ROBUST DESIGN OF DISTRIBUTION NETWORKS CONSIDERING WORST CASE INTERDICTIONS

Alexandre Couedelo

A thesis
in
The Department
of
Mechanical, Industrial and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science
Concordia University
Montréal, Québec, Canada

May 2018

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Alexandre Couedelo

Entitled: Robust Design of Distribution Networks Considering Worst Case Interdictions

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________ Chair

Javad Dargahi

______________________ Examiner

Ivan Contreras

______________________ Examiner

Navneet Vidyarthi

______________________ Thesis Supervisor(s)

Onur Kuzgunkaya

____

Approved by ______________________

Javad Dargahi Chair of Department or Graduate Program Director

____

Dean,

Date ______________________

Abstract

Robust Design of Distribution Networks Considering Worst Case Interdictions

Alexandre Couedelo

Multi-echelon facility location models are commonly employed to design transportation systems. While they provide cost-efficient designs, they are prone to severe financial loss in the event of the disruption of any of its facilities. Additionally, the recent crisis in the world motivates OR practitioners to develop models that better integrate disruptive event in the design phase of a distribution network.

In this research, we propose a two-echelon capacitated facility location model under the risk of a targeted attack, which identifies the optimal location of intermediate facilities by minimizing the weighted sum of pre and post interdiction flow cost and the fixed cost of opening intermediate facilities. The developed model results in a tri-level Mixed Integer Programming (MIP) formulation, reformulated in a two-level MIP. Hence, we prescribe solution methods based on Bender Decomposition as well as two variants that enhance the speed performance of the algorithm.

The results reveal the importance of selecting backup facilities and highlight that premium paid to design a robust distribution network is negligible given the benefit of reducing the post-interdiction cost when a disruptive event occurs.

Acknowledgments

I would first like to thank my thesis supervisor Professor Onur Kuzgunkaya of the Departement of Mechanical, Industrial and Aerospace Engineering Industrial Engineering at Concordia University for his many helpful suggestions on my research direction.

I would also express my very profound gratitude to my parents for providing me with support and continuous encouragement throughout my years of studies in France and abroad, This accomplishment would not have been possible without them. Thank you.

Contents

List of Figures

List of Tables

List of Abbreviations

BD Cender Decomposition

BLMIP Bi-level Mixed Integer Programming

KKT Karush-Kuhn-Tucker

FLP Facility Location Problem

MCFP Multi-commodity Flow Problem

MEFLP Multi-Echelon Facility Location Problem

MFP Max-Flow Problem

MIP Mixed Integer Programming

MP Master Problem

PMP P-Median Problem

R-FLP Robust Facility Location Problem

R-TCFLP Robust Two-Echelon Capacitated Facility Location Problem

RPMP Reliable P-Median Problem

RUFLP Reliables Facility Location Problem

SPP Shortes Path Problem

SP Sub Problem

SVI Super-Valid Inequalities

TCFLP Two-Echelon Capacitated Facility Location Problem

UFLP Uncapacitated Facility Location Problem

Chapter 1

Introduction

Disasters or intentional attack on supply chains, even though they rarely occur, may result in significant financial losses due to the complexity and scale of today's logistic networks. Multiple recent crisis emphasis the need for more studies in network disruption such as the US border and air traffic shut down following the 11 September 2001 terrorist attack, which forced Ford to idle five plants because of shortage in supplies (Sheffi and Rice Jr, 2005), or more recently in 2015, after Paris terrorist attack the transport of goods has been reduced between France, Belgium and Luxembourg resulting in higher transportation cost in Europe (BSI, 2017)

In addition to the growing threat, today‘s logistics network design methods creates particularly vulnerable networks (Snyder et al., 2006), (Li et al., 2013). Undoubtedly, the just-in-time paradigm designs are cost efficient but may result in unreliable network under interdiction of facilities or transportation arcs. Taking disruption risk into account would be beneficial, not only because it limits repercussion of operating the network under degraded conditions, but also it leaves room for better strategic decision as the system gains more flexibility.

As a result of both the increase in recent severe disruption in supply chains affecting major corporation in the world and a weakness mentioned above in network design, studies in network disruption have gain ground within the past decade (Snyder et al., 2006). Those contributions are part of a more general effort put into supply chain risk management and network reliability. The current strategy to protect a distribution system is extending fundamental problems such as the P-median problem or the Fixed-Charge problem that focused on a location and allocation of customers to suppliers. Indeed, the supply chain network function is to distribute goods from supplier to customer. However, modern networks are more complicated than the supplier and customer relationship. It usually involves a sequence of facilities of different types that fulfill each a specific function or service to complete the manufacturing process. Since the topology of a supply chain network dramatically influences the operational capabilities under a disruption, considering such risk to enhance existing multilevel facility location model is critical.

Strategies that integrate the risk aspect into the design process are divided into stochastic and deterministic interdiction. The first category look at potential failure at different level including machine malfunction, road closure, plant shut down. This strategy aims to quantify the risk of interdiction through the enumeration of possible scenarios. On the other hand, deterministic interdiction focuses on a game theory approach where two opponents face each other. Therefore the design emerging from this approach aims to prevent the worst-case scenario. Both techniques assure the selection of a robust set of facility, such that an attack on the network will have little effect on its operation. Furthermore, a robust set must necessarily be considered in the design phase since the type of disruption in question is very sudden and would immediately result in negative financial impacts. On top of that, one may not optimally harden/protect a facility location design by adding backup facilities later on, because the number of facilities involved impact considerably the topology of a network.

The purpose of this thesis is to design distribution networks by considering the network interdiction aspect at the design stage by combining network interdiction and Multi-echelon facility location problem. While most studies considered mainly random and single failure, we focus on multiple deterministic interdictions which correspond to the worst case scenario possible in case of disruption. Furthermore to the best of our knowledge, there is no proposed model in literature to design complex supply chain under the assumption of a deterministic disruption. Therefore, we based our model on the standard Two-echelon Capacitated Facility Location Problem (TCFLP), in which the designer create a distribution network between supplier and client with an intermediary storage facility, and derive a new version robust against target attack. The enemy is assumed to have a fixed budget to eliminate facilities, and the designer is taking the potential threat into account while selecting the network design. The formulation of this problem is generic enough to extend it to k-echelon.

We provide algorithms based on Bender Decomposition, much as Smith et al. (2007) solved Multi-commodities flow network design under optimal interdiction. Besides, we explore several well-known enhancement techniques to accelerate the standard Bender Decomposition.

This thesis focuses on the optimal selection of transshipment facilities in a hierarchical network to minimize the design cost as well as both operating the network under usual and worst case disruption scenario. The rest of the thesis is organized as follows: Section 2 review the relevant literature; In Section 3 the problem statement is presented; In Section 4 we describe the proposed solution method, which includes a standard Bender Decomposition and two enhancement techniques; the numerical experiments in Section 5 illustrate the managerial insights and the computational performances of the algorithms; and Section 6 states the conclusions and future research areas.

Chapter 2

Literature Review

In this Chapter, we present an overview of the literature for network interdiction and Multi-Echelon Facility Location Problems (MEFLP), the two area of research that are brought together in the present thesis. We first look at real-world applications of network interdiction problems that have been considered in the literature. Later, in section 2.2 and 2.3 we expose the main contributions related to this thesis in both network interdiction and MEFLP. This work leads us to propose a classification of network introduction problems based on the initial model they study (e.g., flow routing, facility location, etc.) and the characteristic associated with the way potential disruptions are introduced into the model.

2.1 Topics in Network Interdiction

In this section, we introduce the notion of network interdiction and present the topics tackled in this field. The selected papers highlight the most common application of network interdiction relevant to this thesis.

A network interdiction problem involves two parties (i.e players, sides), an attacker (also called leader, enemy, interdictor or adversary) and a defender (also called follower, operator or owner). Both opponent participate to a game in which the defender operate a network and therefore aim to optimize its actions such as minimizing the operation cost, maximize the profitability or service level. On the other hand, the attacker attempt to disrupt the network such that it inflicts the maximum damage to the defender. The two person structure of such game designated as an attacker-defender model (defender-attacker depending on who is acting first) result in nested optimization problems that address simultaneously the criticality of infrastructures, and the robustness of a network. A structure is defined as critical for a network if its removal degrades the performances of the network significantly. While a network is said robust if he can fulfill its function after random or deliberate disruption.

The network interdiction problem has been studied for more than sixty years starting with the military and Homeland security concerns. For instance McMasters and Mustin (1970) and Ghare et al. (1971) studied the best possible way to plan an aircraft strike that interdicts the maximum flow of enemy supply. Brown et al. (2005) designed a defensive ballistic missile network that prevents the threat of an enemy strike. Barkley (2008) research on creating a robust IP-Network with regard to potential interdiction on routers (nodes) and connections (arcs). Brown et al. (2009) aim to best delay the completion of batches of nuclear weapon as long as possible based on scheduling theory.

Later, in this area of research Wood also tackled civil-world problems such as minimizing drug smuggling in two papers Wood (1993) and Washburn and Wood (1995). Assimakopoulos (1987) developed a strategy to limit the propagation of infections in hospitals. Anandalingam and Apprey (1991) used network interdiction to help resolve water sharing conflict between two nations with the intervention of an arbitrator such as the United Nation.

2.2 Network Interdiction Problems

Table 1 provides an overview of the literature of network interdiction problems with respect to the type of problem which can be a General model (G), Shortest Path Problem (SPP), Maximum Flow Problem (MFP), Multi-Commodity Flow network design (MCFP), Facility Location Problem (FLP), Multi-Echelon Facility Location Problem (MEFLP). The also differentiate the formulation approach which can be either Stochastic or deterministic; the type of disruption that corresponds to the attacker decision that is either continuous (i.e., the attacker can partially interdict an element of the network) or discrete (i.e., the attacker remove the target entirely from the network). Additionally, we indicate if the model integrates capacity constraint in the model. The designer may attempt to protect the network by using fortification (F), to design of the network (D) or focus on the optimal attack strategy (A).

