, Israeli and Wood [14] examine the problem of Fulkerson and Harding [13] under the complete interdiction assumption of the arcs. Bayrak and Bailey [15] study a shortest path network interdiction problem while the attacker and the system operator have asymmetric information on the arc lengths. Khachiyan et al. [16] also address the shortest path interdiction problem by Israeli and Wood [14] by considering the removal of a limited number of nodes from the network. Lim and Smith [17] apply the same idea for multi-commodity network flow problem. In this study, the aim of the attacker is to minimize the maximum profit that is obtained from delivering commodities through the network, where interdiction can be complete or partial.

The stochastic shortest network interdiction problem has also been studied in the literature. Zhang et al. [18] examine a stochastic shortest path network interdiction problem which maximizes the minimum expected shortest travelling time. The defender allocates sensors to the arcs within a budget limit. A detection probability of the sensor is determined to detect the attacker. Therefore, the attacker wants to find the path with minimum detection probability. They take two models into account for the problem: a single source and a single destination model, and a multiple source and a multiple destination model. They conclude that an increase in the budget or detection likelihood leads to an increase in the expected shortest travelling time. Furthermore, the single source-destination model has longer time than the multiple source-destination model.

The problem of maximizing the expected shortest distance between a given source and destination pair is also analyzed in the work of Held and Woodruff [19], where they propose a heuristic solution procedure. In addition, Held et al. [20] solve the same problem using a decomposition method.

Yates and Lakshmanan [21] suggests a binary knapsack approximation with the aim of minimizing the maximum network undetected probability. In this study, an attacker chooses the highest path probability of not being detected by a defender, whereas a defender assigns its limited resources in the network to decrease the maximum undetected path probabilities between origin-destination pairs. Path non-detection probabilities are formed from the corresponding arc non-detection probabilities. Statistical and spatial analysis are also applied to two real-world networks to understand the quality of knapsack approximation method.

Minimization of the maximum flow between two nodes is also studied by Lunday and Sherali [22]. They analyze various synergies of the resources to determine their effects on the arcs in a capacitated network. Synergies are examined according to the ratio of the synergy effect on each arc to the minimum interdiction by any of the possible resources. Three different synergy relationships are observed, namely linear synergy relationships, concave, and convex-concave. For linear synergy relationships,

a direct solution is proposed using ILOG CPLEX which is a commercial optimization software. For concave and convex-concave synergy relationships, linearization based heuristics are employed. In addition, partial arc interdiction is also allowed in this study.

Bertsimas et al. [23] identify a randomized network interdiction problem that enables randomly selecting the arcs to be removed. They formulate an arc-based and a path-based model, where the flow is either on arcs or on paths.

Maximizing the disconnectedness in undirected graphs is studied by Shen et al. [24]. The disconnectedness is obtained by removing a subset of nodes. They use three metrics to measure the network-connectivity of a graph: maximization of the number of connected components, minimization of the largest component size and maximization of the minimum cost to reconnect the graph after the nodes are eliminated. Randomly generated instances are used in these problems and a mixed-integer program is defined for each problem. They also solve the first two problems by using valid inequalities to increase the computational efficiency.

Zenklusen [25] take two different interdiction problems into consideration: an edge interdiction problem and a vertex interdiction problem. These problems consider the possibility of removal of a set of the edges and vertices. Edges and vertices have removing costs as well as positive weights in an undirected graph. The vertex interdiction problem aims to minimize the weight of a maximum matching in the graph under a budget constraint. The same logic is applied to the edge interdiction problem. Since both problems are NP-hard, they are approximately solved.

Akgün et al. [26] consider a maximum flow interdiction problem in a multi-terminal network. They develop a mixed-integer model by remodeling a bilevel min-max problem which minimizes the maximum objective value. They also propose an approximating binary integer program. They observe that the approximating model can be solved on all instances in less time than the mixed-integer model. The results of both models are different in some instances but they showed that their model has a

good approximation method to solve the problem.

Hausken and Zhuang [27] examine four different defense and attack games between a government and a terrorist: first, the government neither attacks nor defends when the terrorist attacks; second, the government attacks and the terrorist is deterred; third, the terrorist attacks and the government defends its assets; fourth, both the government and the terrorist attack and defend. The authors try to find the critical factors and their interactions in coping with terrorism. Therefore, they consider the government's asset valuations, the terrorist's resources, the terrorist's asset valuations, unit defense and attack costs, the government's discount factors and the terrorist's discount factors. It is assumed that the resource allocation of the terrorist follows a geometric distribution with a stockpiling parameter over a planning horizon. They examine the periods when the terrorist allocates its resources.

Shan and Zhuang [28] take into account the terrorist's strategic or non-strategic decisions that impacts the government's defense allocation decision to minimize total expected loss. Based on the terrorist's strategic and non-strategic decisions, game theoretic and non-game-theoretic models are respectively considered for defensive resource allocations. The major difference between these models is that game-theoretic models account for the attacker's reaction as a function of the defender's allocation choice, whereas in non-game-theoretic models, the attacker's reaction is externally derived. Their findings show that the defender prefers game-theoretic models to non-game-theoretic models while allocating its defensive resources.

Bell et al. [29] try to detect a road network vulnerability and decision making by using a game theoretic approach where the attacker and the defender do not know each other's decisions precisely. They aim to find the minimum of the maximum expected loss due to predefined interdiction scenarios. The worst case scenario probabilities are calculated for the use of a road network, which is called as a mixed route strategy and is similar to risk-averse route selection.

On the other hand, multi-objective models have also been formulated for interdiction problems. For example, Royset and Wood [30] propose an extension of the deterministic maximum flow network interdiction problem using a single-level bi-objective interdiction model. In the proposed model, the attacker wants to minimize the total cost of interdiction and the maximum flow on the network, while completely eliminating a set of arcs in a capacitated network. Lagrangian relaxation and a specialized branch and bound algorithm are employed to solve the problem.

Rocco and Ramirez-Marquez [31] develop an algorithm to solve bi-objective shortest path problems. Two discrete objectives are simultaneously used: maximizing the shortest-path length and minimizing the interdiction cost. A Monte Carlo simulation based algorithm is developed to obtain possible interdiction strategies and graph theoretical approaches are used to find the shortest paths or the most reliable paths for those strategies. In addition, a probabilistic scenario show that the maximization of the shortest path is equivalent to the minimization of the most reliable path.

A bilevel network interdiction problem of hazardous material transportation is given in [32]. The goal is to minimize the total distribution cost by considering the selection of safe arcs although the attacker wants to maximize the number of total risky arcs. This proposed bilevel model is solved by two meta-heuristic algorithms: co-evolutionary algorithm and a disjunctive bound embedded co-evolutionary algorithm. Co-evolutionary bilevel method is developed based on Legillon's vehicle routing algorithm [33]. Two different sub-populations are created from two levels of the problem. Each population refers to a distinct genetic algorithm. During the iterations, the best solutions are collected. Then, two sub-populations are evaluated in cooperation at the final iteration. Co-evolutionary algorithm with disjunctive bound method differentiates from co-evolutionary method by adapting disjunctive lower bound while using branch-and-bound to have better lower level solutions. After the authors test their algorithms on randomly generated problems, co-evolutionary method with disjunctive bound method result in a better average total risk and rationality than the co-evolutionary method.

Electricity distribution network interdiction problems are also investigated in the literature. Salmeron et al. [34] develop a two level mathematical model and define the significant components in electric power grids by utilizing a heuristic method. Different than this study, Arroyo and Galiana [35] write the Karush–Kuhn–Tucker optimality conditions to convert the nonlinear bilevel model into a single level one and linearize the resulting model.

Enayaty-Ahangar et al. [36] study a multi-period bilevel network interdiction problem which aims to minimize the cumulative maximum flow of illegal drugs over a finite time horizon within a network. They adopt a logic based decomposition approach to solve the problem.

A maximum dynamic network flow interdiction problem is investigated in the study of Afshari Rad and Kakhki [37]. The authors introduce a transit time for the flow on each arc and minimize the maximum flow through arc within a given time limit. They make use of Ford and Fulkerson's Temporally Repeated Flows [38] that are obtained from dividing a feasible flow into sub-flows. Then, they reformulate the problem and utilize Benders' decomposition to solve it.

The first facility interdiction model in the literature is introduced by Church et al. [39]. In this study, the authors develop two models for site location problems: the r-interdiction covering problem and the r-interdiction median problem. In the r-interdiction covering problem, the aim of the attacker is to find out which r of p facilities to destroy, so that the number of covered customers has the highest decrease after the interdiction. In the r-interdiction median problem, they try to determine which r of p facilities should be chosen by the attacker to increase the total demand-weighted distance. After the interdiction, customers who cannot get service from a facility are assigned to the closest of remaining p-r operating facilities.

Murray et al. [40] are concerned with telecommunications network problems. They consider a facility interdiction model by allowing only node interdiction, where the number of interdicted facilities is given in advance. The objective of the attacker

is to either maximize or minimize the flow after the attack.

2.2. Operator-Attacker Models

Operator-Attacker models are also known as protection-interdiction models in the literature. Protection can reduce or completely prevent the disruption of a component in the network. As in the attacker-operator models mentioned above, these models can be divided into two groups as network protection-interdiction and facility protection-interdiction.

A multi-level protection-interdiction model is formulated by Snyder et al. [41] to determine the components that should be reinforced or protected in a critical infrastructure. The amount of protection was limited to either a budget constraint or the number of components to be protected.

Church and Scaparra [42] include the facility protection into the previously developed r-interdiction median problem and analyze the protected interdiction median problem as a single-level mathematical model. Scaparra and Church [43] solve the same problem using an implicit enumeration solution method in a search tree. Scaparra and Church [44] extends this study by taking into account the facility capacity and they develop a three-level defender-attacker-user model. They transform the model into a two-level one using the dual of the operational problem in the third level. Various network sizes and protection resource levels are analyzed to observe the impact on the system cost.

Smith and Lim [45] review the models where the operator increases the capacity of the facilities before the interdiction, reduce the flow costs and protect some components.

Aksen et al. [46] discuss another version of the r-interdiction median problem, and investigate the situation where an attacker damages facilities under a certain interdiction budget constraint rather than a fixed number of fortified facilities.

Liberatore et al. [47] study the stochastic r-interdiction median problem with protection and make use of heuristic approaches to solve it. The number r is not known with certainty in the addressed problem.

Losada et al. [48] develop a bilevel model that includes the partial protection of facilities of infinite capacity in median type networks. In addition, the recovery time of the facilities after the interdiction is taken into account.

In the literature, studies on network protection-interdiction are much more limited compared to those including facility protection-interdiction. Cappanera and Scaparra [49] determine the best distribution of protection resources in the shortest path network where the attacker targets unprotected network components. Alguacil et al. [50] formulate a three-level model for optimal protection decisions in an electricity distribution network. Sarhadi et al. [51] also develop a multi-level mathematical programming model in which protection decisions are made for intermodal stations in a railway network. Jin et al. [52] propose a model for optimal distribution of protection resources in a railway network where stations have different disruption intensities. Sadeghi et al. [53] highlight the partial reinforcement while proposing a three-level shortest path network interdiction model.

There is a very limited number of multi-period network protection-interdiction and facility protection-interdiction models in the literature. Two remarkable studies belong to Losada et al. [48] and Malaviya et al. [54], where the periodic interdiction decisions and constraints are considered in the models. On the other hand, in the study of Starita and Scaparra [55], fortification or protection decisions are also periodically given by the leader.

3. PROBLEM DEFINITION

The critical infrastructure network addressed in this study is a transportation network. Components of such a network are the stations and links between the stations, where a station is represented by a node and a link by an arc. It is assumed that only the stations are interdicted by the attacker.

A bilevel attacker-operator model is formulated to solve a multi-period network interdiction problem. In this model, the leader is the attacker who wants to cause the most damage in order to minimize the flow in the transportation network after the attack, whereas the follower is the operator who wants to maximize the flow after the virtual attacker causes a disruption in the network. We try to determine the most vulnerable components in the transportation network during the planning horizon. In other words, the aim is to identify the most significant stations leading to the maximum flow decrease in the network.

The problem is studied over a planning horizon. Each station has an interdiction cost and the attacker makes his/her interdiction decisions under a limited resource or budget within the planning horizon. Interdiction decisions given by the attacker result in complete interdiction, which means that each interdicted component in the network is completely destroyed, and hence it cannot provide any service for some time. Stations have capacity restrictions and these capacities are sufficient to supply all demand when there is no disruption in the network.

The recovery periods of the stations are known with certainty. They refer to the repair time required to make an interdicted station operational again. Repair times will connect the periods in the planning horizon and make it impossible to decompose the problem with respect to periods. For example, if a station is attacked, it does not affect only the attacked station during the repair period, it also affects the flow on all paths on which the station exists. The system operator attempts to maximize the flow in the system affected by the repair time of the station.

We assume that a passenger can use an alternative path other than the shortest path, which is the second shortest path to reach the destination node. We refer to the shortest path and the alternative path as admissible paths. There is a preliminary phase to detect the admissible paths between source and destination nodes. Yen [56] and Eppstein [57] propose different algorithms in their studies in order to find the k-shortest paths in a network. However, we use the shortest path models below because they are also efficient and sufficient for the preliminary phase. Flow demand occurs between a given pair of nodes and its value is known. They are referred to as a source-destination (SD) pair.

The shortest path models are solved for each SD pair in the network with the help of GAMS by using CPLEX. For the first shortest path model, we define the following sets, parameter and decision variable.

Sets:

$i, j \in N$: set of nodes in the network
$b \in B \subseteq N$: set of source nodes in the network
$s \in S \subseteq N$: set of destination nodes in the network

Parameter:

g_{ij} : distance from node i to node j

Decision Variable:

Y_{ij} : 1 if the arc between node i and node j is in the selected path;
0 otherwise

The first shortest path model is formulated as follows:

$$\min \sum_{i} \sum_{j} g_{ij} Y_{ij} \tag{3.1}$$

$$\text{s.t.} \quad \sum_j Y_{bj} - \sum_j Y_{jb} = 1 \tag{3.2}$$

$$\sum_j Y_{js} - \sum_j Y_{sj} = 1 \tag{3.3}$$

$$\sum_j Y_{ij} = \sum_j Y_{ji} \qquad i \in N \backslash \{b, s\} \tag{3.4}$$

$$Y_{ij} \geq 0 \qquad i, j \in N \tag{3.5}$$

Objective function (3.1) minimizes the distances between node i and node j. Constraint (3.2) describes the difference in the number of arcs leaving source node b and entering into it is one. Constraint (3.3) describes the difference in the number of arcs entering into destination node s and leaving it is one. Constraints (3.4) are balance equations for each node i excluding source node b and destination node s. Constraints (3.5) show that decision variable can only take non-negative values.

In order to find the second shortest path between SD pairs, we need to reformulate the first shortest path model by adding some parameters, decision variables, and constraints.

Additional Parameters:

m : number of nodes

a_{ij} : 1 if there is an arc between node i and node j in the first shortest path; 0 otherwise

Additional Decision Variables:

U_i : order of visiting node i in the second shortest path

Additional Constraints:

$$\sum_{i:a_{ij}=0} \sum_{j:a_{ij}=0} Y_{ij} \geq 1 \tag{3.6}$$

$$U_i - U_j + mY_{ij} \leq m - 1 \qquad i, j \in N \quad \text{where } U_b = 1 \text{ and } U_i \neq U_j \qquad (3.7)$$

$$U_i > 0 \qquad i \in N \qquad (3.8)$$

Constraints (3.6) ensure that the same shortest path is not selected. Constraints (3.7) are sub-tour elimination constraints formed by Miller, Tucker, and Zemlin [58]. Constraints (3.8) provide that node orders take only positive values.

The shortest path examples between pairs can be seen in Figure 3.1. In the represented network, there are three pairs, which are (4, 1), (6, 2), and (10, 9). There are arcs between nodes and they are two-way. The distances between nodes are shown on corresponding arcs. Each pair has two paths found by the above shortest path models. The arrows between the nodes indicate the flow direction. For example, the first admissible path is 4-9-1 and the second admissible path is 4-9-2-3-1 from source node 4 to destination node 1.

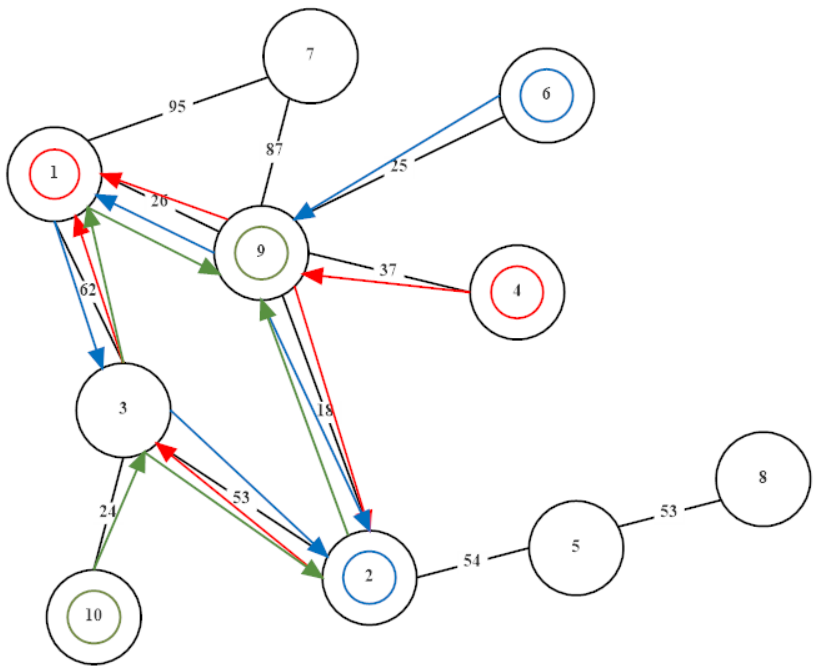

Figure 3.1. Shortest paths on a network

3.1. Bilevel Multi-period Network Interdiction Model

The upper level problem of BMPNIP is an integer linear programming model, whereas its lower level is a mixed-integer linear one. Although the objective functions of the attacker (leader) and the operator (follower) are the same, the attacker wants to minimize it, whereas the operator maximizes it. The BMPNIP is presented with the following sets, parameters and decision variables.

Sets:

$i \in N$: set of nodes in the network
$b \in B \subseteq N$: set of source nodes in the network
$s \in S \subseteq N$: set of destination nodes in the network

$w \in W$: set of node pairs (b, s) which have passenger flow demand
$k \in K_w$: set of admissible paths between node pair w
$t \in T$: set of periods
$l \in L$: set of recovery periods, i.e. $L = \{0, ..., \bar{L}\}$

Parameters:

f_{wt} : flow demand between node pair w in period t
d_{it} : capacity of node i in period t,
o_{it} : amount of resource required for complete interdiction of node i in period t
r : budget of the attacker
α_w^{ki} : 1 if path k in node pair w includes node i; 0 otherwise

Decision Variables:

Z_{wt}^k : amount of flow between node pair w on path k in period t
X_{it} : 1 if node i is interdicted in period t; 0 otherwise
P_{it} : 1 if node i is operating in period t; 0 otherwise

Now we can formulate our mathematical model below:

$$\text{BMPNIP}: \min_{\mathbf{X}} \sum_{w\in W}\sum_{t\in T}\sum_{k\in K_w} Z_{wt}^k \tag{3.9}$$

s.t.

$$\sum_{i\in N}\sum_{t\in T} o_{it}X_{it} \le r \tag{3.10}$$

$$X_{it} \in \{0,1\} \qquad i\in N,\ t\in T \tag{3.11}$$

$$\max_{\mathbf{Z},\mathbf{P}} \sum_{w\in W}\sum_{t\in T}\sum_{k\in K_w} Z_{wt}^k \tag{3.12}$$

s.t.

$$P_{it} \le 1 - X_{it} \qquad i\in N,\ t\in T \tag{3.13}$$

$$P_{i,t+l} - P_{it} \le 1 - X_{it} \qquad i\in N,\ t\in T,\ l\in L \tag{3.14}$$

$$\sum_{w\in W}\sum_{k\in K_w} \alpha_w^{ki} Z_{wt}^k \le\ d_{it}P_{it} \qquad i\in N,\ t\in T \tag{3.15}$$

$$\sum_{k\in K_w} Z_{wt}^k \le\ f_{wt} \qquad w\in W,\ t\in T \tag{3.16}$$

$$Z_{wt}^k \ge\ 0 \qquad w\in W,\ t\in T,\ k\in K_w \tag{3.17}$$

$$P_{it} \in \{0,1\} \qquad i\in N,\ t\in T \tag{3.18}$$

The objective function (3.9) minimizes the total flow on admissible paths between all SD pairs in all periods. Constraints (3.10) show the resource restriction of the attacker, which is used to disrupt the nodes. Constraints (3.11) put binary restriction on the attacker's decision variables. The follower (operator) wants to maximize the same objective function given by (3.12). The constraints of the operator are given from (3.13) to (3.18). Constraints (3.13) indicate that if node i in period t is disrupted, then node i will not be operating in period t. Constraints (3.14) provide that if node i is interdicted in period t, then it will not be operating during its recovery periods either. Constraints (3.15) ensure that the total flow on the admissible paths cannot exceed the remaining capacity of node i after interdiction. Constraints (3.16) guarantee that there is an upper limit on the amount of flow between source nodes and destination nodes for every period. Constraints (3.17) ensure that the flow on the admissible paths between

node pair w can only take non-negative values. Constraints (3.18) define whether node i operates in period t.

Note that node i may be interdicted in period t, in that case X_{it}=1 which implies that $P_{i,t+l}$=0 as it is not operational in period t and in the next l periods during which the node will be recovered. Hence, it is possible to eliminate constraints (3.13) and (3.14). In addition, we can rewrite constraint (3.15) according to the values of P_{it}. So if X_{it}=0, then P_{it}=1 and the right hand side of the constraint is $d_{i,t}$. If X_{it}=1, then $P_{i,t+l}$=0 and the right hand side of the constraint is zero.