Paper	Type of model	Formulation	Disruption	Capa.	Objective
	General model (G)				
Brown et al., 2006	G	Det.	Disc.	✓	F
Wood, 2011	G	Det.	Disc.	N.A.	N.A.
	Maximum Flow Problem (MFP)				
Wollmer, 1964	MFP	Det.	Disc.	✓	A
McMasters and Mustin, 1970	MFP	Det.	Cont.	✓	A
Ghare et al., 1971	MFP	Det.	Cont.	✓	A
	Multi-Commodity Flow network design (MCFP)				
Wood, 1993	MCFP	Det.	Disc. and Cont.	✓	D
Washburn and Wood, 1995	MCFP	Det.	Disc.	✓	N.A.
Lim and Smith, 2007	MCFP	Det.	Cont.	✓	D
Smith et al., 2007	MCFP	Det.	Disc. and Cont.	✓	D
Barkley, 2008	MCFP	Det.	Disc.	✓	F
Azad et al., 2013	MCFP	Det.	Disc. and Cont.	✓	F
	Shortest Path Problem (SPP)				
Fulkerson and Harding, 1977	SPP	Det.	Cont.	✓	A
Israeli and Wood, 2002	SPP	Det.	Disc.	✓	A
Bayrak and Bailey, 2008	SPP	Det.	Disc.	✓	A
Cormican et al., 1998	SPP	Sto.	Disc.	✓	A
Morton et al., 2007	SPP	Sto.	Disc.	✓	A
Pan and Morton, 2008	SPP	Sto.	Disc.	✓	A
Brown et al., 2009	SPP	Det.	Disc.	✓	A
	Facility Location Problem (FLP)				
Drezner, 1987	FLP	Sto.	Disc.		D
Bundschuh et al., 2003	FLP	Sto.	Disc.	✓	D
Church et al., 2004	FLP	Det.	Disc.		D
Snyder and Daskin, 2005	FLP	Sto.	Disc.		D
Snyder et al., 2006	FLP	Sto.	Disc.	✓	D
Li et al., 2013	FLP	Sto.	Disc.		D
Shishebori et al., 2014	FLP	Sto.	Disc.		D
	Multi-Echelon Facility Location Problem (MEFLP)				
Peng et al., 2011	MEFLP	Sto.	Disc. and Cont	✓	D
An et al., 2014	MEFLP	Sto.	Disc.	✓	D
Our Contribution	**MEFLP**	**Det.**	Disc.	✓	D

Table 1: Classification of network interdiction literature

2.2.1 Maximum Flow Interdiction

Reducing the maximum flow has received the most attention in the deterministic network interdiction problems. This problem extends the max flow problem (MFP) which constitutes the basis of the multi-commodity flow problem (MCFP) used to design transportation networks with multiple sources and sink in which the flow must travel from adjacent nodes.

Initially, Wollmer (1964) work on a flow network design with single source and sink. He investigates the consequences of arc interdiction while assessing the vulnerability of an existing network. He develops an algorithm that minimizes the maximum flow in a network by removing precisely k-arcs.

Later, McMasters and Mustin (1970) and Ghare et al. (1971) apply minimizing the maximum flow model to transportation network represented as a planar connected graph. They consider that the attacker is able to reduce arc capacity within a restricted budget, as a result not only he decides on the target but also on the associated effort. The more effort, the more budget is allocated, the more the capacity is reduced.

Wood (1993) generalized the integer programming formulation of deterministic interdiction problem applied to MCFP. Unlike the previous studies, his model does not require the network to be source-sink planar. Even thought source-sink planar assumption allows the problem to be solved efficiently via a dual-primal algorithm it imposes a single source and a single sink located on the periphery of the network. This assumption is not realistic for most real-life applications. Relaxing that strong assumption let the designer select multiple sources and sink anywhere in the network.

Washburn and Wood (1995) developed a general model for network interdiction using game theory. Their problem is modeled as a Two-player zero-sum game in which an evader (i.e., prisoner) tries to cross a network from source to sink by minimizing the probability of being discovered (i.e., arrested). The optimal solution to this problem is difficult to find, because it exists an exponentiation number of paths for the evader can choose. However, by reformulating the problem as minimizing the maximizing flow, then the optimal the solution may be obtained in polynomial time.

Lim and Smith (2007) formally characterize model and solution methods for multi-commodity flow with continuous and discrete interdiction. In a discrete interdiction problem, the attacker decision variables can only take integer values, while in continuous interdiction they can be any real value. They present two formulations for the problem in the discrete case, and they develop exact and heuristic algorithm for the continuous case. In a second contribution, Smith et al. (2007) tackle MCFP again, but solve the worst-case scenario that corresponds to disruption made by an intelligent enemy with full information, and also studied the attacks of an enemy with only partial information on the network. They modeled the worst-case scenario using a nested attacker-defender model. Attacks based on incomplete information are modeled using heuristics such as targeting arc with maximum capacity or targeting arc with maximum flow. The heuristic approaches can be easily solved using a cutting plane algorithm that rapidly generates solutions while minimizing the impact of the enemy strategy.

2.2.2 Shortest Path Interdiction

A second growing topic in this area of network interdiction is the shortest path interdiction. These models are more simplistic than MFP or MCFP; however authors on this subject develop efficient algorithms to solve their problems, and those techniques can be applicable to more general interdiction problems.

The first shortest path interdiction model was introduced by Fulkerson and Harding (1977), assuming that the enemy is able the extend the length of an arc within a constrained budget. To solve this problem, they used a reformulation of the maximum minimum-path problem into a single maximization problem by using the dual of the inner minimum-path problem.

Israeli and Wood (2002) develop a model where the interdictor maximizes the shortest path by removing or increasing the length of an arc by a fixed amount. Unlike Fulkerson and Harding (1977) paper the interdiction is modeled as a binary decision (i.e., discrete interdiction) therefore reformulation technique is not possible, instead of in that paper the authors develop an extreme point enumeration technique based on Bender Decomposition and super-valid inequalities.

Similar studies aim to maximize the critical path Brown et al. (2009), Granata et al., 2013 in that particular case the goal of the interdictor is to target the longest path in the optimal schedule.

The stochastic version of the SPP interdiction problem is studied in the following papers Cormican et al. (1998), Morton et al. (2007), Pan and Morton (2008). for a more detail literature review on shortest path network interdiction we refer to Sadeghi et al. (2017).

2.2.3 Facility Location and Network Disruption

Few papers consider discrete interdiction associated with the topic of facility location and very little tackle multi-level supply chain network design (Snyder et al., 2016). The early literature that integrates risk management in facility location focuses on randoms event such demand and uncertainty or machine failure.

A primary stream of the literature pursues the design of reliable supply chain. The *reliability* of a network refers to its ability to fulfill the demand completely even when subject to unplanned event. On the other hand, *robustness* focuses on the performance of the system under disruption; thus a solution is robust if a disruption has little impact on its objective value. For instance, Bundschuh et al. (2003) focus on supplier failure and presents a stochastic model that integrates both reliability and robustness in a multi-stage single customer facility location problem.

Drezner (1987) is the first to introduce potential disruption touching the network itself such as facilities or transportation arc. Most studied in FL consider stochastic interdiction represented by the probability that a facility will fail and become inactive.

Snyder and Daskin (2005) and Li et al. (2013) worked on the reliable model for facility location and developed the reliable p-median problem (RPMP) and the reliable uncapacitated facility location problem (RUFLP). Shishebori et al. (2014) established reliable facility location and network design that associate each customer with multiples facilities (backup facilities) considering the maximum allowable failure cost; therefore it limits the increase in cost incurred during the disruption to an acceptable loss in any circumstances.

Peng et al. (2011) work a reliable two-echelon (supplier, transshipment, customer) network design problem based on the p-robustness approach that protects the network against potentially unreliable transshipment facilities. An et al. (2014) introduce a two-stage stochastic interdiction problem and formulate the objective as to minimize the weighted sum of the operation both before and after disruption much as Smith et al. (2007) did for MCFP.

Church et al. (2004) studies the deterministic interdiction of facilities, applied to a standard facility location problem, he introduces the r-interdiction median problem which is a version of p-median problem robust against intentional strikes.

2.3 Multi-level Facility Location Problems

In this subsection, we summarize the main contribution in terms of formulation and characteristic of the model.

Multi-echelon facility location problem (MEFLP) is a traditional problem in supply chain design. It aims to simultaneously select the required facilities in a network (Network Design Problem) and satisfy the customer demand by assigning them to a sequence of open facilities (Flow Routing Problem). This family of problems is also commonly designated by the terms multi-stage, multi-level, hierarchical, multi-layer. Those models provide a tightly optimize solution, which is often very sensitive to disruption. Indeed the unexpected losses of critical infrastructures may substantially impact the cost of operation and distribution of products through the network. The best performance of the supply chain is the trade-off between the location of facilities and allocation of customers.

Problems in facility location can be categorized based on their objective function (Daskin, 2008). This classification defines two categories of problems (i) covering base model, (ii) median base model that constitute the foundation for most of the advanced models in either network interdiction models or hierarchical network models. The first category focuses on covering the customers, by only looking at their allocation to facilities. For instance, the Set Covering Problem (SCP) is one of the problems in this category and has the objective of minimizing the number of facilities required to satisfy the diniemand. The second category mainly targets the cost of delivering a given amount of demand. The P-Median Problem (PMP) for instance aim to minimize the demand-weighted average distance between customers and facilities (Hakimi, 1964). This category also includes Fixed-Charge Location Problem that does not contain a limit on the number of facilities but instead has the objective to minimize the cost of utilizing a facility (i.e., operation, construction) and the cost of serving each customer. The uncapacitated facility location problem (UFLP) is a well-known problem fixed-charge problem (Cornuéjols et al., 1983).

The location-allocation models stated before are usually too simplistic to manage the complete distribution channel. Indeed, to best manage a supply chain we would like to integrate multiple types of infrastructures such as suppliers, plants, distribution centers, customers, etc. Therefore one important extension looks at the inclusion of multiple layers (i.e., levels) where each one of which playing a specified role. This family of models is designated in the literature as hierarchical facility location models. Şahin and Süral (2007) develop a classification of the research on this topic.

Facility interdiction in multi-echelon distribution networks has received scarce attention in the literature. Therefore we want to expand this area of by tackling a fundamental problem in the topic. The simplest MEFLP is without a doubt the two-echelon uncapacitated facility location, which can be used to design a distribution network with intermediate facilities (e.g., distribution center). This problem is one of the most studied in the field (Ortiz-Astorquiza et al., 2017). A more interesting version is the capacitated two-echelon facility location (TCFLP) however less effort in the litterature has been made on the capacitated case. Aardal (1992) presented an early study on this problem and later a reformulation called the two-index formulation (Aardal, 1998). For a detailed review of multi-level facility location problem we refer to Melo et al. (2009) and for a more recent review focusing on the formulation approaches we refer to Ortiz-Astorquiza et al. (2017).

2.4 Conclusion

The above literature review shows that network interdiction is a growing topic in network operations. Even though it emerged from military concerns, authors on the subject demonstrate its use in numerous domains, particularly, the design of distribution network. However, we found that the literature only scratches the surface of the problem of designing robust distribution network, and little or no work consider the following aspects together (i) the protection against worst case scenario in facility location; (ii) The design of complex distribution systems such as multi-echelon facility location that reflect real-world application; (iii) the interdiction of capacitated facility.

In this regard, we identify in the multi-echelon literature an initial model namely TCFLP that would answer the aforementioned gaps in the literature if it is coupled with the deterministic network interdiction approach. The proposed TCFLP is a capacitated model, and it addresses the design of distribution network with intermediate facilities which are a typical complex structure in the field. Additionally, deterministic network interdiction focuses on the study of the worst-case scenario. Finally, we aim to develop efficient solution methods for our problem by taking inspiration from the state of the art in determianistic network interdiction such as multi-commodity flow interdiction and shortest path network interdiction. We compare our model to the previous work in the literature in Table 1.