4. SOLUTION PROCEDURE

In general, bilevel mathematical programming problems are more difficult than single-level mathematical programming problems. Jeroslow [59] and Bard [60] show that these problems are NP-hard. Hence, a tabu search heuristic is developed to find good feasible solutions. On the other hand, the BMPNIP is solved by using complete enumeration and a greedy approach to assess the performance and the quality of the tabu search heuristic.

4.1. Tabu Search Heuristic

Tabu search algorithm is a well-known metahuristic introduced by Glover [61]. The algorithm starts with an initial solution and moves to better solutions. The neighborhood of a solution is generated by using a set of move operators for current solution at each iteration. A tabu list is used to avoid visiting the same solutions for a number of iterations called tabu tenure. The best neighbor solution is chosen as the new solution unless it is obtained by a tabu move. If the best neighbor solution is found by a tabu move and better than the incumbent solution, then it can be chosen as the new solution. This is called the aspiration criterion. Tabu search procedure is continued until the termination criteria are satisfied.

In order to find good feasible solutions to BMPNIP in a reasonable amount of time, a TSH is employed. In accordance with this purpose, TSH makes the search over the attacker's binary decision variables. These variables are fixed in the lower level problem. Then, the mixed-integer lower level problem is exactly solved with the commercial solver CPLEX.

A binary integer problem (BIP) is solved from the perspective of the attacker to find an initial solution. The objective of the model maximizes the number of interdicted nodes under the budget constraint by ignoring the reaction of the follower. However, in order to prevent attacking the same nodes in consecutive periods and thus not wasting

the attack budget, the recovery time of each node is considered in the model.

$$\text{BIP}: \max \sum_{i \in N} \sum_{t \in T} X_{it} \tag{4.1}$$

s.t.

$$\sum_{i \in N} \sum_{t \in T} o_{it} X_{it} \leq r \tag{4.2}$$

$$X_{i,t+l} \leq 1 - X_{it} \qquad i \in N,\ t \in T,\ l \in L \tag{4.3}$$

$$X_{it} \in \{0, 1\} \qquad i \in N,\ t \in T \tag{4.4}$$

The objective function (4.1) maximizes the number of attacked nodes in the planning horizon. Constraint (4.2) imposes the resource budget restriction of the attacker. Constraints (4.3) imply that if node i is interdicted in period t, it will not be interdicted again during its recovery periods. Finally, constraints (4.4) represent binary restrictions on the attacker's decision variables.

The parameters of TSH have a significant effect on the solution quality and CPU times. Therefore, some preliminary experiments are done to determine appropriate parameter values for the algorithm. The termination criteria used in the algorithm are the maximum number of iterations and the maximum number of non-improving iterations. While searching the neighbor solutions, three move operators are applied in TSH: 1-Swap move, 1-Add move, and 1-Drop move.

1-Swap move: We randomly choose a node interdicted in a period (i.e. a node-period pair for which X_{it}=1 in the current solution) and a node which is not interdicted in a period (i.e. a node-period pair for which either $X_{it'}$=0 or $X_{i't}$=0 in the current solution). We calculate the total cost of the current interdiction by subtracting the cost of the selected interdicted node-period pair, and adding the cost of the selected non-interdicted node-period pair. If the resulting total cost is higher than the budget, then we eliminate the selected non-interdicted node-period pair from the possible combinations and randomly select another non-interdicted node-period pair. If the total cost is less than the budget, then we make the selected interdicted node-period pair

as non-interdicted and the selected non-interdicted node-period pair as interdicted. In other words, we switch the selected interdicted node-period pair with the selected non-interdicted node-period pair, and update the current solution. We continue the search until either there is no possible combination or the number of moves reaches the neighborhood size.

1-Add move: It is used if the total cost of the current interdiction is less than the budget. We randomly choose a non-interdicted node-period pair from all combinations of non-interdicted node-period pairs. We calculate the total cost of the current interdiction by adding the cost of the selected non-interdicted node-period pair. If the new total cost is higher than the budget, then we eliminate the selected non-interdicted node-period pair from the possible combinations and randomly select another non-interdicted node-period pair. We continue the search until either there is no possible combination or the number of moves reaches the neighborhood size.

1-Drop move: We randomly choose an interdicted node-period pair from the current solution. The selected interdicted node-period pair is made non-interdicted. The search is continued for all interdicted node-period pairs in the current solution.

In every iteration of TSH, these move operators are executed and the best neighbor is determined. Each move has a separate tabu list. After the best neighbor is chosen, the opposite of the best move is added into the corresponding tabu list. For example, if the best neighbor is obtained by a 1-Add move, then it is tabu to drop the selected node-period pair. The same logic applies for 1-Drop moves. If 1-Swap move gives the best neighbor, then it is tabu to swap the selected node-period pairs.

The proposed tabu search algorithm can be seen in Figure 4.1, Figure 4.2 and Figure 4.3 with the following notations.

$iter$: iteration counter
$maxiter$: maximum number of iterations

nonimpiter	: counter of non-improving iteration during which the incumbent solution does not improve
maxnonimpiter	: maximum number of non-improving iterations
$Z_{bestneigh}$	: objective value of the best neighbor
Z_{neigh}	: objective value of a neighboring solution
Z_{incum}	: objective value of the incumbent solution
c_{neigh}	: total cost of a neighbor solution
c_{curr}	: total cost of the current solution
h	: tabu tenure
h'	: current size of the tabu list

Set *nonimpiter*=0, *iter*=0, h'=0;
Solve BIP to find the initial values of the attacker's decision variable;
Fix the initial values of the attacker's decision variable in the lower level problem;
Calculate the objective value of the follower by solving it with Cplex;
Set the obtained solution as the current solution and incumbent solution;
while *iter* < *maxiter* and *nonimpiter* < *maxnonimpiter* **do**
    Set $Z_{bestneigh} \leftarrow \infty$;
        for 1-Add move and 1-Swap move **call** Algorithm 1
        for 1-Drop move **call** Algorithm 2
        if the best move is not previously added to the corresponding tabu list **then**
            Add tabu move to the tabu list;
            if $h' > h$ **then**
                Remove the oldest tabu move from the list;
            end if
        end if
    Set the best neighbor solution as the current solution;
    iter=*iter*+1;
    nonimpiter=*nonimpiter*+1;
end while
return incumbent solution

Figure 4.1. Tabu Search Main Algorithm

```
for each feasible neighbor do
    Compute Z_neigh by solving the lower level problem;
    if Z_neigh < Z_incum then
        Update the best neighbor solution and incumbent solution as neighbor solution;
        Set nonimpiter = 0;
    else if move to the neighbor solution is in the tabu list then
        Discard it and Continue with the next neighbor;
    end if
    if Z_neigh < Z_bestneigh and move to the neighbor solution is not in the tabu list then
        the best neighbor solution ← neighbor solution;
    end if
end for
```

Figure 4.2. Algorithm 1

```
for each feasible neighbor do
    Compute Z_neigh by solving the lower level problem;
    if Z_neigh < Z_incum then
        Update the best neighbor solution and incumbent solution as neighbor solution;
        Set nonimpiter = 0;
    else if move to the neighbor solution is in the tabu list then
        Discard it and Continue with the next neighbor;
    end if
    if Z_neigh < Z_bestneigh and move to the neighbor solution is not in the tabu list then
        the best neighbor solution ← neighbor solution;
    else if Z_neigh = Z_bestneigh and c_neigh < c_curr and move to the neighbor solution is not in the tabu list
    then
        the best neighbor solution ← neighbor solution;
    end if
end for
```

Figure 4.3. Algorithm 2

4.2. Complete Enumeration Method

The complete enumeration method (CEM) explores all possible solutions that are feasible with respect to budget constraint (i.e. interdiction configurations). In order to find the objective value of the attacker for a given interdiction configuration, the values of the of the attacker's decision variables are fixed in the lower level problem, which is solved to optimality by Cplex. The CEM can be used in small sized networks with node interdiction cost equal to one. It is not possible to obtain a feasible solution for large sized networks or medium sized networks with differentiated node interdiction costs in a reasonable amount of time.

4.3. Greedy Heuristic

A greedy heuristic (GH) is also designed to assess the quality of TSH on large sized networks. First, a ratio (β_{itl}) is defined by using the amount of flow at a node in a period (f_{it}), the amount of flow at a node in a recovery period ($f_{i,t+l}$), and the amount of resource required for complete interdiction of a node in a period (o_{it}).

$$\beta_{itl} = \frac{f_{it} + f_{i,t+l}}{o_{it}} \qquad \forall i, t, l \quad \text{where} \quad l > 0 \tag{4.5}$$

Node-period pairs are sorted in descending order of the calculated ratios and the node-period pair with the highest value is selected as the interdicted node-period pair provided that it does not exceed the budget of the attacker. In case two ratios are equal, the number of paths passing through a node in a period are taken into account. The nodes that appear in the same path with the interdicted node-period pair are also affected by the attacker's decision. Therefore, after a node-period pair is selected for interdiction, the number of paths passing through the affected nodes and the amount of flow at these nodes are recalculated. The search continues until the attacker totally exhausts its budget. Finally, when the search is terminated, the follower's objective value is found by using the current selection.

5. COMPUTATIONAL RESULTS

In this chapter, we compare the heuristics developed for the solution of bilevel network interdiction problem in terms of solution quality and computational efficiency. As there is no benchmark study in the literature, we generate random test instances for this purpose.

All algorithms have been coded in C# and the computations have been performed on a workstation with Intel Xeon 2.60 GHz processor and 64 GB of RAM working under Windows 10 operating system. The follower's problem as well as the BIP model generating the initial solution for TSH are solved by CPLEX 12.7.

A number of instances are created by varying the number of nodes (m), the density of arcs, and the number of SD pairs in a network. The x and y-coordinates of the nodes are integer numbers generated from a uniform distribution defined in the interval [1, 100×(m-1)]. Arc densities are set as 30% and 44% of the total number of arcs given as m×(m-1). The distance between any two nodes are assumed to be symmetric and calculated as the Euclidean distance. The flow demand, i.e the number of passengers who want to travel from a source node to a destination node, is set to one of the values from the set {100, 200, 300, 400}. The node capacities are set equal to the multiplication of the number of SD pairs in the network and the highest flow demand in SD pairs. The amount of resource required for complete interdiction of a node in a period is a real number generated from a uniform distribution defined in the interval (0,4). Node interdiction costs vary with respect to nodes or nodes and periods. Unit node interdiction cost is also used in the experiments. Six scenarios are created based on three different node interdiction cost settings and two different flow demand realizations.

- Scenario 1: different flow demands with respect to SD pairs and periods, and different node interdiction costs with respect to nodes and periods

- Scenario 2: different flow demands with respect to SD pairs and periods, and different node interdiction costs with respect to nodes
- Scenario 3: different flow demands with respect to SD pairs and periods, and unit node interdiction cost
- Scenario 4: different flow demands with respect to SD pairs, and different node interdiction costs with respect to nodes and periods
- Scenario 5: different flow demands with respect to SD pairs, and different node interdiction costs with respect to nodes
- Scenario 6: different flow demands with respect to SD pairs, and unit node interdiction cost

Two admissible paths are generated for each SD pair using the shortest path models. These admissible paths are shown in Table A.1. Finally, we determine the periods as weeks. The number of periods is set to 4 and the number of recovery periods is set to 2 weeks by including the interdicted period.

The maximum number of iterations is set to 1000 and the maximum number of non-improving iterations is set to 100. Tabu tenure in the algorithm is set to 6. The neighborhood size is set to 40% of the problem size which is obtained by multiplying the number of periods with the number of nodes in the network. For example, the problem size is 40 for a network consisting of 10 nodes and 4 periods. Thus, neighborhood size of such a network is set to 16.

5.1. Comparing the Tabu Search Heuristic and Complete Enumeration

First, we compare the performance of TSH and CEM in terms of accuracy and efficiency on a set consisting of eight instances. To evaluate the results of TSH and CEM, three runs have been executed for each instance to observe the results of TSH. The results are presented in Table 5.1 and Table 5.2. The number of nodes, the density of arcs, and the number of SD pairs are respectively indicated as “No. of nodes”, “Arc density” and “No. of SD pairs” columns. The average CPU time of three runs is given in the columns titled “CPU”. The best objective values are the minimum objective

values. The accuracy of the solutions given by TSH and CEM is measured by computing the percent deviation (PD) of the best objective value $z_{\text{TS}}^{\text{best}}$ obtained by TSH from the optimal objective value z^* provided by CEM. PD is computed by the formula given below:

$$PD = 100 \times \frac{z_{\text{TS}}^{\text{best}} - z^*}{z^*} \tag{5.1}$$

The average percent deviation and average CPU time values computed over all the instances are given in the last rows of the tables. In both Scenario 3 and Scenario 6, it is remarkable that PD of TSH is zero and TSH can find an optimal solution for every instance. In Scenario 3, the CPU time requirement of TSH is 3.58 seconds, whereas CEM spends 327.98 seconds on the average. The TSH is also more efficient than CEM in Scenario 6. As a result, although CEM is an exact technique to solve the problem, it is not efficient with respect to required computational time, even for small problem sizes. It is notable that TSH can produce good quality solutions in less CPU time.

Table 5.1. TSH and CEM comparison for Scenario 3

No. of nodes	**Arc density**	**No. of SD pairs**	**CEM**		**TSH**	
			z*	**CPU**	**PD**	**CPU**
10	30%	3	0	72.52	0%	3.07
10	30%	5	0	78.17	0%	2.34
10	44%	3	0	69.95	0%	2.98
10	44%	5	700	74.80	0%	2.04
15	30%	3	0	548.35	0%	4.78
15	30%	5	900	642.33	0%	4.08
15	44%	3	0	529.56	0%	5.16
15	44%	5	1400	608.18	0%	4.16
Average				327.98	0%	3.58

Table 5.2. TSH and CEM comparison for Scenario 6

No. of nodes	Arc density	No. of SD pairs	CEM		TSH	
			z^*	CPU	PD	CPU
10	30%	3	0	112.99	0%	3.07
10	30%	5	0	115.91	0%	2.12
10	44%	3	0	112.78	0%	2.98
10	44%	5	400	118.36	0%	2.26
15	30%	3	0	629.94	0%	3.92
15	30%	5	400	655.41	0%	4.09
15	44%	3	0	562.95	0%	3.92
15	44%	5	1200	630.67	0%	3.81
Average				367.38	0%	3.27

5.2. Comparing the Tabu Search Heuristic and Greedy Heuristic

The performance of TSH in six scenarios is compared with GH by utilizing 28 different instances. Three runs have been executed for each instance to observe the results of TSH. The results are presented in Tables 5.3–5.8. The average objective value of the runs performed in TSH is shown as $z_{\text{TS}}^{\text{avg}}$. If the best objective value z_{GH} provided by GH is equal to the best objective value $z_{\text{TS}}^{\text{best}}$ obtained by TSH, it is shown by "-" in the "Better algorithm" column. Otherwise, the algorithm providing a better solution is indicated in this column.

It is seen from the following tables that the average CPU time of TSH is between 21 seconds and 23 seconds for Scenario 1, Scenario 2, Scenario 4, and Scenario 5 whereas approximately 14 seconds for Scenario 3 and Scenario 6. On the other hand, GH spends at most 0.16 seconds for an instance. However, although GH can solve the problem in less computational time, TSH provides slightly better results compared to GH. For example, in Scenario 1 and Scenario 4, TSH finds better solutions in 22 instances whereas GH is only better in one instance. Both heuristics have the same solution for

five instances. A similar observation can be made for Scenario 2 as seen in Table 5.4, GH finds a better solution for one instance but TSH is better in 14 instances and same as GH in 13 instances. It is noticeable that GH cannot find a better objective value than TSH for Scenario 3, Scenario 5, and Scenario 6, whereas TSH obtains better solutions in these scenarios. On the other hand, due to the fact that the objective values are zero in 26 instances, it can be said that TSH finds an optimal solution for these instances.

In addition to three runs, TSH has been run ten times for better resulting instances of GH. The results are indicated in Table 5.9. It is seen that TSH finds the same objective values with GH in Scenario 1 and Scenario 4. On the other hand, TSH can obtain a better solution in Scenario 2.

Finally, interdiction decisions on 10-node and 20-node networks are presented for Scenario 1 in Table A.2, for Scenario 2 in Table A.3, for Scenario 3 in Table A.4, for Scenario 4 in Table A.5, for Scenario 5 in Table A.6, and for Scenario 6 in Table A.7. When we compare Scenario 1 with Scenario 4, and Scenario 2 with Scenario 5, it is noticed that the amount of flow demand between SD pairs is important for the interdiction decision of the attacker. Therefore, the most frequently used nodes are usually interdicted by the attacker as they have high flow demands. It is also observed that the proposed TSH obtains the best solutions for periodic interdiction decisions in Scenario 3 and Scenario 6. Thus, it can be said that in case of unit node interdiction cost, the attacker mostly chooses early periods to cause more damage in the network and does not attack during recovery periods. However, there is no such observation in case of differentiated node interdiction costs because of the resource budget restriction. Hence, it is seen that the attacker tends to attack the nodes with low interdiction costs in Scenario 1, Scenario 2, Scenario 4, and Scenario 5.

Table 5.3. TSH and GH comparison for Scenario 1

No. of nodes	**Arc density**	**No. of SD pairs**	**GH**		**TSH**			**Better algorithm**
			$\mathbf{z_{GH}}$	**CPU**	$\mathbf{z_{TS}^{best}}$	$\mathbf{z_{TS}^{avg}}$	**CPU**	
10	30%	3	400	0.13	400	400	3.35	-
10	30%	5	700	0.01	500	500	3.21	Tabu
10	44%	3	400	0.02	0	0	3.48	Tabu
10	44%	5	1200	0.02	900	900	3.18	Tabu
20	30%	3	300	0.02	100	100	9.51	Tabu
20	30%	5	500	0.02	300	367	8.79	Tabu
20	44%	7	1000	0.02	800	933	11.50	Tabu
20	44%	3	100	0.02	100	100	8.28	-
20	30%	5	100	0.02	100	100	9.40	-
20	30%	7	800	0.02	700	700	8.40	Tabu
30	44%	3	1100	0.05	600	600	17.71	Tabu
30	44%	5	2300	0.02	1800	1833	14.29	Tabu
30	30%	7	3300	0.03	2700	2767	14.96	Tabu
30	30%	3	700	0.02	300	300	13.32	Tabu
30	44%	5	1600	0.02	1400	1400	18.13	Tabu
30	44%	7	2700	0.02	2500	2500	17.72	Tabu
40	30%	3	100	0.02	400	400	23.52	Greedy
40	30%	5	1700	0.03	1300	1300	26.72	Tabu
40	44%	7	3500	0.02	2800	2933	27.62	Tabu
40	44%	3	300	0.02	100	100	26.01	Tabu
40	30%	5	2200	0.02	1500	1500	27.49	Tabu
40	30%	7	3600	0.02	2700	2833	31.90	Tabu
50	44%	3	1100	0.02	500	567	46.31	Tabu
50	44%	5	1900	0.02	1400	1400	46.53	Tabu
50	30%	7	2600	0.02	2000	2033	50.98	Tabu
50	30%	3	300	0.02	300	300	39.29	-
50	44%	5	1100	0.02	1100	1133	47.80	-
50	44%	7	1700	0.02	1500	1633	40.77	Tabu
Average				0.03			21.43	

Table 5.4. TSH and GH comparison for Scenario 2

No. of nodes	**Arc density**	**No. of SD pairs**	**GH**		**TSH**			**Better algorithm**
			$\mathbf{z_{GH}}$	**CPU**	$\mathbf{z_{TS}^{best}}$	$\mathbf{z_{TS}^{avg}}$	**CPU**	
10	30%	3	0	0.13	0	0	2.45	-
10	30%	5	0	0.00	0	0	2.56	-
10	44%	3	0	0.02	0	0	2.84	-
10	44%	5	900	0.02	900	900	2.76	-
20	30%	3	900	0.02	300	300	10.52	Tabu
20	30%	5	900	0.02	300	567	10.21	Tabu
20	44%	7	900	0.02	900	900	9.63	-
20	44%	3	300	0.02	300	300	10.62	-
20	30%	5	300	0.02	300	300	11.81	-
20	30%	7	900	0.02	500	500	12.56	Tabu
30	44%	3	1100	0.03	900	900	20.52	Tabu
30	44%	5	1100	0.02	900	967	20.59	Tabu
30	30%	7	1900	0.02	1600	1600	21.03	Tabu
30	30%	3	1100	0.02	500	500	17.06	Tabu
30	44%	5	1100	0.02	900	1033	22.48	Tabu
30	44%	7	2300	0.02	1900	1900	22.49	Tabu
40	30%	3	400	0.04	400	400	28.68	-
40	30%	5	1100	0.02	1100	1167	27.56	-
40	44%	7	2900	0.03	2900	2900	30.27	-
40	44%	3	500	0.02	500	500	26.06	-
40	30%	5	1100	0.02	1200	1200	24.77	Greedy
40	30%	7	2900	0.02	2700	2900	24.90	Tabu
50	44%	3	0	0.02	0	0	40.81	-
50	44%	5	1000	0.02	300	300	46.54	Tabu
50	30%	7	2200	0.02	500	500	43.80	Tabu
50	30%	3	0	0.02	0	200	48.59	-
50	44%	5	1800	0.02	1100	1167	43.12	Tabu
50	44%	7	2500	0.02	1300	1300	56.45	Tabu
Average				0.03			22.92	