Chapter 3

Problem Statement and Notation

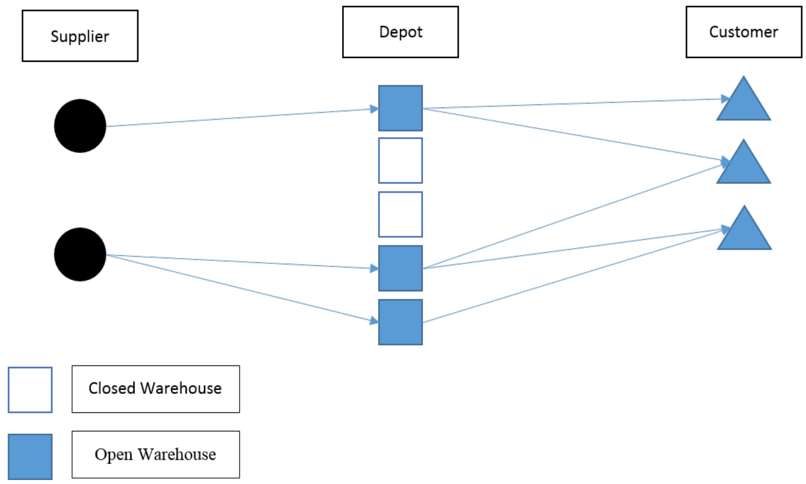
The problem concerned with the best placement of infrastructures and the best allocation of customers is one of the most studied topics in operation research. Indeed, selecting the location of intermediate facilities reflect a critical long-term strategic decision for a company. Moreover, a dominant structure in transportation systems employs multi-echelon design.

In this context, our objective is to provide a network design that minimizes the total costs while preventing the financial damage that could result from a disruption affecting facilities in a network. To achieve this, we applied network interdiction approach to a well studied hierarchical facility location problem.

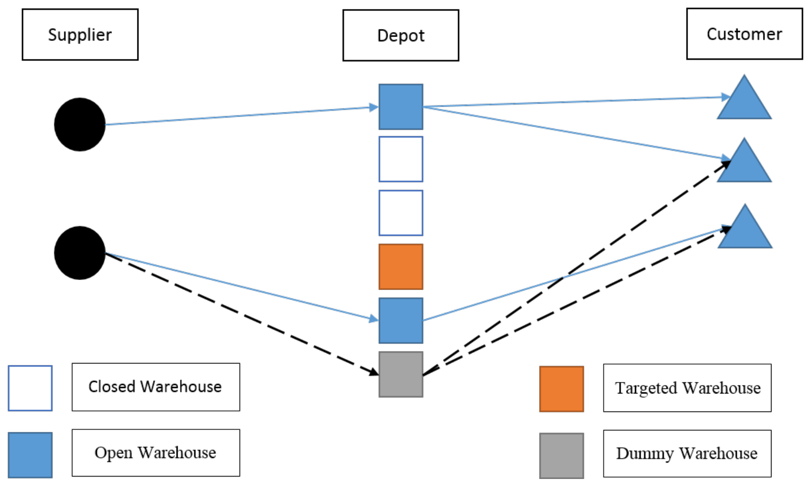
In the following subsections, we first introduce and illustrate the problem statement of this research, then present the mathematical formulation that provides a robust solution with respect to intentional attack. Finally, we offer an interpretation of the formulation to help the reader to understand its meaning and practical implications.

3.1 Problem Description

We consider a single product supply chain design that includes a set of suppliers, a set of intermediate facilities (e.g., distribution center, or depot), and a set of customer. In this problem, the designer aims to construct a two-echelon distribution network while minimizing two types of cost (i) a fixed annualized setup cost related to the acquisition of distribution center and (ii) the transportation cost associated with



(a) Pre-interdiction



(b) Post-interdiction

Figure 1: Example of Network Design and Interdiction

the satisfaction a deterministic demand. The designer's decisions are the selection of intermediate facilities among a pre-selected set of candidates, and the number of products to flow between each level of the network to meet the demand. Also, the capacity of every intermediate facility is limited. Figure 1a this problem illustrates and show the pre-interdiction flow routing decision. Nonetheless, this problem does not protect the network against enemy attack.

To better design the network against a targeted attack, we must consider the optimal attacker strategy. The enemy will, given a limited budget, maximize the post-interdiction flow routing cost by destroying intermediate facilities (completely). As a consequence of the facility interdiction, the designer is forced to find an emergency routing strategy. It is essential to state the assumption that the designer can not open new facilities after a disruptive event. This assumption is justified by the fact that disruptions are very sudden an unpredictable. Therefore the design is only capable of reacting on an operational level. This problem reflects the cost of operating the network after an attack. Figure 1b illustrates the post-interdiction flow routing decision, the ennemy interdict one fqcility from Figure 1a, and it shows the possibles consequences of such disruptive scenario: (i) the need of rerouting the flow of product (ii) the incapability of satisfying the demand due to a lack of capacity (the extra flow goes the dummy facility).

Finally, The design of a robust distribution network under the risk of interdiction can be modeled as a three-stage game: (i) the operator design an initial multi-echelon distribution network by selecting a set of facilities and allocate the flow of supply in the network (ii) the attacker removes facilities within its budget as to maximize the post-interdiction cost (iii) the operator redistribute the flow among the survived facilities as to minimize the post-interdiction cost.

However, the network we design is not always under attack, so we which to consider both the cost under normal and disrupted circumstances. Therefore we define the objective of this problem as a weighted sum of the pre-interdiction flow cost, the post-interdiction flow cost and the fixed setup cost of opening facilities. As a result that the designer is capable of taking into account the attacker strategy by defining the importance he which to give to the threat to design a network robust against the worst disruption scenario.

3.2 Mathematical Formulation

The purpose of this research is to provide a robust solution against targeted attack for multi-echelon facility location network. Consequently, we introduce a generic formulation that applies to all facility location problem, and we called it robust facility location problem R-FLP because it derives from a generic facility location problem (FLP). Then we extend the two-level capacitated facility location problem (TCFLP) using that formulation. We choose TCFLP because it has been the most studied among the multi-level facility location problem (Ortiz-Astorquiza et al., 2017). The result of this section is the explicit formulation of the robust two-level capacitated facility location problem (R-TCFLP). R-TCFLP is composed of two subproblem namely Pre-interdiction ($Pre(\boldsymbol{y})$) and Post-interdiction ($Post(\boldsymbol{y})$) flow routing associated with the network topology $\boldsymbol{y}$. Table 2 present the list of mathematical terms introduced in this section as well as their notation.

Notation	Description
FLP	Facility Location Problem
$R-FLP$	Robust Facility Location Problem
$TCFLP$	Two-echelon Capacitated Facility Location Problem
$R-TCFLP$	Robust Two-echelon Capacitated Facility Location Problem
$Pre(y)$	Pre-interdiction flow routing Problem given network topology $\boldsymbol{y}$
$Post(y)$	Post-interdiction flow routing Problem given network topology $\boldsymbol{y}$

Table 2: List of Problems

The lists of notations and decision variables are used in this chapter are defined in Tables 3 and 4.

Notation	Description
i	Index of customer; i $\in$I
j_1	Index of warehouse; $j_1 \in V_1$
j_2	Index of supplier; $j_2 \in V_2$
c_{ab}	Unit transportation cost from node a to node b
f_j	Setup cost to open warehouse j
α_j	Capacity of facility j
B	Attacker budget
b_j	Budget required to target facility j

Table 3: List of decision variables

Notation	Description
v_{ab}	Unit of flow from node a to b in the pre-interdiction problem
w_{ab}	Unit of flow from node a to b in the post-interdiction problem
y_j	1 if warehouse j is selected
t_j	1 if warehouse j is not targeted by an attack

Table 4: List of decision variables

3.2.1 Robust Facility Location Problem

As described previously, we begin with considering a multi-echelon facility location problem composed of two type of cost noted $\Psi(\boldsymbol{y})$ for the flow routing and $F(\boldsymbol{y})$ for the fixed setup cost. We aim to find the set of facilities y that minimize those two costs and respect the system of constraint Y_Ψ associated with the distribution of flow in the system. Equation 1 provide a general formulation of this initial facility location problem (FLP).

$$(FLP)\ min_{\boldsymbol{y} \in Y_\Psi} : \Psi(\boldsymbol{y}) + F(\boldsymbol{y}) \tag{1}$$

We modify FLP formulation in order to take into account a potential threat by considering a weighted combination of the pre-interdiction flow cost $\Psi(y)$, the post-interdiction flow cost $\Phi(\boldsymbol{y})$, and the fixed setup cost of opening facilities $F(\boldsymbol{y})$. We introduce the weight ρ to control the importance of the pre-interdiction cost over the post-interdiction cost. The decision variable $\boldsymbol{y}$ has to respect both the post-interdiction and pre-interdiction constraint noted respectively Y_Ψ and Y_Φ. Equation 2 represent the formulation for the robust multi-echelon facility location (R-FLP).

$$(R - FLP)\ min_{\boldsymbol{y} \in Y_\Psi \cap Y_\Phi} : \rho\Psi(\boldsymbol{y}) + (1-\rho)\Phi(\boldsymbol{y}) + F(\boldsymbol{y}) \tag{2}$$

The post-interdiction problem is a critical part of our formulation because it involves a two player game structures resulting in a nested min-max optimization also called attacker-defender model (Brown et al., 2006). In an attacker-defender model, the defender model is the foundation of the problem. In our research, the defender aims to minimize the operation cost represented by $\Psi(\boldsymbol{y})$. Given a network topology $\boldsymbol{y}$ and the defender model, the enemy maximize the damages he deals $\Phi(\boldsymbol{y})$ described by Equation 3. The enemy modifies the network topology $\boldsymbol{y}$ by targeting facilities represented by the vector $\boldsymbol{t}$. The survived network topology is given by $\boldsymbol{t} \circ \boldsymbol{y}$ (where $\circ$ is the element-wise multiplication of vectors).

$$\Phi(\boldsymbol{y}) = max_{\boldsymbol{t} \in T} \Psi(\boldsymbol{t} \circ \boldsymbol{y}) \tag{3}$$

We have now a general formulation for our problem. The rest of this section apply this formulation to TCFLP, which leads us to define in that context the pre-interdiction problem $Pre(\boldsymbol{y})$ and the post interdiction problem $Post(\boldsymbol{y})$. The explicit formulation of TCFLP is introduced in the next subsection.

3.2.2 Robust Two Echelon Facility Location Problem

Two Echelon Facility Location Problem

The commonly accepted formulation of TCFLP was introduced by Aardal, 1998 with the particularity of using two sets of continuous variables v_{ij_1} and $w_{j_1j_2}$ representing the flow to customer i from warehouse j_1, and to warehouse j_1 from the source j_2 respectively. The unit transportation cost associated to v_{ij_1} and $w_{j_1j_2}$ is respectively c_{ij_1} and $c_{j_1j_2}$. The demand of a customer is represented by d_i and the capacity of the warehouse and the source are respectively α_{j_1} and α_{j_2}. Finally, the binary decision variable y_{j_1} is equal to 1 if we decide to open the facility j_1 at a cost fixed f_{j_1}. The designer problem TCFLP is formulated as follows:

$$(TCFLP)$$

$$min_{v,y} : \sum_{i \in I} \sum_{j_1 \in V_1} c_{ij_1} v_{ij_1} + \sum_{j_1 \in V_1} \sum_{j_2 \in V_2} c_{j_1j_2} v_{j_1j_2} + \sum_{j1 \in V1} f_{j1} y_{j1} \tag{4a}$$

$$s.t : \sum_{j_1 \in V_1} v_{ij_1} \geq d_i \qquad \forall i \in I \tag{4b}$$

$$\sum_{j_2 \in V_2} v_{j_1j_2} \geq \sum_{i \in I} v_{ij_1} \qquad \forall j_1 \in V_1 \tag{4c}$$

$$\sum_{i \in I} v_{j_1j_2} \leq \alpha_{j_2} y_{j_2} \qquad \forall j_2 \in V_2 \tag{4d}$$

$$\sum_{j_2 \in V_2} v_{j_1j_2} \leq \alpha_{j_1} y_{j_1} \qquad \forall j_1 \in V_1 \tag{4e}$$

The objective function 4a includes flow routing cost and setup cost for warehouses. The demand must be satisfied 4b. The networks follows the flow conservation property 4c. The capacity of each facility must be respected 4d and 4e.

Pre-Interdiction Problem

We identify the pre-interdiction problem $Pre(y)$ and the fixed setup cost $F(y)$ in the context of TCFLP. Given TCFLP formulation we identify the element of Equation 1. The fixed cost is expressed in Equation 5.

$$F(\boldsymbol{y}) = \sum_{j1 \in V1} f_{j1} y_{j1} \tag{5}$$

The rest the problem once we remove $F(\boldsymbol{y})$ is the pre-interdiction flow network problem $Pre(\boldsymbol{y})$. You may notice that this problem is purely a linear, this will be useful in the following section.

$$
\begin{aligned}
& (Pre(\hat{\boldsymbol{y}})) \\
\Psi(\hat{\boldsymbol{y}}) = min_v :& \sum_{i \in I} \sum_{j_1 \in V_1} c_{ij_1} v_{ij_1} + \sum_{j_1 \in V_1} \sum_{j_2 \in V_2} c_{j_1 j_2} v_{j_1 j_2} && && \text{(6a)} \\
s.t :& \sum_{j_1 \in V_1} v_{ij_1} \geq d_i && \forall i \in I && \text{(6b)} \\
& \sum_{j_2 \in V_2} v_{j_1 j_2} \geq \sum_{i \in I} v_{ij_1} && \forall j_1 \in V_1 && \text{(6c)} \\
& \sum_{i \in I} v_{j_1 j_2} \leq \alpha_{j_2} \hat{y_{j_2}} && \forall j_2 \in V_2 && \text{(6d)} \\
& \sum_{j_2 \in V_2} v_{j_1 j_2} \leq \alpha_{j_1} \hat{y_{j_1}} && \forall j_1 \in V_1 && \text{(6e)}
\end{aligned}
$$

Post-Interdiction Problem

We define the post-interdiction problem in the context of TCFLP using the attacker-defender model, and then we use reformulation technique to convert the bilevel min-max formulation into a single level maximization model.

Equation 7 uses the general attacker-defender model introduced by Equation 3. We assume that only intermediate facilities can be targeted. Therefore the facility interdiction is only reflected in the constraint 7f, and it signifies that when a facility is targeted its capacity becomes null. To differentiate the decision variables between post-interdiction and pre-interdiction, we note v the decision variables related to the pre-interdiction flow and w the one related to the post-interdiction flow.

$$
\begin{aligned}
&(Post(\hat{\boldsymbol{y}})) \\
&\Phi(\hat{\boldsymbol{y}}) = max_{\boldsymbol{t}\in T}\Psi(\hat{\boldsymbol{y}} \circ \boldsymbol{t}) && && \text{(7a)} \\
&\Psi(\hat{\boldsymbol{y}} \circ \boldsymbol{t}) = min_w : \sum_{i\in I}\sum_{j_1\in V_1} c_{ij_1}w_{ij_1} + \sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{j_1j_2}w_{j_1j_2} && && \text{(7b)} \\
&s.t: \sum_{j_1\in V_1} w_{ij_1} \geq d_i && \forall i \in I && \text{(7c)} \\
&\sum_{j_2\in V_2} w_{j_1j_2} \geq \sum_{i\in I} w_{ij_1} && \forall j_1 \in V_1 && \text{(7d)} \\
&\sum_{i\in I} w_{j_1j_2} \leq \alpha_{j_2}\hat{y_{j_2}} && \forall j_2 \in V_2 && \text{(7e)} \\
&\sum_{j_2\in V_2} w_{j_1j_2} \leq \alpha_{j_1}\hat{y_{j_1}}t_{j_1} && \forall j_1 \in V_1 && \text{(7f)}
\end{aligned}
$$

where: $T = \{t \in \mathbb{R}^{|V1|} | \sum_{j_1\in V1} b_{j_1}(1 - t_{j_1}) = B, t_{j_1} \in \{0,1\}, \forall j_1 \in V1\}$

In this study, the attacker is allowed to remove nodes from the set of the selected warehouse $\hat{\boldsymbol{y}}$ to maximize the designer's (minimum) operational cost within the limited budget B. The binary vector $\boldsymbol{t}$ represent the enemy targets. If the decision variable t_{j_1} it is equal to 0 then the facility j_1 is interdicted.

In a capacitated case, it may happen that after an attack the combined capacity of the remaining intermediate facilities is not sufficient to handle all the demand. To handle that case, we assume that we dispose of an untargetable *emergency facility* that collect and distribute the remaining flow, applying a penalty for each unit of flow delivered via this dummy facility. For the sake of simplicity, we add the emergency facility to the dataset as a free warehouse with unlimited capacity, and a fixed transportation cost designated as *emergency cost*.

$Post(\hat{\boldsymbol{y}})$ is a bi-level mixed integer problem, however, as mentioned before the inner flow routing problem is a linear problem; therefore we can use the duality theory to convert this bi-level formulation into in single level maximization model by taking the dual of the inner optimization. We consider $\beta, \gamma, \delta, \epsilon$ the dual variables associated with the inner minimization problem and reformulate our problem as follows:

$$
\begin{aligned}
&(Post(\hat{\boldsymbol{y}}))) \\
\Phi(\hat{\boldsymbol{y}}) = max_{\beta,\gamma,\delta,\epsilon,t} : &\sum_{i\in I} d_i\beta_i - \sum_{j_2\in V2} (\alpha_{j_2} y_{j_2})\delta_{j_2} - \sum_{j_1\in V1} (\alpha_{j_1} \hat{y_{j_1}}) t_{j_1}\epsilon_{j_1} && \text{(8a)}\\
s.t: &\beta_i - \gamma_{j_1} \le c_{ij_1} \qquad , \forall i \in I, \forall j_1 \in V1 && \text{(8b)}\\
&\gamma_{j_1} - \delta_{j_2} - \epsilon_{j_1} \le c_{j_1 j_2} \qquad \forall j_1 \in V1, \forall j_2 \in V2 && \text{(8c)}\\
&\sum_{j_1\in V1} b_{j_1} t_{j_1} \le \sum_{j_1\in V1} b_{j_1} - B && \text{(8d)}
\end{aligned}
$$

The terms $t_{j_1} \times \epsilon_{j_1}$ in the objective function 8a is nonlinear because it involves the product of two decision variables. This problem can be linearized by substituting t_{j_1} and ϵ_{j_1} by a single decision variable that we call $t\epsilon_{j_1}$ and by adding the standard linearizion constraints 9a, 9b, and 9c.

$$
\begin{aligned}
t\epsilon_{j_1} - \bar{\epsilon_{j_1}} * t_{j_1} &\le 0 \qquad , \forall j_1 \in V1 && \text{(9a)}\\
t\epsilon_{j_1} - \epsilon_{j_1} &\le \qquad , \forall j_1 \in V1 && \text{(9b)}\\
t\epsilon_{j_1} - \epsilon_{j_1} - \bar{\epsilon_{j_1}} t_{j_1} &\ge -\bar{\epsilon_{j_1}} \qquad , \forall j_1 \in V1 && \text{(9c)}
\end{aligned}
$$

ϵ_{j_1} is the shadow cost associated with the capacity constraint of facility j_1. $\bar{\epsilon_{j_1}}$ represents an upper bound for ϵ_{j_1}. However Lim and Smith, 2007 emphasize that to obtain the best result from this linearized mixed-integer programming formulation one must use the smallest possible upper bound for $\bar{\epsilon_{j_1}}$. A good upper bound we choose is the longest simple path linking a facility j_1 to any couple customer-supplier $(i, j_2) \in V1 * V2$ defined in Equation 10.

$$
\epsilon_{j_1} \le \bar{\epsilon_{j_1}} = max(c_{ij_1} + c_{j_1 j_2}), \forall i \in I, \forall j_2 \in V2 \tag{10}
$$

Final Formulation

By pulling together all the elements of the previous sections, we define the final formulation for R-TCFLP as follows:

$$
\begin{aligned}
&(R-TCFLP)\\
&\qquad min_{y,v} : \rho[\sum_{i\in I}\sum_{j_1\in V_1} c_{ij_1}v_{ij_1} + \sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{j_1j_2}v_{j_1j_2}]\\
&\qquad\qquad + (1-\rho)max_{\beta,\gamma,\delta,\epsilon,t}[\sum_{i\in I} d_i\beta_i \; - \sum_{j_2\in V2}(\alpha_{j_2}y_{j_2})\delta_{j_2} - \sum_{j_1\in V1}(\alpha_{j_1}y_{j_1})t\epsilon_{j_1}]\\
&\qquad\qquad + \sum_{j1\in V1} f_{j1}y_{j1} && (11a)\\
&\qquad s.t: \sum_{j_1\in V_1} v_{ij_1} \geq d_i \qquad \forall i\in I && (11b)\\
&\qquad\qquad \sum_{j_2\in V_2} v_{j_1j_2} \geq \sum_{i\in I} v_{ij_1} \qquad \forall j_1\in V_1 && (11c)\\
&\qquad\qquad \sum_{i\in I} v_{j_1j_2} \leq \alpha_{j_2}y_{j_2} \qquad \forall j_2\in V_2 && (11d)\\
&\qquad\qquad \sum_{j_2\in V_2} v_{j_1j_2} \leq \alpha_{j_1}y_{j_1} \qquad \forall j_2\in V_2 && (11e)\\
&\qquad\qquad \beta_i - \gamma_{j_1} \leq c_{ij_1} \qquad \forall i\in I, \forall j_1\in V1 && (11f)\\
&\qquad\qquad \gamma_{j_1} - \delta_{j_2} - \epsilon_{j_1} \leq c_{j_1j_2} \qquad \forall j_1\in V1, \forall j_2\in V2 && (11g)\\
&\qquad\qquad \sum_{j_1\in V1} b_{j_1}t_{j_1} \leq \sum_{j_1\in V1} b_{j_1} - B && (11h)\\
&\qquad\qquad t\epsilon_{j_1} - \bar{\epsilon_{j_1}} * t_{j_1} \leq 0 \qquad \forall j_1\in V1 && (11i)\\
&\qquad\qquad t\epsilon_{j_1} - \epsilon_{j_1} \leq \qquad \forall j_1\in V1 && (11j)\\
&\qquad\qquad t\epsilon_{j_1} - \epsilon_{j_1} - \bar{\epsilon_{j_1}}t_{j_1} \geq -\bar{\epsilon_{j_1}} \qquad \forall j_1\in V1 && (11k)
\end{aligned}
$$