Table 5.5. TSH and GH comparison for Scenario 3

No. of nodes	**Arc density**	**No. of SD pairs**	**GH**		**TSH**			**Better algorithm**
			$\mathbf{z_{GH}}$	**CPU**	$\mathbf{z_{TS}^{best}}$	$\mathbf{z_{TS}^{avg}}$	**CPU**	
10	30%	3	0	0.16	0	0	3.07	-
10	30%	5	0	0.00	0	0	2.34	-
10	44%	3	0	0.02	0	0	2.98	-
10	44%	5	900	0.02	700	700	2.04	Tabu
20	30%	3	0	0.02	0	0	6.28	-
20	30%	5	1100	0.02	1100	1100	6.02	-
20	44%	7	2200	0.02	1900	1900	5.83	Tabu
20	44%	3	0	0.01	0	0	5.39	-
20	30%	5	900	0.02	900	900	5.94	-
20	30%	7	1400	0.02	1400	1400	6.80	-
30	44%	3	1500	0.03	700	700	10.09	Tabu
30	44%	5	2400	0.02	1600	1600	12.36	Tabu
30	30%	7	3700	0.02	2800	2800	14.31	Tabu
30	30%	3	0	0.01	0	0	11.17	-
30	44%	5	1500	0.02	1100	1100	10.54	Tabu
30	44%	7	3100	0.02	2800	2800	13.12	Tabu
40	30%	3	800	0.02	700	700	16.19	Tabu
40	30%	5	2800	0.02	1700	1700	18.80	Tabu
40	44%	7	3900	0.02	3500	3500	18.87	Tabu
40	44%	3	0	0.02	0	0	20.30	-
40	30%	5	900	0.02	900	900	18.77	-
40	30%	7	2700	0.02	2600	2600	19.35	Tabu
50	44%	3	700	0.02	700	700	24.35	-
50	44%	5	2800	0.02	2400	2400	24.60	Tabu
50	30%	7	3900	0.02	2800	2867	29.21	Tabu
50	30%	3	900	0.02	700	700	23.73	Tabu
50	44%	5	2700	0.01	2400	2400	24.02	Tabu
50	44%	7	2900	0.01	2900	2900	32.37	-
Average				0.02			13.89	

Table 5.6. TSH and GH comparison for Scenario 4

No. of nodes	**Arc density**	**No. of SD pairs**	**GH**		**TSH**			**Better algorithm**
			$\mathbf{z_{GH}}$	**CPU**	$\mathbf{z_{TS}^{best}}$	$\mathbf{z_{TS}^{avg}}$	**CPU**	
10	30%	3	500	0.16	500	500	3.31	-
10	30%	5	1000	0.01	600	600	3.22	Tabu
10	44%	3	500	0.03	0	0	3.14	Tabu
10	44%	5	1300	0.02	900	967	3.28	Tabu
20	30%	3	800	0.02	300	300	9.45	Tabu
20	30%	5	800	0.02	600	600	8.67	Tabu
20	44%	7	1100	0.02	800	800	12.60	Tabu
20	44%	3	300	0.02	300	300	9.12	-
20	30%	5	300	0.02	400	400	9.35	Greedy
20	30%	7	600	0.02	600	600	9.22	-
30	44%	3	1100	0.03	500	500	20.99	Tabu
30	44%	5	2100	0.02	1700	1700	16.06	Tabu
30	30%	7	2900	0.02	2300	2433	15.66	Tabu
30	30%	3	400	0.03	200	200	13.94	Tabu
30	44%	5	1600	0.02	1200	1200	19.96	Tabu
30	44%	7	2600	0.02	2000	2000	20.17	Tabu
40	30%	3	400	0.02	300	367	24.39	Tabu
40	30%	5	1200	0.03	1200	1367	28.61	-
40	44%	7	2600	0.02	2400	2600	26.87	Tabu
40	44%	3	300	0.03	300	300	30.34	-
40	30%	5	1500	0.02	1100	1433	26.52	Tabu
40	30%	7	2700	0.02	2200	2200	28.70	Tabu
50	44%	3	1200	0.02	600	600	45.24	Tabu
50	44%	5	2400	0.02	1400	1400	45.61	Tabu
50	30%	7	2400	0.02	2000	2000	43.27	Tabu
50	30%	3	500	0.02	400	400	56.09	Tabu
50	44%	5	1600	0.02	1200	1267	55.90	Tabu
50	44%	7	2000	0.02	1800	1800	42.44	Tabu
Average				0.03			22.58	

Table 5.7. TSH and GH comparison for Scenario 5

No. of nodes	**Arc density**	**No. of SD pairs**	**GH**		**TSH**			**Better algorithm**
			$\mathbf{z_{GH}}$	**CPU**	$\mathbf{z_{TS}^{best}}$	$\mathbf{z_{TS}^{avg}}$	**CPU**	
10	30%	3	0	0.15	0	0	2.42	-
10	30%	5	0	0.00	0	0	2.41	-
10	44%	3	0	0.01	0	0	2.92	-
10	44%	5	1200	0.02	1200	1200	2.83	-
20	30%	3	1200	0.02	600	600	10.49	Tabu
20	30%	5	1200	0.02	600	867	10.16	Tabu
20	44%	7	1200	0.02	1200	1200	9.63	-
20	44%	3	600	0.02	600	600	10.13	-
20	30%	5	600	0.02	600	600	10.51	-
20	30%	7	1200	0.02	800	800	11.73	Tabu
30	44%	3	800	0.03	800	800	14.59	-
30	44%	5	800	0.02	800	800	19.17	-
30	30%	7	1200	0.02	1200	1200	24.26	-
30	30%	3	800	0.02	400	400	17.22	Tabu
30	44%	5	800	0.03	800	800	19.28	-
30	44%	7	1200	0.02	1200	1200	20.07	-
40	30%	3	400	0.02	400	400	20.43	-
40	30%	5	800	0.02	800	800	27.29	-
40	44%	7	2000	0.03	2000	2000	27.68	-
40	44%	3	400	0.02	400	400	28.85	-
40	30%	5	800	0.02	800	1067	24.98	-
40	30%	7	2000	0.02	2000	2267	25.46	-
50	44%	3	0	0.02	0	0	42.76	-
50	44%	5	400	0.02	200	200	48.76	Tabu
50	30%	7	800	0.02	400	400	47.32	Tabu
50	30%	3	0	0.02	0	0	61.97	-
50	44%	5	1400	0.02	800	800	43.53	Tabu
50	44%	7	1800	0.03	1400	1400	48.76	Tabu
Average				0.02			22.70	

Table 5.8. TSH and GH comparison for Scenario 6

No. of nodes	**Arc density**	**No. of SD pairs**	**GH**		**TSH**			**Better algorithm**
			$\mathbf{z_{GH}}$	**CPU**	$\mathbf{z_{TS}^{best}}$	$\mathbf{z_{TS}^{avg}}$	**CPU**	
10	30%	3	0	0.14	0	0	3.07	-
10	30%	5	0	0.00	0	0	2.12	-
10	44%	3	0	0.02	0	0	2.98	-
10	44%	5	1200	0.02	400	400	2.26	Tabu
20	30%	3	0	0.01	0	0	6.79	-
20	30%	5	800	0.02	800	800	6.19	-
20	44%	7	2000	0.03	1200	1200	5.45	Tabu
20	44%	3	0	0.02	0	0	6.06	-
20	30%	5	400	0.02	400	400	6.69	-
20	30%	7	800	0.02	800	800	5.64	-
30	44%	3	800	0.03	800	800	9.85	-
30	44%	5	2000	0.02	2000	2000	10.06	-
30	30%	7	2400	0.03	2400	2533	11.79	-
30	30%	3	0	0.02	0	0	12.29	-
30	44%	5	800	0.02	800	800	10.81	-
30	44%	7	2800	0.02	2000	2000	12.04	Tabu
40	30%	3	800	0.03	800	800	16.28	-
40	30%	5	2800	0.02	2000	2000	18.72	Tabu
40	44%	7	4000	0.02	3200	3200	17.34	Tabu
40	44%	3	0	0.02	0	0	21.21	-
40	30%	5	1200	0.02	800	800	19.10	Tabu
40	30%	7	2400	0.02	2000	2000	19.12	Tabu
50	44%	3	1200	0.02	800	800	24.73	Tabu
50	44%	5	4000	0.02	2400	2400	24.21	Tabu
50	30%	7	4800	0.02	3400	3400	28.47	Tabu
50	30%	3	800	0.02	800	800	23.94	-
50	44%	5	2400	0.02	2400	2400	24.10	-
50	44%	7	3200	0.02	3200	3267	40.60	-
Average				0.03			14.00	

Table 5.9. TSH results

Scenario	**No. of nodes**	**Arc density**	**No. of SD pairs**	$\mathbf{z_{TS}^{best}}$	$\mathbf{z_{TS}^{avg}}$	**CPU**
Scenario 1	40	30%	3	100	280	27.71
Scenario 2	40	44%	5	900	1050	25.95
Scenario 4	20	44%	5	300	390	9.43

6. CONCLUSION

In this study, a bilevel multi-period network interdiction problem considering recovery periods has been analyzed. A mixed-integer linear mathematical model is formulated to solve an attacker-operator problem in the transportation network. In this model, the attacker in the upper level aims to minimize the amount of flow in the network by interdicting some nodes. On the contrary, the operator in the lower level aims to maximize the amount of flow in the network after the attacks are realized. Flow demand from a source node is transferred to a destination node using admissible paths. The admissible paths are determined by the shortest path models.

A tabu search heuristic is developed to solve the proposed model and it is tested on several networks. The tabu search algorithm first choose some of the attacker's binary decision variables to fix in the lower level, then it exactly solves the lower level problem for the fixed decision variables. The tabu search heuristic starts with a feasible solution generated by an initial model aiming to find the maximum number of node interdiction in the network and employs three particular move operators during the search.

Instances are generated under various conditions assuming different arc densities, different network sizes and different number of source-destination pairs in the networks. These instances are examined for several scenarios to observe the change in the network's behavior and the performance of tabu search heuristic in these varied scenarios. The tabu search algorithm is run three times and the minimum objective values are reported as the best objective values for each instance.

Complete enumeration and greedy methods have been used to assess the performance of the tabu search heuristic. Eight instances are used to compare the tabu search heuristic with complete enumeration method. Their testings are limited to small sized networks because it is not possible to apply complete enumeration method on large sized networks or medium sized networks with differentiated node interdiction

costs due to the use of an exhaustive search. Important outcomes observed from the tabu search heuristic and complete enumeration method comparison are that the tabu search heuristic is capable of finding the optimal solutions for all instances and the CPU times of the tabu search heuristic are clearly less than the CPU times of complete enumeration method. On the other hand, six different scenarios are considered in the tabu search heuristic and greedy heuristic comparison. Each scenario is tested on 28 instances. Therefore, 168 instances are used to compare the results of tabu search heuristic and greedy heuristic. It is noticed that tabu search heuristic has better solutions for 91 instances whereas greedy heuristic is better in three instances. Both algorithms find the same solution for 74 instances. The tabu search heuristic also obtains an optimal solution in 26 out of 168 instances. Therefore, it is obvious that tabu search heuristic is an appropriate solution method for the proposed model. In addition, the most common nodes between source-destination pairs are usually interdicted by the attacker. The attacker makes his/her interdiction decision based on the amount of flow demand between source-destination pairs and low interdiction costs. When there is unit node interdiction cost, interdiction decisions are usually made in early periods and recovery periods are not preferred to make an attack.

In our study, we use an integer constant for the recovery period of a node. However, further analyses can be done by considering different approaches for the recovery period. The planning horizon can also be increased to achieve a long-term strategic perspective. Furthermore, the addressed problem can be extended by adding either partial interdiction or arc interdiction decisions.

REFERENCES

1. Grubesic, T. H. and A. T. Murray, "Vital nodes, interconnected infrastructures, and the geographies of network survivability", *Annals of the Association of American Geographers*, Vol. 96, No. 1, pp. 64–83, 2006.

2. Buesa, M., A. Valiño, J. Heijs, T. Baumert and J. G. Gómez, "The economic cost of March 11: Measuring the direct economic cost of the terrorist attack on March 11, 2004 in Madrid", *Terrorism and Political Violence*, Vol. 19, No. 4, pp. 489–509, 2007.

3. Jordán, F., "Predicting target selection by terrorists: a network analysis of the 2005 London underground attacks", *Int. J. Critical Infrastructures J. Critical Infrastructures*, Vol. 4, No. 12, pp. 206–214, 2008, `http://www.colbud.hu/fjordan/ijcis.pdf`.

4. Smith, J. C., "Basic Interdiction Models", *Wiley Encyclopedia of Operations Research and Management Science*, Wiley, Newyork, 2010, `http://doi.wiley.com/10.1002/9780470400531.eorms0089`.

5. von Stackelberg, H., *The theory of the market economy*, Oxford University Press, New York, 1952.

6. Colson, B., P. Marcotte and G. Savard, "An overview of bilevel optimization", *Annals of Operations Research*, Vol. 153, No. 1, pp. 235–256, 2007.

7. Wollmer, R., "Removing Arcs from a Network", *Operations Research*, Vol. 12, No. 6, pp. 934–940, 1964, `https://www.jstor.org/stable/168177`.

8. Wood, R. K., "Deterministic network interdiction", *Mathematical and Computer Modelling*, Vol. 17, No. 2, pp. 1–18, 1993.

9. Altner, D. S., Ö. Ergun and N. A. Uhan, "The Maximum Flow Network Interdiction Problem: Valid inequalities, integrality gaps, and approximability", *Operations Research Letters*, Vol. 38, No. 1, pp. 33–38, 2010, http://dx.doi.org/10.1016/j.orl.2009.09.013.

10. Cormican, K. J., D. P. Morton and R. K. Wood, "Stochastic Network Interdiction", *Operations Research*, Vol. 46, No. 2, pp. 184–197, 1998, https://www.jstor.org/stable/222859.

11. Janjarassuk, U. and J. Linderoth, "Reformulation and sampling to solve a stochastic network interdiction problem", *Networks*, Vol. 52, No. 3, pp. 120–132, 2008, http://doi.wiley.com/10.1002/net.20237.

12. Ramirez-Marquez, J. E. and C. M. Rocco S., "Stochastic network interdiction optimization via capacitated network reliability modeling and probabilistic solution discovery", *Reliability Engineering and System Safety*, Vol. 94, No. 5, pp. 913–921, 2009.

13. Fulkerson, D. and G. C. Harding, "Maximizing the Minimum Source-Sink Path Subject to A Budget Constraint", *Mathematical Programming*, Vol. 13, pp. 116–118, 1977.

14. Israeli, E. and R. K. Wood, "Shortest-Path Network Interdiction", *Networks*, Vol. 40, No. 2, pp. 97–111, 2002.

15. Bayrak, H. and M. D. Bailey, "Shortest Path Network Interdiction with Asymmetric Information", *Networks*, Vol. 52, No. 3, pp. 133–140, 2008.

16. Khachiyan, L., E. Boros, K. Borys, K. Elbassioni, V. Gurvich, G. Rudolf and J. Zhao, "On short paths interdiction problems: Total and node-wise limited interdiction", *Theory of Computing Systems*, Vol. 43, No. 2, pp. 204–233, 2008.

17. Lim, C. and J. C. Smith, "Algorithms for discrete and continuous multicommo-

dity flow network interdiction problems", *IIE Transactions (Institute of Industrial Engineers)*, Vol. 39, No. 1, pp. 15–26, 2007.

18. Zhang, J., J. Zhuang and B. Behlendorf, "Stochastic shortest path network interdiction with a case study of Arizona–Mexico border", *Reliability Engineering and System Safety*, Vol. 179, No. November 2017, pp. 62–73, 2018, https://doi.org/10.1016/j.ress.2017.10.026.

19. Held, H. and D. L. Woodruff, "Heuristics for multi-stage interdiction of stochastic networks", *Journal of Heuristics*, Vol. 11, No. 5-6 SPEC. ISS., pp. 483–500, 2005.

20. Held, H., R. Hemmecke and D. L. Woodruff, "A decomposition algorithm applied to planning the interdiction of stochastic networks", *Naval Research Logistics*, Vol. 52, No. 4, pp. 321–328, 2005.

21. Yates, J. and K. Lakshmanan, "A constrained binary knapsack approximation for shortest path network interdiction", *Computers and Industrial Engineering*, Vol. 61, No. 4, pp. 981–992, 2011, http://dx.doi.org/10.1016/j.cie.2011.06.011.

22. Lunday, B. J. and H. D. Sherali, "Minimizing the maximum network flow: Models and algorithms with resource synergy considerations", *Journal of the Operational Research Society*, Vol. 63, No. 12, pp. 1693–1707, 2012.

23. Bertsimas, D., E. Nasrabadi and J. B. Orlin, "On the power of randomization in network interdiction", *Operations Research Letters*, Vol. 44, No. 1, pp. 114–120, 2016.

24. Shen, S., J. C. Smith and R. Goli, "Exact interdiction models and algorithms for disconnecting networks via node deletions", *Discrete Optimization*, Vol. 9, No. 3, pp. 172–188, 2012, http://dx.doi.org/10.1016/j.disopt.2012.07.001.

25. Zenklusen, R., "Matching interdiction", *Discrete Applied Mathematics*, Vol. 158,

No. 15, pp. 1676–1690, 2010, http://dx.doi.org/10.1016/j.dam.2010.06.006.

26. Akgün, İ., B. Tansel and R. K. Wood, "The multi-terminal maximum-flow network-interdiction problem", *European Journal of Operational Research*, Vol. 211, No. 2, pp. 241–251, 2011.

27. Hausken, K. and J. Zhuang, "Defending against a stockpiling terrorist", *Engineering Economist*, Vol. 56, No. 4, pp. 321–353, 2011.

28. Shan, X. and J. Zhuang, "Hybrid defensive resource allocations in the face of partially strategic attackers in a sequential defender-Attacker game", *European Journal of Operational Research*, Vol. 228, No. 1, pp. 262–272, 2013, http://dx.doi.org/10.1016/j.ejor.2013.01.029.

29. Bell, M. G., U. Kanturska, J. D. Schmocker and A. Fonzone, "Attacker-defender models and road network vulnerability", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 366, No. 1872, pp. 1893–1906, 2008.

30. Royset, J. O. and R. K. Wood, "Solving the bi-objeetive maximum-flow network-interdiction problem", *INFORMS Journal on Computing*, Vol. 19, No. 2, pp. 175–184, 2007.

31. Rocco S., C. M. and J. E. Ramirez-Marquez, "A bi-objective approach for shortest-path network interdiction", *Computers and Industrial Engineering*, Vol. 59, No. 2, pp. 232–240, 2010, http://dx.doi.org/10.1016/j.cie.2010.04.004.

32. Kheirkhah, A., H. Navidi and M. M. Bidgoli, "A bi-level network interdiction model for solving the hazmat routing problem", *International Journal of Production Research*, Vol. 54, No. 2, pp. 459–471, 2016, http://dx.doi.org/10.1080/00207543.2015.1084061.

33. Legillon, F., A. Liefooghe and E.-G. Talbi, "CoBRA: A cooperative coevolutionary

algorithm for bi-level optimization", *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, 2012, http://ieeexplore.ieee.org/document/6256620/.

34. Salmeron, J., K. Wood and R. Baldick, "Analysis of electric grid security under terrorist threat", *IEEE Transactions on Power Systems*, Vol. 19, No. 2, pp. 905–912, 2004.

35. Arroyo, J. M. and F. D. Galiana, "On the solution of the bilevel programming formulation of the terrorist threat problem", *IEEE Transactions on Power Systems*, Vol. 20, No. 2, pp. 789–797, 2005.

36. Enayaty-Ahangar, F., C. E. Rainwater and T. C. Sharkey, "A Logic-based Decomposition Approach for Multi-Period Network Interdiction Models", *Omega (United Kingdom)*, Vol. 0, pp. 1–15, 2018, https://doi.org/10.1016/j.omega.2018.08.006.

37. Afshari Rad, M. and H. T. Kakhki, "Maximum dynamic network flow interdiction problem: New formulation and solution procedures", *Computers and Industrial Engineering*, Vol. 65, No. 4, pp. 531–536, 2013, http://dx.doi.org/10.1016/j.cie.2013.04.014.

38. Ford, L. and D. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

39. Church, R. L., M. P. Scaparra and R. S. Middleton, "Identifying critical infrastructure: The median and covering facility interdiction problems", *Annals of the Association of American Geographers*, Vol. 94, No. 3, pp. 491–502, 2004.

40. Murray, A. T., T. C. Matisziw and T. H. Grubesic, "Critical network infrastructure analysis: Interdiction and system flow", *Journal of Geographical Systems*, Vol. 9, No. 2, pp. 103–117, 2007.

41. Snyder, L. V., M. P. Scaparra, M. S. Daskin and R. L. Church, "Planning for Disruptions in Supply Chain Networks", *Models, Methods, and*

Applications for Innovative Decision Making, pp. 234–257, INFORMS, 2006, http://pubsonline.informs.org/doi/abs/10.1287/educ.1063.0025.

42. Church, R. L. and M. P. Scaparra, "Protecting critical assets: The r-interdiction median problem with fortification", *Geographical Analysis*, Vol. 39, No. 2, pp. 129–146, 2007.

43. Scaparra, M. P. and R. L. Church, "A bilevel mixed-integer program for critical infrastructure protection planning", *Computers and Operations Research*, Vol. 35, No. 6, pp. 1905–1923, 2008, http://linkinghub.elsevier.com/retrieve/pii/S0305054806002395.

44. Scaparra, M. P. and R. Church, "Protecting Supply Systems to Mitigate Potential Disaster: A Model to Fortify Capacitated Facilities", *International Regional Science Review*, Vol. 35, No. 2, pp. 188–210, 2012.

45. Smith, J. C. and C. Lim, "Algorithms for Network Interdiction and Fortification Games", *Pareto Optimality, Game Theory And Equilibria*, pp. 609–644, Springer, 2008, http://link.springer.com/10.1007/978-0-387-77247-9_24.

46. Aksen, D., N. Piyade and N. Aras, "The budget constrained r-interdiction median problem with capacity expansion", *Central European Journal of Operations Research*, Vol. 18, No. 3, pp. 269–291, 2010.

47. Liberatore, F., M. P. Scaparra and M. S. Daskin, "Analysis of facility protection strategies against an uncertain number of attacks: The stochastic R-interdiction median problem with fortification", *Computers and Operations Research*, Vol. 38, No. 1, pp. 357–366, 2011, https://linkinghub.elsevier.com/retrieve/pii/S0305054810001231.

48. Losada, C., M. P. Scaparra and J. R. O'Hanley, "Optimizing system resilience: A facility protection model with recovery time", *European Journal of Operational Research*, Vol. 217, No. 3, pp. 519–530, 2012,

http://dx.doi.org/10.1016/j.ejor.2011.09.044.