3.3 Interpretation of the Formulation

The formulation presented in the previous section is convenient for two reasons; first, the designer can adjust the importance given to a disruption or threat, second, it offers it is suitable regarding resolution methods. However, it is not easy for the manager to visualize the ins and outs of this formulation as it is. Therefore in this section, we offer comprehensive insight on the subject.

To illustrate the use of this methodology we propose a more user-friendly formulation of R-TCFLP and we introduce $r(\boldsymbol{y})$ the increase in operating cost due to an optimal interdiction when given a set of open intermediary facility $\boldsymbol{y}$.

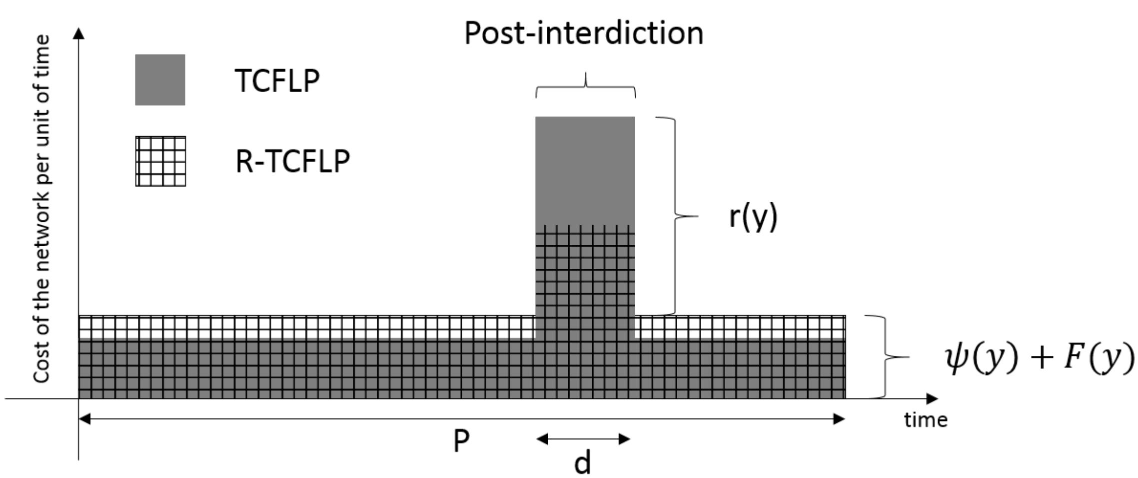
$$\Phi(\boldsymbol{y}) = \Psi(\boldsymbol{y}) + r(\boldsymbol{y}) \tag{12}$$

R-TCFLP can be reformulated as the sum of the annualized cost of the network under normal circumstances $\Psi(\boldsymbol{y}) + F(\boldsymbol{y})$ and the weighted additional cost due to any malicious disruptive event.

$$Z = min_{\boldsymbol{y} \in Y} \Psi(\boldsymbol{y}) + F(\boldsymbol{y}) + (1 - \rho) r(\boldsymbol{y}) \tag{13}$$

Figure 2 illustrates the difference between TCFLP and R-TCFLP per unit of time. The design produced with R-TCFLP increase the network pre-interdiction operation cost and fixed setup cost $\Psi(\boldsymbol{y}) + F(\boldsymbol{y})$. However, in return, the cost of operating the network during the post-interdiction phase (d) will be significantly reduced. Adjusting the parameter ρ let the managers decide to which extent the threat must be taken into account in the design of a supply chain network. One must not forget that the budget of the interdictor B is the second parameter that influences $r(\boldsymbol{y})$.

Figure 2: Example of network life cycle under disruption



Chapter 4

Solution Methodology and Algorithm

The mathematical description of our problem is a bi-level mixed-integer linear program (BLMIP). The traditional method used to solve such problem is reformulation based on Karush-Kuhn-Tucker (KKT) optimality condition to convert the bi-level problem into a single level problem. However, the KKT condition does not apply to the inner optimization when its variables are not all continuous, because the problem has, in general, no strong duality property. Therefore it is in general not possible to use reformulation technique when solving discrete interdiction problems. Instead, decomposition techniques using an enumeration of extreme points are used to solve such problem. It is proven that in this type of problem at least one optimal solution is reached at an extreme point of the constraint region (Saharidis and Ierapetritou, 2009).

In this chapter, we developed a Bender Decomposition (BD) algorithm and improved Bender decomposition to solve as efficiently as possible our problem to optimality. We formally introduce the Standard Bender Decomposition in the next section, and then we apply BD to our problem. Finally, we review the methods dedicated to accelerating the Bender Decomposition and we develop two ways of improving the computation time of this algorithm. The Section 4.3 deals with integrated the so-called super-valid inequalities to further constrain our problem at each iteration and converge faster toward the solution. Section 4.4 uses the Papadakos (2008) method which aims to find a non-dominated cut at each iteration by adding cuts called Pareto-optimal.

4.1 Standard Bender Decomposition

4.1.1 Formal Definition

Decomposition methods are often used to solve a large-scale optimization problem because they reduce the size of the problem that the computer has to consider at once. Such techniques as introduced by Benders (1962) can be used to solve bi-level linear problems (Saharidis and Ierapetritou, 2009). The Bender decomposition algorithm (BD) divide any problem into a master problem (MP) and a slave problem (SP). Then the result of MP provide a solution that is used to set variable constant for SP, and by solving SP we can create a Bender cut that will constraint MP. We repeat the process of solving variations of SP based on the solution of MP and constrain MP based on SP solution until MP and SP objective function reach an acceptable gap. Upper and lower bound of the original problem are provided by MP and SP respectively at each iteration, and Benders (1962) demonstrated that after a finite number of iteration this algorithm converges to an optimal solution.

We formally introduce some general minimization problem P:

$$
\begin{aligned}
(P)\ min: &\quad c^T\boldsymbol{w} + f(\boldsymbol{y}) && \text{(14a)}\\
s.t: &\quad A\boldsymbol{w} + F(\boldsymbol{y}) \leq \boldsymbol{b} && \text{(14b)}\\
&\quad \boldsymbol{w} \geq 0 && \text{(14c)}\\
&\quad \boldsymbol{y} \in Y && \text{(14d)}
\end{aligned}
$$

Where $\boldsymbol{c}$ is a vector of coefficient associated to the vector of continuous variables $\boldsymbol{w}$, and f is scalar functions that take the vector of variables $\boldsymbol{y}$ as input. Similarly, $\boldsymbol{A}$ is the matrix of coefficient associated with the constraints multiplied with the vector of variables $\boldsymbol{w}$ and F a set of scalar functions that takes as input the vector of variables $\boldsymbol{y}$. Given this problem P we isolate the set of variable $\boldsymbol{w}$, and we formulate the following sub-problem (SP) where $\boldsymbol{y}$ become a constant vector noted $\bar{\boldsymbol{y}}$. SP is mathematical describe as following:

$$
\begin{aligned}
(SP)\, min: \quad & z(\boldsymbol{w}) := c^T\boldsymbol{w} && (15a)\\
s.t: \quad & A\boldsymbol{w} \leq \boldsymbol{b} - \bar{\boldsymbol{y}} && (15b)\\
& \boldsymbol{u} \geq 0 && (15c)
\end{aligned}
$$

The Bender Decomposition algorithm deals with on the dual problem of SP instead (Dual-SP), because the dual variables' are related to constraint right-hand side value and are required to generate Bender cut for MP. We define $\boldsymbol{u}$ the set of dual variables of Dual-SP.

$$
\begin{aligned}
(Dual - SP)\, max: \quad & z(\boldsymbol{u}, \bar{\boldsymbol{y}}) := (b - \bar{\boldsymbol{y}})^T\boldsymbol{u} && (16a)\\
s.t: \quad & A^T\boldsymbol{u} \geq c && (16b)\\
& \boldsymbol{u} \geq 0 && (16c)
\end{aligned}
$$

By successively solving Dual-SP for different values of $\boldsymbol{y}$ provided by MP we characterize the polyhedron U which define the feasible region of our problem P by enumerating the set of extreme points P and extreme rays R. As a result, the corresponding Bender Master Problem (MP) is:

$$
\begin{aligned}
(MP)\, min: \quad & f(\boldsymbol{y}) + z && && (17a)\\
s.t: \quad & (\boldsymbol{b} - \boldsymbol{y})^T\bar{\boldsymbol{u}} \leq z && , \forall \bar{\boldsymbol{u}} \in P && (17b)\\
& (\boldsymbol{b} - \boldsymbol{y})^T\bar{\boldsymbol{u}} \leq 0 && , \forall \bar{\boldsymbol{u}} \in R && (17c)\\
& \boldsymbol{y} \in Y && && (17d)
\end{aligned}
$$

Constraints 17b and 17c represent the set of all the Bender cuts. If all possible elements of P and R were added at once, the problem would be solved in one iteration. Instead of enumerating all possible extreme ray and point which is very difficult, Bender proposes an iterative procedure which starts with empty sets P and R, and then iteratively solve Dual-SP and MP. Where Dual-SP uses the solution $\bar{\boldsymbol{y}}$ of the MP and the solution $\bar{\boldsymbol{u}}$ of Dual-SP provides 17b if the solution is unbounded otherwise 17c is generated. The process repeats as long as the optimality gap is not closed. As a result, it is often not necessary to find all elements of P and R to find an optimal solution.

According to the previous decomposition, Algorithm 1 provide an ϵ-optimal solution where ϵ is the user-defined acceptable optimality gap.