49. Cappanera, P. and M. P. Scaparra, "Optimal Allocation of Protective Resources in Shortest-Path Networks", *Transportation Science*, Vol. 45, No. 1, pp. 64–80, 2011, http://pubsonline.informs.org/doi/abs/10.1287/trsc.1100.0340.

50. Alguacil, N., A. Delgadillo and J. M. Arroyo, "A trilevel programming approach for electric grid defense planning", *Computers and Operations Research*, Vol. 41, No. 1, pp. 282–290, 2014.

51. Sarhadi, H., D. M. Tulett and M. Verma, "A defender-attacker-defender approach to the optimal fortification of a rail intermodal terminal network", *Journal of Transportation Security*, Vol. 8, No. 1-2, pp. 17–32, 2015.

52. Jin, J. G., L. Lu, L. Sun and J. Yin, "Optimal allocation of protective resources in urban rail transit networks against intentional attacks", *Transportation Research Part E: Logistics and Transportation Review*, Vol. 84, pp. 73–87, 2015, http://dx.doi.org/10.1016/j.tre.2015.10.008.

53. Sadeghi, S., A. Seifi and E. Azizi, "Trilevel shortest path network interdiction with partial fortification", *Computers and Industrial Engineering*, Vol. 106, pp. 400–411, 2017, http://dx.doi.org/10.1016/j.cie.2017.02.006.

54. Malaviya, A., C. Rainwater and T. Sharkey, "Multi-period network interdiction problems with applications to city-level drug enforcement", *IIE Transactions (Institute of Industrial Engineers)*, Vol. 44, No. 5, pp. 368–380, 2012.

55. Starita, S. and M. P. Scaparra, "Optimizing dynamic investment decisions for railway systems protection", *European Journal of Operational Research*, Vol. 248, No. 2, pp. 543–557, 2016, https://linkinghub.elsevier.com/retrieve/pii/S037722171500658X.

56. Yen, J. Y., "Finding the K Shortest Loopless Paths in a Network", *Management*

Science, Vol. 17, No. 11, pp. 712–716, 1971.

57. Eppstein, D., "Finding the k Shortest Paths", *SIAM Journal on Computing*, Vol. 28, No. 2, pp. 652–673, 1998, http://epubs.siam.org/doi/10.1137/S0097539795290477.

58. Miller, C. E., A. W. Tucker and R. A. Zemlin, "Integer Programming Formulation of Traveling Salesman Problems", *Journal of Association for Computing Machinery*, Vol. 7, No. 4, pp. 326–329, 1960, http://portal.acm.org/citation.cfm?doid=321043.321046.

59. Jeroslow, R. G., "The polynomial hierarchy and a simple model for competitive analysis", *Mathematical Programming*, Vol. 32, No. 2, pp. 146–164, 1985.

60. Bard, J. F., "Some properties of the bilevel programming problem", *Journal of Optimization Theory and Applications*, Vol. 68, No. 2, pp. 371–378, 1991, http://link.springer.com/10.1007/BF00941574.

61. Glover, F., "Future paths for integer programming and links to artificial intelligence", *Computers and Operations Research*, Vol. 13, No. 5, pp. 533–549, 1986, http://linkinghub.elsevier.com/retrieve/pii/0305054886900481.

APPENDIX A: RESULTS

Table A.1 Admissible paths

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
10	44%	3	4-1	4	2	1			
				4	2	9	1		
			6-2	6	4	2			
				6	9	2			
			10-9	10	2	9			
				10	3	2	9		
10	44%	5	4-1	4	2	1			
				4	2	9	1		
			6-2	6	4	2			
				6	9	2			
			10-9	10	2	9			
				10	3	2	9		
			3-8	3	1	8			
				3	1	5	8		
			7-5	7	1	5			
				7	8	5			
10	30%	3	4-1	4	9	1			
				4	9	2	3	1	
			6-2	6	9	2			
				6	9	1	3	2	
			10-9	10	3	2	9		
				10	3	1	9		
10	30%	5	4-1	4	9	1			
				4	9	2	3	1	

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
			6-2	6	9	2			
				6	9	1	3	2	
			10-9	10	3	2	9		
				10	3	1	9		
			3-8	3	2	5	8		
				3	1	9	2	5	8
			7-5	7	9	2	5		
				7	1	9	2	5	
15	44%	3	4-1	4	7	13	1		
				4	12	1			
			6-2	6	3	4	7	2	
				6	3	11	2		
			10-9	10	6	9			
				10	12	1	9		
15	44%	5	4-1	4	7	13	1		
				4	12	1			
			6-2	6	3	4	7	2	
				6	3	11	2		
			10-9	10	6	9			
				10	12	1	9		
			13-15	13	7	11	15		
				13	7	4	15		
			7-5	7	10	5			
				7	4	12	10	5	
15	30%	3	4-1	4	7	13	1		
				4	7	8	12	1	
			6-2	6	3	8	2		

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
				6	10	8	2		
			10-9	10	8	12	1	9	
				10	7	9			
15	30%	5	4-1	4	7	13	1		
				4	7	8	12	1	
			6-2	6	3	8	2		
				6	10	8	2		
			10-9	10	8	12	1	9	
				10	7	9			
			13-15	13	7	4	15		
				13	12	11	15		
			7-5	7	13	5			
				7	13	1	5		
20	44%	3	4-1	4	11	1			
				4	20	11	1		
			5-8	5	19	16	8		
				5	17	12	16	8	
			10-7	10	4	7			
				10	16	4	7		
20	44%	5	4-1	4	11	1			
				4	20	11	1		
			5-8	5	19	16	8		
				5	17	12	16	8	
			10-7	10	4	7			
				10	16	4	7		
			15-18	15	1	18			
				15	1	11	18		

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
			19-12	19	16	12			
				19	16	10	12		
20	44%	7	4-1	4	11	1			
				4	20	11	1		
			5-8	5	19	16	8		
				5	17	12	16	8	
			10-7	10	4	7			
				10	16	4	7		
			15-18	15	1	18			
				15	1	11	18		
			19-12	19	16	12			
				19	16	10	12		
			20-16	20	11	18	16		
				20	4	16			
			13-17	13	15	12	17		
				13	16	12	17		
20	30%	3	4-1	4	11	1			
				4	7	2	1		
			5-8	5	19	16	8		
				5	6	20	8		
			10-7	10	4	7			
				10	12	4	7		
20	30%	5	4-1	4	11	1			
				4	7	2	1		
			5-8	5	19	16	8		
				5	6	20	8		
			10-7	10	4	7			

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
				10	12	4	7		
			15-18	15	12	16	18		
				15	13	16	18		
			19-12	19	16	12			
				19	16	10	12		
20	30%	7	4-1	4	11	1			
				4	7	2	1		
			5-8	5	19	16	8		
				5	6	20	8		
			10-7	10	4	7			
				10	12	4	7		
			15-18	15	12	16	18		
				15	13	16	18		
			19-12	19	16	12			
				19	16	10	12		
			20-16	20	8	16			
				20	12	16			
			13-17	13	15	12	17		
				13	16	12	17		
30	44%	3	4-1	4	20	1			
				4	11	18	1		
			5-8	5	27	8			
				5	29	8			
			7-26	7	4	20	1	26	
				7	4	20	16	26	
30	44%	5	4-1	4	20	1			
				4	11	18	1		

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
			5-8	5	27	8			
				5	29	8			
			7-26	7	4	20	1	26	
				7	4	20	16	26	
			10-11	10	12	16	11		
				10	19	17	12	16	11
			19-12	19	17	12			
				19	10	12			
30	44%	7	4-1	4	20	1			
				4	11	18	1		
			5-8	5	27	8			
				5	29	8			
			7-26	7	4	20	1	26	
				7	4	20	16	26	
			10-11	10	12	16	11		
				10	19	17	12	16	11
			19-12	19	17	12			
				19	10	12			
			20-25	20	8	25			
				20	16	25			
			17-29	17	27	29			
				17	12	27	29		
30	30%	3	4-1	4	20	1			
				4	11	15	1		
			5-8	5	9	25	8		
				5	16	8			
			7-26	7	11	15	26		

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
				7	11	16	26		
30	30%	5	4-1	4	20	1			
				4	11	15	1		
			5-8	5	9	25	8		
				5	16	8			
			7-26	7	11	15	26		
				7	11	16	26		
			10-11	10	12	16	11		
				10	19	17	12	16	11
			19-12	19	17	12			
				19	10	12			
30	30%	7	4-1	4	20	1			
				4	11	15	1		
			5-8	5	9	25	8		
				5	16	8			
			7-26	7	11	15	26		
				7	11	16	26		
			10-11	10	12	16	11		
				10	19	17	12	16	11
			19-12	19	17	12			
				19	10	12			
			20-25	20	9	25			
				20	19	25			
			17-29	17	27	29			
				17	12	27	29		
40	44%	3	4-1	4	18	1			
				4	7	18	1		

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
			5-8	5	37	8			
				5	25	37	8		
			7-11	7	18	11			
				7	4	18	11		
40	44%	5	4-1	4	18	1			
				4	7	18	1		
			5-8	5	37	8			
				5	25	37	8		
			7-11	7	18	11			
				7	4	18	11		
			15-18	15	1	18			
				15	1	23	11	18	
			19-14	19	29	14			
				19	8	14			
40	44%	7	4-1	4	18	1			
				4	7	18	1		
			5-8	5	37	8			
				5	25	37	8		
			7-11	7	18	11			
				7	4	18	11		
			15-18	15	1	18			
				15	1	23	11	18	
			19-14	19	29	14			
				19	8	14			
			20-16	20	38	16			
				20	33	16			
			13-31	13	35	32	31		

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
				13	17	31			
40	30%	3	4-1	4	18	1			
				4	7	18	1		
			5-8	5	37	8			
				5	34	8			
			7-11	7	24	11			
				7	4	24	11		
40	30%	5	4-1	4	18	1			
				4	7	18	1		
			5-8	5	37	8			
				5	34	8			
			7-11	7	24	11			
				7	4	24	11		
			15-18	15	1	18			
				15	4	18			
			19-14	19	29	14			
				19	8	14			
40	30%	7	4-1	4	18	1			
				4	7	18	1		
			5-8	5	37	8			
				5	34	8			
			7-11	7	24	11			
				7	4	24	11		
			15-18	15	1	18			
				15	4	18			
			19-14	19	29	14			
				19	8	14			

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
			20-16	20	33	16			
				20	6	16			
			13-31	13	35	18	31		
				13	17	12	32	31	
50	44%	3	5-1	5	9	23	1		
				5	41	23	1		
			8-22	8	16	20	22		
				8	18	22			
			7-26	7	18	23	26		
				7	18	26			
50	44%	5	5-1	5	9	23	1		
				5	41	23	1		
			8-22	8	16	20	22		
				8	18	22			
			7-26	7	18	23	26		
				7	18	26			
			10-12	10	17	12			
				10	50	17	12		
			19-15	19	17	12	15		
				19	50	15			
50	44%	7	5-1	5	9	23	1		
				5	41	23	1		
			8-22	8	16	20	22		
				8	18	22			
			7-26	7	18	23	26		
				7	18	26			
			10-12	10	17	12			

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	Arc density	No. of SD pairs	SD node pair	Path nodes					
				10	50	17	12		
			19-15	19	17	12	15		
				19	50	15			
			20-50	20	23	50			
				20	12	17	50		
			17-29	17	19	29			
				17	10	29			
50	30%	3	5-1	5	25	1			
				5	9	6	1		
			8-22	8	45	22			
				8	45	20	22		
			7-26	7	18	23	26		
				7	45	20	23	26	
50	30%	5	5-1	5	25	1			
				5	9	6	1		
			8-22	8	45	22			
				8	45	20	22		
			7-26	7	18	23	26		
				7	45	20	23	26	
			10-12	10	17	12			
				10	50	17	12		
			19-15	19	17	12	15		
				19	50	15			
50	30%	7	5-1	5	25	1			
				5	9	6	1		
			8-22	8	45	22			
				8	45	20	22		

Continued on next page

Table A.1 Admissible paths (cont.)