Algorithm 1: Bender decomposition

Result: ϵ-optimal solution $\bar{\boldsymbol{y}}$ and $\bar{\boldsymbol{w}}$
initialization;
while *UB-LB* $\geq \epsilon$ **do**
  Step 1: Solve MP and obtain $\bar{\boldsymbol{y}}$;
  Step 2: Solve Dual-SP and obtain $\bar{\boldsymbol{u}}$;
  if *SP is unbounded* **then**
    R $\leftarrow$ R $\cup$ { $\bar{\boldsymbol{u}}$ } ;
  else
    P $\leftarrow$ P $\cup$ { $\bar{\boldsymbol{u}}$ } ;
    LB $\leftarrow$ c($\bar{\boldsymbol{y}}$) + $\bar{z}$;
    UB $\leftarrow$ min {UB, c($\bar{\boldsymbol{y}}$) + $z(\bar{\boldsymbol{u}})$};
  end
end

The Bender decomposition methods convert the initial problem into a bi-level min-max problem. In the specific case of an attacker-defender model, we are already dealing with such structure. Thus solving our bi-level interdiction network is equivalent to apply a special case of BD where the decomposition splits attackers and defenders decision variables. Wood (2011) demonstrated the correctness of such methodology and several studies applied it to solve similar problems (Lim and Smith, 2007), (Smith et al., 2007).

4.1.2 Application to R-TCFLP

As mentioned before we use Bender Decomposition to be able to solve our bi-level mixed integer problem. In our case, the dual sub-problem (Dual-SP) correspond exactly to $Post(\boldsymbol{y})$ the post-interdiction flow routing problem. By doing so, the master problem (MP) is single level mixed integer program that we can solve with any commercial solver. At each iteration the following (MP) is solved:

$$Z_{MP} = min_y : \rho\Psi(y) + (1-\rho)Z_\Phi + F(y) \tag{18a}$$

$$s.t : (6b), (6c), (6d), (6e) \tag{18b}$$

$$Z_\Phi \geq z_\Phi(\bar{u}, y) \quad , \forall \pi \in \Pi \tag{18c}$$

$$y_{j1} \in \{0, 1\} \quad , \forall j1 \in V1 \tag{18d}$$

Where constraint (18c) represents the Bender cut added to the master problem at iteration π and Π is the set of generated cut.

At each iteration the problem $Post(y)$ (Equation 8) is solved and the decision variables $u = \beta||\delta||\epsilon$ (|| Concatenation of vector) values are used to define a new Bender cut, we note those values $\bar{u}_\pi = \bar{\beta}_\pi||\bar{\delta}_\pi||\bar{\epsilon}_\pi$. $z_\Phi(\bar{u}, y)$ designate the objective function of $Post(y)$ (see 8a) at iteration π where the decision variables are replaced by$\bar{\beta}_\pi, \bar{\delta}_\pi, \bar{\epsilon}_\pi$ and the decision variable $\boldsymbol{y}$ is set free. The explicit form of the Bender cut at iteration π is the following.

$$Z_\phi \geq \sum_{i \in I} d_i \bar{\beta}_{i\pi} - \sum_{j2 \in V2} (\alpha_{j2} \bar{\delta_{j2}}_\pi) y_{j2} - \sum_{j1 \in V1} (\alpha_{j1} t \bar{\epsilon_{j1}}_\pi) y_{j1} \tag{19}$$

We consider ϵ an acceptable optimality gap, then the lower bound $\underline{z_\phi}$ is provided by the (MP) as the value of the decision variable Z_Φ, while the upper bound $\bar{z_\phi}$ is equal to (Dual-SP) objective value. Finally, we implemented Algorithm 2 to solve our specific problem.

Algorithm 2: Bender decomposition applied to T-TCFLP

Result: ϵ-optimal solution $\bar{y}$ (robust set of facility)
initialization;
while $\bar{z_\phi} - \underline{z_\phi} \geq \epsilon$ **do**
  Step 1: Solve MP and obtain $\bar{y}$ and $\underline{z_\phi}$;
  Step 2: Solve Dual-SP and obtain $\bar{\beta}, \bar{\delta}, \bar{t\epsilon}$ and $\bar{z_\phi}$;
  Step 3: add a new Bender cut $Z_\Phi \geq z_\Phi(\bar{u}, y)$;
end

4.2 Background in Acceleration Methods for BD

The standard Bender decomposition is known for its slow convergence (Saharidis and Ierapetritou, 2010). This is particularly true for network design problems as demonstrated by Magnanti and Wong (1981). Indeed, the deficiency of DB on that type of problem leads to the generation of an exorbitant number of cut in order to converge. The issues of BD in general are (i) The quality of the BD cut at each iteration, (ii) The weak lower bound generated in case of minimization problem Azad et al. (2013), (iii) The deterioration of the solution as the number of cut increase. Decomposition procedures are attractive candidates since a lot of design problem involves at least two types of decisions and it is very tempting to isolate the complicating variable. Therefore Bender decomposition has been extensively studied and researcher developed accelerated versions of BD algorithm.

Early work on the topic such as Geoffrion and Graves (1974) focuses on the importance of finding a proper formulation of the problem that would improve the convergence of the BD algoritm. Later works looked on ways to reduce computation time at each iteration especially when the MP or SP arehHard to solve. For instance McDaniel and Devine (1977) proposed to solve the LP relaxation of MP in a first phase until we reach an acceptable optimality gap and then terminating the algorithm by reintroducing the integrality constraints. Cote and Laughton, 1984 uses heuristics that replace the MP and/or SP. Zakeri et al. (2000) introduces inexact cuts in the MP by not solving the SP to optimality.

The most recent literature offers numerous efficient techniques to accelerate BD algorithm. We can summarize those new techniques applied to improve BD into three categories namely (i) generation of valid inequalities that speed-up the algorithm, (ii) the generation of super-valid inequalities that cut-off non optimal feasible solution such that it reduces the number of iteration, and (iii) the generation of cut bundles such that each iteration generates multiple cuts to further constraint the MP. In our thesis we focuses our efforts on the two first techniques and implemented them into our solution methods. The third technique is not suitable for our problem because SP is relatively difficult while this technique is a tradoff between the number of iteration (i.e solving MP and SP) and the number of cut generated at each iteration (i.e solving MP and multiple time variations of SP) Azad et al., 2013.

Magnanti and Wong (1981) studied the generation of non-dominated cut at each iteration, what they refer to as *Pareto-optimal* cuts. They design a modified version of BD in which at each iteration the Magnanti-Wong cut generation problem is solve using a MP *core point*. Later on Papadakos (2008) proposed to solve the independent Magnanti-Wong problem in order to enhance the Pareto-optimal cut generation and demonstrate that it is not necessary to use a core point to generate Pareto-optimal cut, but instead what he called alternative *Magnanti-Wong point* are suffisant.

Israeli and Wood (2002) to improve the bender decomposition applied the the shortest path network interdiction problem develop a specific type of cut that called super-valid inequalities. The particularity of a SVI is that it removes feasible solution but not the optimal solution, as a result it helps solving the problem by reducing the volume of the convex hull. Later, Wood (2011) generalize the idea for bilevel network interdiction problem.

4.3 Super-Valid Inequalities

In this subsection, the strategy adopted to speed-up the BD algorithm is to generate additional cuts called super-valid inequalities (SVIs) that will strengthen the linear relaxation of MP. The principle of SVIs is introduced in Israeli and Wood, 2002 and extend the theory of valid inequalities in integer-programing theory. Valid inequalities are cutting down the feasible region of a linear relaxation but keep all feasible integer solution, whereas SVIs reduce the size of the integer feasible area and are guaranteed not to eliminate all optimal solutions. In the section we introduce the formal concep of super-valid inqualites and provide two type of SVIs to solve our problem.

Definition 4.3.1 *ibid. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ denote, respectively, the vectors of continuous and integer variables in an MIP, let c_1 and c_2 be two conforming vectors of constants, respectively, and let c_0 be a scalar constant. The inequality $c_1\boldsymbol{x}+c_2\boldsymbol{y} \geq c_0$ is supervalid for this MIP, that is, it is a supervalid inequality for the MIP, if (i) adding that inequality to the MIP does not eliminate all optimal solutions or (ii) an incumbent solution to the MIP, $(\widehat{\boldsymbol{x}},\widehat{\boldsymbol{y}})$, is (already) optimal.*

The SVI are added to MP as the same time as the Bender cut generated. Algorithm 3 shows the enhanced Bander decomposition using super-valid inequalities. This algorithm is composed of two phases. The first phases relax the integrality constraint in the MP and follow the Bender procedure until an ϵ-optimal is found for the relaxed problem. The second phase reintroduce the itegrality constraint and continue the Bender procedure untill we find an ϵ-optimal.

Algorithm 3: SVI: BD and Super-Valid Inequalities applied to T-TCFLP

Result: ϵ-optimal solution $\bar{y}$ (robust set of facility)

Initialization:

Solve MP and obtain initial y_0 ;

Phase 1:

$integrality\_tolerance \leftarrow 0.5$;

while $\bar{z_\phi} - \underline{z_\phi} \geq \epsilon$ **do**

  Step 1: Solve MP and obtain $\bar{\boldsymbol{y}}$ and $\underline{z_\phi}$;

  Step 2: Solve Dual-SP and obtain $\bar{\boldsymbol{t\epsilon}}$ and $\bar{z_\phi}$;

  Step 3: add a new Bender cut $Z_\Phi \geq \Phi_\pi(\boldsymbol{y})$ to MP ;

  Step 4: Add a new SVI to MP ;

end

Phase 2:

Add integer constraint: $integrality\_tolerance \leftarrow 1 \times 10^{-5}$;

while $\bar{z_\phi} - \underline{z_\phi} \geq \epsilon$ **do**

  Step 1: Solve MP and obtain $\bar{\boldsymbol{y}}$ and $\underline{z_\phi}$;

  Step 2: Solve Dual-SP and obtain $\bar{\boldsymbol{t\epsilon}}$ and $\bar{z_\phi}$;

  Step 3: add a new Bender cut $Z_\Phi \geq z_\Phi(\bar{u}, y)$;

  Step 4: Add a new SVI to MP ;

end

4.3.1 Type-I SVI applied to R-TCFLP

The first type of SVIs (Type-I) is directly derived from Wood, 2011. The decision variable of SP $t\bar{\epsilon}_{j1_\pi}$ is equal to 0 if the facility j1 was a target of the interdictor or if the shadow cost associated to the capacity of j1 is null. In that case, at least one of the remaining facilities must be selected. Lets considere the following Bender cut generated at iteration π:

$$Z_\phi \geq \sum_{i \in I} d_i \bar{\beta}_{i\pi} - \sum_{j2 \in V2} (\alpha_{j2} \bar{\delta}_{j2_\pi}) y_{j2} - \sum_{j1 \in V1} (\alpha_{j1} t\bar{\epsilon}_{j1_\pi}) y_{j1}$$

Then according to Wood, 2011 the following inequality is super-valid:

$$\sum_{j1 \in V1} I(t\bar{\epsilon}_{j1_\pi}) y_{j1} \geq 1 \tag{20}$$

$$I(t\bar{\epsilon}_{j1_\pi}) = \begin{cases} 1 & if t\bar{\epsilon}_{j1_\pi} \leq 0 \\ 0 & otherwise \end{cases} \tag{21}$$

Type-I SVI attempt to convert the problem into a purely combinatorial problem. Note that we can also design a lifting procedure for Type-I inequalities. Let $A = (\alpha_{j1} t\bar{\epsilon}_{j1_\pi})_{j1 \in V1}$, we define $S = (s_i)_{i=1}^{|V1|} = sort(A)$ as the unique non-increasing sequence of element of A. We are looking for le largest k such that:

$$\sum_{i \in I} d_i \bar{\beta}_{i\pi} - \sum_{j2 \in V2} (\alpha_{j2} \bar{\delta}_{j2_\pi}) \bar{y_{j2}} - \bar{z_\phi} - \sum_{i=1}^{k} s_i \geq 0 \tag{22}$$

As a result, the previous SVI can be tighten as following:

$$\sum_{j1 \in V1} I(t\bar{\epsilon}_{j1_\pi}) y_{j1} \geq k \tag{23}$$

4.3.2 Type-II SVI applied to R-TCFLP

The second type of SVI (Type-II) uses a particularity of our problem. As a matter of fact, the post-interdiction problem will always be more expensive than the pre-interdiction problem, because the cost of operating the network after removing any facility increase. Thus it is true that for any feasible solution $\bar{y}$ and we can state that $\Psi(\bar{y}) \leq \Phi(\bar{y})$

$$Z_{MP}(\bar{y}) = \rho\Psi(\bar{y}) + (1-\rho)\Phi(\bar{y}) + F(\bar{y}) \tag{24}$$

$$Z_{MP}(\bar{y}) \leq \rho\Psi(\bar{y}) + (1-\rho)\Psi(\bar{y}) + F(\bar{y}) \tag{25}$$

$$Z_{MP}(\bar{y}) - F(\bar{y}) \leq \Psi(\bar{y}) \tag{26}$$

For any given Bender cut π, the following inequality is super-valid for MP:

$$Z_\Phi \geq Z_{MP}(\bar{y}) - F(\bar{y}) \tag{27}$$

Table 5: Example - Problem *6b43* first iteration

j1	1	2	3	4	5	6	7	8	9	10
$t\epsilon_{j1}$	7,755	7,077	0	8,115	7,200	10,385	0	7,755	6,164	9,020
α_{j1}	21	8	19	17	13	10	21	15	13	9

Each iteration provides a Type-II, resulting in a strong lower bound for Z_Φ that will remove inadequate solutions and faster the convergence ot this algorithm.

4.3.3 Illustration of Type-I and Type-II SVI

Here, we demonstrate on an example the application of Type-I and Type-II for a given iteration of the Bender Decomposition algorithm. As an example we consider the dataset *6b43* that contain 10 customers, 10 potential facilities and 5 suppliers. The first iteration of this problem provide the following bender cut:

$$Z_\phi + \sum_{j1 \in V1} (\alpha_{j1} t\bar{\epsilon}_{j1_\pi}) y_{j1} \geq 659310$$

The coefficients α_{j1} and $t\bar{\epsilon}_{j1_\pi}$ are summarized in table 5. Additional informations are summarize below.

$$\begin{aligned} \bar{y} &= [2, 6, 7, 8] \\ Z_{MP}(\bar{y}) &= 322209 \\ F(\bar{y}) &= 285999 \\ \sum_{i \in I} d_i \bar{\beta}_{i\pi} - \sum_{j2 \in V2} (\alpha_{j2} \bar{\delta}_{j2_\pi}) \bar{y_{j2}} &= 659310 \end{aligned}$$

This Bender cuts result into a lifted Type-I SVI where the sum of y_{j1} with $ji \in V1$ $\{3, 7\}$ must be greater or equal to 2 and a Type-II SVI defining a lower bound for Z_Φ

$$Type - I: \qquad y_1 + y_2 + y_4 + y_5 + y_6 + y_8 + y_9 + y_{10} \geq 2 \tag{28a}$$

$$Type - II: \qquad Z_\Phi \geq 36210 \tag{28b}$$

4.4 Pareto Optimal cuts

In this subsection, the strategy adopted to speed-up the BD algorithm is adding a non-dominated cut at each iteration. That particular type of cut is called Pareto-Optimal.

Definition 4.4.1 *Pareto-Optimal (Magnanti and Wong, 1981) A cut is called Pareto-optimal if no other cut dominate it. Since, a cut is defined by the point* $\boldsymbol{u_0}$*. This is said Pareto-optimal.*

Magnanti and Wong (ibid.) created a method to systematically find Pareto-Optimal cut by using a core point of the problem studied.

Definition 4.4.2 *Core Point (ibid.)* $\boldsymbol{y_0}$ *is a core point of* Y *if and only if it is contained in the relative interior* $\boldsymbol{y_0} \in ri(Y^c)$ *of the convex hull* Y^c *of* Y*.*

However Magnanti and Wong (ibid.) cut generation method is difficult to solve and may reduce the performance of the algorithm. Instead, we used in our algorithm the Papadakos cut generation problem (Papadakos, 2008) also called Independent Magnanti-Wong problem, because it has better performances. Moreover, Papadakos methods elegantly fits our needs, because it is equivalent to the Bender subproblem $\Phi(y)$ for a core point of our problem.

Definition 4.4.3 *Papadakos cut generation problem (ibid.) Let* $\boldsymbol{y_0}$ *be a core point of* Y*, then the optimal solution solution* $\boldsymbol{u_0}$ *of* $\Phi(y_0)$ *is pareto optimal*

Another difficulty of Magnanti and Wong (1981) methods is finding a core point. Fortunately Papadakos, 2008 also demonstrate that a core point is not necessary to generate a Pareto-Optimal cut. He introduces instead the Magnanti-Wong Point that is sufficient to generate Pareto-Optimal cuts.

Definition 4.4.4 *Magnanti-Wong Point (ibid.) A Magnati-Wong point is a point* $\boldsymbol{y_0}$ *for which the solution of* $\Phi(y_0)$ *gives a Pareto-Optimal sulution* $\boldsymbol{u_0}$

Finally, Magnanti and Wong, 1981 provided multiple techniques to generate Magnanti-Wong Point. We use Equation 29 to generate approximate Magnanti-Wong Point.

The approximate point of the next iteration $y_0^{\pi+1}$ is the combination of the previous approximate Magnanti-Wong Point y_0^{π} and the previous solution of MP $\bar{y}^{\pi}$.

$$y_0^{\pi+1} = \frac{1}{2}y_0^{\pi} + \frac{1}{2}\bar{y}^{\pi} \tag{29}$$

Algorithm 4 was implemented using Equation 29 where $y_0^{\pi=1}$ has all its elements equals to 1 at the first iteration. This algorithm is similar to Algorithm 3 in the sense that it is composed of two phases: (i) relaxation of the integrality constraint in MP, (ii) reintroducing integrality constraint in MP. The differences are located in step 1 and 5.

Algorithm 4: BD and Independent Magnanti-Wong problem applied to R-TCFLP

Result: ϵ-optimal solution $\bar{y}$ (robust set of facility)
Initialization:
Solve MP and obtain initial y_0 ;
Phase 1:
$integrality\_tolerance \leftarrow 0.5$;
while $UB\text{-}LB \geq \epsilon$ **do**
  Step 1: Solve I-MW and obtain cut $Z_\Phi \geq \Phi_{y_0,\pi}(y)$;
  Step 2: Solve MP and obtain $\bar{\boldsymbol{y}}$ and $\underline{z}_\phi$;
  Step 3: Solve Dual-SP and obtain $\bar{\boldsymbol{t\epsilon}}$ and $\bar{z}_\phi$;
  Step 4: Add a new Bender cut $Z_\Phi \geq z_\Phi(\bar{u}, y)$;
  Step 5: Update $y_0 \leftarrow \frac{1}{2}y_0 + \frac{1}{2}\bar{y}$
end
Phase 2:
Add integer constraint: $integrality\_tolerance \leftarrow 1 \times 10^{-5}$;
while $UB\text{-}LB \geq \epsilon$ **do**
  Step 1: Solve I-MW and obtain cut $Z_\Phi \geq \Phi_{y_0,\pi}(y)$;
  Step 2: Solve MP and obtain $\bar{\boldsymbol{y}}$ and $\underline{z}_\phi$;
  Step 3: Solve Dual-SP and obtain $\bar{\boldsymbol{t\epsilon}}$ and $\bar{z}_\phi$;
  Step 4: Add a new Bender cut $Z_\Phi \geq z_\Phi(\bar{u}, y)$;
  Step 5: Update $y_0 \leftarrow \frac{1}{2}y_0 + \frac{1}{2}\bar{y}$
end

Chapter 5

Numerical Experiments

We perform a series of numerical experiments to evaluate the performance of our proposed model and solution methods. We tested all algorithm on three datasets containing 49, 88 nodes. The datasets are directly extracted from Snyder and Daskin (2004). We used our solution method on networks for $|V2|$ equals 5 suppliers located in a randomly selected city based on Li et al. (2013) which states that five to eight distributor is a reasonably large regional distribution network for up to 150 customers. Each dataset size was solved over 5 instances with different randomly selected supplier location. In all datasets the set of customer I is equal to the set of potential warehouse V1, in other words, each city can host both customer and facility.

Tables 6 and 7 provides a description of the instances we used in this study. They displays TCFLP solution in terms of cost and selected intermediate facilities.

problem	id	$\|V1\|$	$\|V2\|$	$F(\bar{y})$ \$k	$F(\bar{y}) + \Psi(\bar{y})$ \$k	$\Psi(\bar{y})$ \$k	$\bar{y}$
P0	1276	49	5	263	1,620	1,356	11, 38, 40, 42, 47
P1	1277	49	5	268	817	549	3, 9, 13, 30, 35
P2	1278	49	5	340	1,576	1,236	1, 10, 18, 21, 22
P3	1279	49	5	197	1,792	1,594	6, 12, 31, 33
P4	127a	49	5	379	1,006	627	6, 10, 19, 21, 23, 43

Table 6: 49-node instances

The algorithms are coded in Python, uses CPLEX version 12.8 as a linear programming solver and run on Windows 7 SP1 PC with 3.20 GHz 2 Duo core CPU and 16.0 GB of physical RAM. The gap tolerance is set at 5% and the algorithm after

problem	id	$\lvert V1 \rvert$	$\lvert V2 \rvert$	$F(\bar{y})$ $k	$F(\bar{y}) + \Psi(\bar{y})$ $k	$\Psi(\bar{y})$ $k	$\bar{y}$
P5	127c	88	5	236	2,893	2,656	8, 49, 54, 59, 75
P6	127d	88	5	248	2,937	2,689	6, 9, 15, 46, 66
P7	127e	88	5	380	2,040	1,660	11, 33, 52, 75, 77, 78
P8	127f	88	5	320	2,709	2,388	22, 28, 31, 49, 71
P9	1280	88	5	246	3,142	2,895	7, 11, 30, 48, 85

Table 7: 88-node instances

7200s if it did not attain the tolerance gap.