No. of nodes	**Arc density**	**No. of SD pairs**	**SD node pair**	**Path nodes**					
			7-26	7	18	23	26		
				7	45	20	23	26	
			10-12	10	17	12			
				10	50	17	12		
			19-15	19	17	12	15		
				19	50	15			
			20-50	20	23	50			
				20	12	17	50		
			17-29	17	19	29			
				17	10	29			

Table A.2. Interdiction decisions in Scenario 1

No. of nodes	**Arc density**	**No. of SD pairs**	**Interdiction**
10	30	3	$X_{1,1} = X_{1,2} = X_{2,2} = X_{2,3} = X_{4,3} = X_{10,3} = 1$
		5	$X_{1,1} = X_{2,2} = X_{2,3} = X_{4,3} = X_{5,2} = X_{7,1} = X_{10,3} = 1$
	44	3	$X_{2,1} = X_{2,2} = X_{2,3} = X_{8,2} = 1$
		5	$X_{1,2} = X_{1,3} = X_{2,2} = X_{2,3} = X_{5,2} = X_{7,1} = 1$ $X_{8,2} = X_{8,4} = 1$
20	30	3	$X_{1,1} = X_{2,3} = X_{5,2} = X_{5,4} = X_{8,1} = X_{9,4} = 1$ $X_{10,2} = X_{10,3} = X_{11,3} = 1$
		5	$X_{1,1} = X_{2,3} = X_{5,2} = X_{5,4} = X_{8,1} = X_{9,4} = 1$ $X_{10,3} = X_{11,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{1,1} = X_{2,3} = X_{5,2} = X_{5,4} = X_{8,1} = X_{10,3} = 1$ $X_{11,3} = X_{13,2} = X_{16,1} = X_{16,3} = 1$
	44	3	$X_{1,1} = X_{5,2} = X_{5,4} = X_{8,1} = X_{9,4} = X_{10,2} = 1$ $X_{10,3} = X_{11,3} = X_{12,1} = 1$
		5	$X_{1,1} = X_{10,2} = X_{10,3} = X_{11,3} = X_{16,1} = X_{16,3} = 1$ $X_{18,3} = 1$
		7	$X_{1,1} = X_{8,1} = X_{10,3} = X_{11,3} = X_{13,2} = X_{15,4} = 1$ $X_{16,1} = X_{16,3} = 1$

Table A.3. Interdiction decisions in Scenario 2

No. of nodes	**Arc density**	**No. of SD pairs**	**Interdiction**
10	30	3	$X_{2,1} = X_{2,3} = X_{4,1} = X_{4,2} = X_{4,4} = X_{10,1} = 1$ $X_{10,2} = X_{10,3} = 1$
		5	$X_{2,1} = X_{2,3} = X_{4,1} = X_{4,3} = X_{10,1} = X_{10,2} = X_{10,4} = 1$
	44	3	$X_{2,1} = X_{2,3} = X_{4,1} = X_{4,2} = X_{4,4} = X_{10,1} = 1$ $X_{10,2} = X_{10,4} = 1$
		5	$X_{2,1} = X_{2,3} = X_{3,1} = X_{3,2} = X_{3,3} = X_{4,3} = 1$
20	30	3	$X_{1,1} = X_{1,2} = X_{1,3} = X_{7,3} = X_{8,1} = X_{8,3} = 1$
		5	$X_{1,1} = X_{1,3} = X_{7,3} = X_{8,1} = X_{8,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{1,1} = X_{1,2} = X_{1,3} = X_{1,4} = X_{8,1} = X_{8,2} = 1$ $X_{8,3} = X_{8,4} = X_{12,1} = X_{12,3} = X_{16,1} = X_{16,3} = 1$ $X_{16,4} = X_{18,2} = X_{18,3} = X_{18,4} = X_{20,1} = X_{20,2} = 1$
	44	3	$X_{1,1} = X_{1,3} = X_{7,3} = X_{16,1} = X_{16,3} = X_{18,2} = 1$
		5	$X_{1,1} = X_{1,3} = X_{7,3} = X_{16,1} = X_{16,2} = X_{16,3} = 1$
		7	$X_{1,1} = X_{1,3} = X_{7,3} = X_{12,3} = X_{16,1} = X_{16,3} = 1$

Table A.4. Interdiction decisions in Scenario 3

No. of nodes	**Arc density**	**No. of SD pairs**	**Interdiction**
10	30	3	$X_{1,1} = X_{9,1} = X_{9,3} = X_{9,4} = 1$
		5	$X_{2,1} = X_{3,3} = X_{9,1} = X_{9,3} = 1$
	44	3	$X_{1,1} = X_{2,1} = X_{2,3} = X_{3,1} = 1$
		5	$X_{1,3} = X_{2,1} = X_{2,3} = X_{7,1} = 1$
20	30	3	$X_{4,1} = X_{4,3} = X_{8,1} = X_{8,3} = 1$
		5	$X_{4,1} = X_{4,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{4,1} = X_{4,3} = X_{16,1} = X_{16,3} = 1$
	44	3	$X_{4,1} = X_{4,3} = X_{8,1} = X_{16,3} = 1$
		5	$X_{1,1} = X_{1,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{4,3} = X_{15,3} = X_{16,1} = X_{16,3} = 1$

Table A.5. Interdiction decisions in Scenario 4

No. of nodes	**Arc density**	**No. of SD pairs**	**Interdiction**
10	30	3	$X_{1,1} = X_{1,2} = X_{2,2} = X_{2,3} = X_{4,3} = X_{5,2} = X_{10,3} = 1$
		5	$X_{1,2} = X_{2,2} = X_{2,3} = X_{4,1} = X_{4,3} = X_{5,2} = 1$ $X_{7,1} = X_{10,3} = 1$
	44	3	$X_{2,1} = X_{2,2} = X_{2,3} = X_{4,3} = X_{5,2} = 1$
		5	$X_{1,1} = X_{2,2} = X_{2,3} = X_{5,2} = X_{7,1} = X_{8,2} = X_{10,3} = 1$
20	30	3	$X_{1,1} = X_{2,3} = X_{5,2} = X_{5,4} = X_{8,1} = X_{10,2} = 1$ $X_{10,3} = X_{11,3} = 1$
		5	$X_{1,1} = X_{2,3} = X_{5,2} = X_{5,4} = X_{8,1} = X_{10,3} = 1$ $X_{11,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{1,1} = X_{2,3} = X_{5,2} = X_{5,4} = X_{8,1} = X_{10,2} = 1$ $X_{11,3} = X_{13,2} = X_{16,1} = X_{16,3} = 1$
	44	3	$X_{1,1} = X_{2,3} = X_{5,4} = X_{8,1} = X_{9,4} = X_{10,2} = 1$ $X_{10,3} = X_{11,3} = X_{16,3} = 1$
		5	$X_{1,1} = X_{8,1} = X_{9,4} = X_{10,2} = X_{10,3} = X_{11,3} = 1$ $X_{16,1} = X_{16,3} = X_{18,2} = 1$
		7	$X_{1,1} = X_{8,1} = X_{10,2} = X_{10,3} = X_{11,3} = X_{12,1} = 1$ $X_{12,2} = X_{16,1} = X_{16,3} = 1$

Table A.6. Interdiction decisions in Scenario 5

No. of nodes	**Arc density**	**No. of SD pairs**	**Interdiction**
10	30	3	$X_{2,1} = X_{2,2} = X_{2,4} = X_{4,1} = X_{4,3} = X_{10,1} = X_{10,3} = 1$
		5	$X_{2,1} = X_{2,3} = X_{3,4} = X_{4,1} = X_{4,3} = X_{10,1} = X_{10,2} = 1$
	44	3	$X_{2,1} = X_{2,2} = X_{2,3} = X_{4,1} = X_{4,3} = X_{4,4} = 1$ $X_{10,2} = X_{10,4} = 1$
		5	$X_{1,1} = X_{2,1} = X_{2,3} = X_{3,3} = X_{4,2} = X_{4,4} = 1$
20	30	3	$X_{1,1} = X_{1,2} = X_{1,3} = X_{1,4} = X_{7,3} = X_{8,1} = X_{8,3} = 1$
		5	$X_{1,1} = X_{1,3} = X_{7,1} = X_{8,1} = X_{8,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{1,1} = X_{1,2} = X_{1,3} = X_{1,4} = X_{8,1} = X_{8,2} = 1$ $X_{8,3} = X_{12,1} = X_{12,3} = X_{16,1} = X_{16,2} = X_{16,3} = 1$ $X_{16,4} = X_{18,2} = X_{18,3} = X_{18,4} = X_{20,1} = X_{20,2} = 1$
	44	3	$X_{1,1} = X_{1,2} = X_{1,3} = X_{7,3} = X_{8,1} = X_{8,2} = X_{8,3} = 1$
		5	$X_{1,1} = X_{1,2} = X_{1,4} = X_{7,1} = X_{8,1} = X_{16,1} = X_{16,3} = 1$
		7	$X_{1,1} = X_{1,3} = X_{7,1} = X_{12,3} = X_{16,1} = X_{16,3} = 1$

Table A.7. Interdiction decisions in Scenario 6

No. of nodes	**Arc density**	**No. of SD pairs**	**Interdiction**
10	30	3	$X_{1,1} = X_{2,1} = X_{4,1} = X_{9,3} = 1$
		5	$X_{1,1} = X_{2,1} = X_{8,3} = X_{9,3} = 1$
	44	3	$X_{2,1} = X_{2,3} = X_{3,1} = X_{4,1} = 1$
		5	$X_{2,1} = X_{2,3} = X_{5,1} = X_{7,3} = 1$
20	30	3	$X_{4,1} = X_{4,3} = X_{5,1} = X_{8,3} = 1$
		5	$X_{4,1} = X_{4,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{4,1} = X_{4,3} = X_{16,1} = X_{16,3} = 1$
	44	3	$X_{4,1} = X_{4,3} = X_{5,3} = X_{16,1} = 1$
		5	$X_{4,1} = X_{4,3} = X_{16,1} = X_{16,3} = 1$
		7	$X_{4,1} = X_{4,3} = X_{16,1} = X_{16,3} = 1$