Parameter values for our datasets are given in Table 8, those values are extracted from Snyder and Daskin (2005).

Description	notation	value
Transportation cost	$c_{a,b}$	Euclidean distance between city a and city b
Demand	d_i	equal to the state population divided by 10^5
Fixed cost	f_{j1}	Median home value in the city
Emergency cost	e	10^4
Optimality tolerance	ϵ	0.05

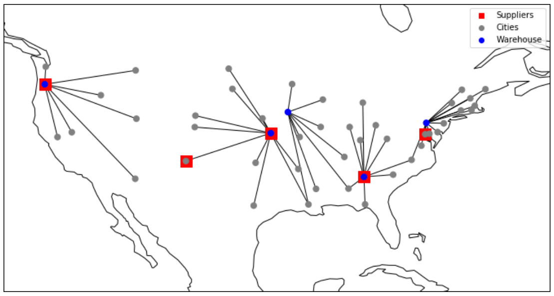
Table 8: Parameters for datasets

5.1 Benefit of Considering Facility Interdiction

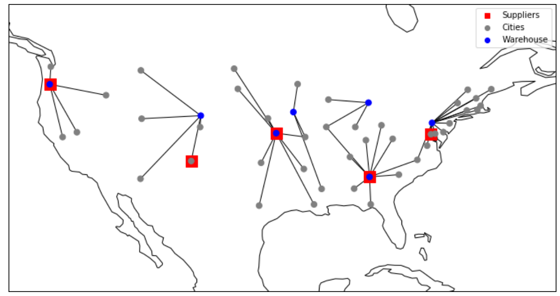
In this section, we investigate the benefit of the proposed design model. First, the proposed methods are illustrated on a single example to visualize and compare its solution to the classical TCFLP, and then we analyze the impact of parameters such as the weight of the post-interdiction (ρ), and the attacker budget (B) on the decision.

5.1.1 Topology and Cost Comparison

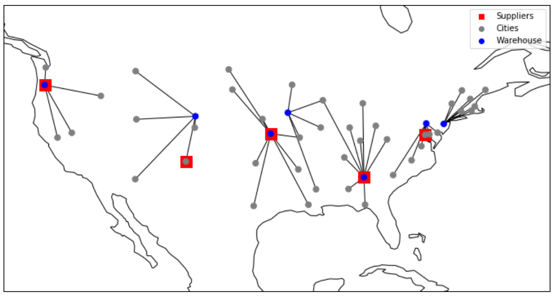
In this subsection, we study the benefits of considering potential disruption resulting from a targeted attack while designing a supply chain network. For this purpose, we look at the robustness of a solution for different values ρ and budget B. We use a detailed example derived from the 49-node dataset.



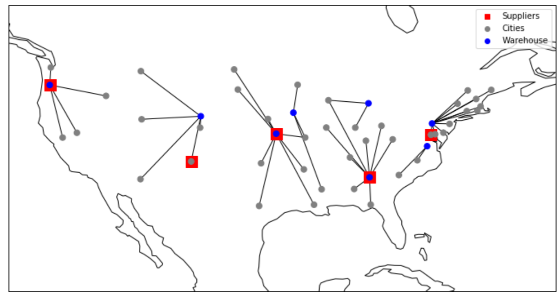
(a) TCFLP



(b) R-TCFLP $\rho = 0.99$, B=2



(c) R-TCFLP $\rho = 0.90$, B=2

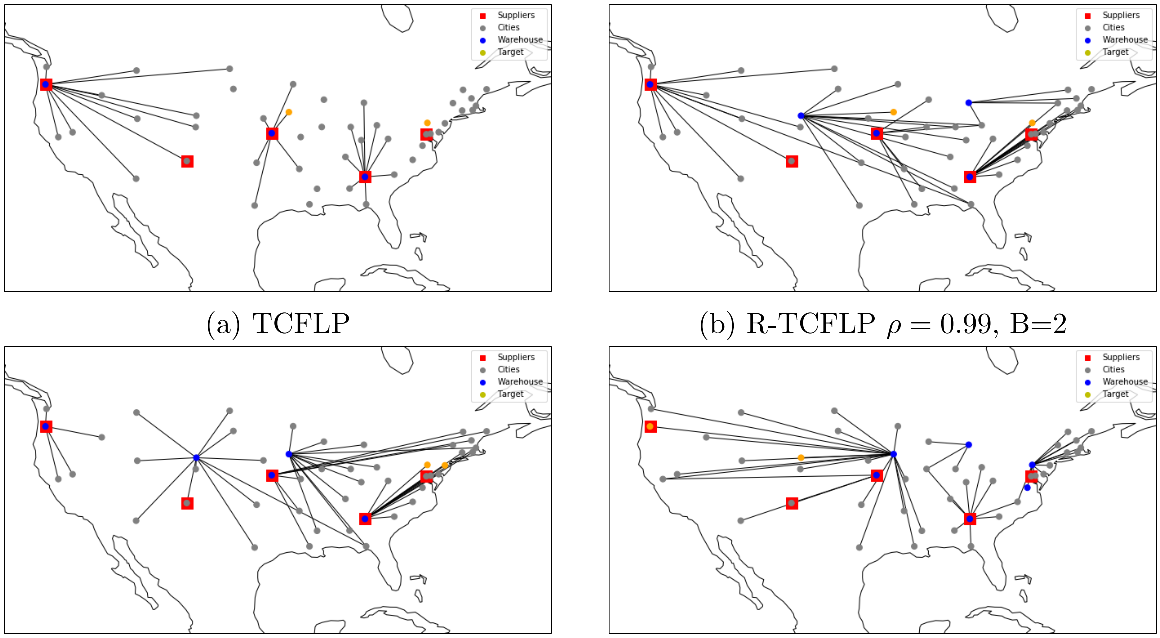


(d) R-TCFLP $\rho = 0.80$, B=2

Figure 3: Pre-Interdiction flow routing

This experiment is designed as follows: first, we consider problem *P1* of the 49-node dataset which serves the best our explanation and solves TCFLP using this dataset. The topology of this solution is represented in Figure 3a. Then we compute the best possible attack on TCFLP solution and the post-interdiction routing using the attacker-defender model. Figure 4a represents the topology of the network after an attack on TCFLP solution. We applied the same procedure to compare R-TCFLP solutions for different values of ρ and TCFLP solution. Figure 3 represent the topology of the network designed using TCFLP and R-TCFLP with ρ equals to 0.99, 0.90, and 0.8. Each solution can be compared with their topology after an intelligent attack in figure 4. For this experiment, we assume that the attacker as sufficient budget to eliminate two facilities (i.e., $B = 2$).

Table 9 display the selected facilities and the costs associated to each of the four solutions studied in this section. TCFLP selects five intermediate facilities resulting in an annualized fixed cost of \$268,200 and an annual transportation cost of \$549,008. However, in the hypothesis of a targeted attack, this design will undergo tremendous financial loss. In that event, the designer implements mitigation operation by reassigning the customers to a surviving facilities; however the impossibility of satisfying



(a) TCFLP

(b) R-TCFLP $\rho = 0.99$, B=2

(c) R-TCFLP $\rho = 0.90$, B=2

(d) R-TCFLP $\rho = 0.80$, B=2

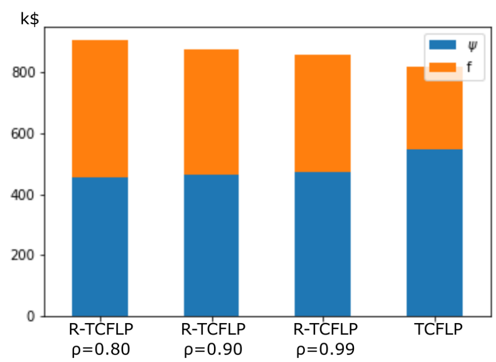
Figure 4: Post-Interdiction flow routing

the demand facilities result in a tremendous increase is post-interdiction cost. The post-interdiction cost is 37 times larger than the pre-interdiction cost with this model. In that specific example, 40% of the demand can not be met, and 20 cities out of 49 are ignored in the disrupted distribution network. This situation is represented in Figure 4a. A tightly optimized solution in a capacitated case, result in catastrophic financial impact due to the limited capacity of the remaining intermediate facilities.

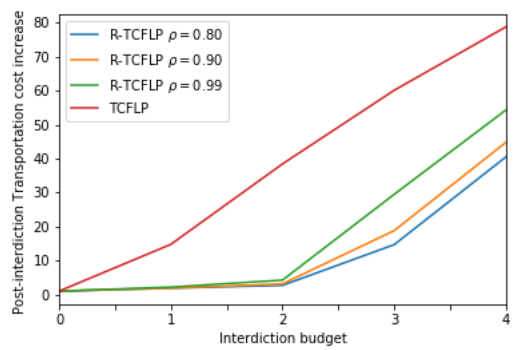
	solution	F	Φ	Ψ
R-TCFLP $\rho = 0.80$	3, 9, 13, 15, 24, 30, 35, 41	451,900	1,227,128	455,039
R-TCFLP $\rho = 0.90$	3, 9, 13, 24, 25, 30, 35	408,200	1,459,292	466,041
R-TCFLP $\rho = 0.99$	3, 9, 13, 15, 24, 30, 35	385,300	2,009,783	473,741
TCFLP	3, 9, 13, 30, 35	268,200	21,102,407	549,008

Table 9: Solutions and costs for problem *P1*

A quantitative evaluation of those four solutions are displayed in Figure5 with respect to annual investment $\Psi(\vec{y}) + F(\vec{y})$ and in terms of post-interdiction cost $\Phi(\vec{y})$. A robust solution generated with R-TCFLP contains between 7 and 8 facilities depending on the importance given to the threat (i.e., the closer to ρ is from 1, the less the threat is taken into account). It is reasonable to say that we are selecting backup facilities to prevent the losses in case of major disruption, because if we compare the solution in Table 9 we see that the facilities selected by TCFLP are also in R-TCFLP solution. However, the location of the backup facilities may change based on the importance of the threat. Adding intermediate facilities to the network comes at a cost. For instance, in the robust design of the experiment, the fixed cost (F) increases from 40% to 70% compared to TCFLP (Table 9), however having more intermediate facilities also reduce the transportation cost due to the fact that each facility delivers customers in a smaller area. As a result, the total annual cost (figure 5a) of designing a robust network increase by 5% to 10% compared to TCFLP. In return, in the event of an attack the post-interdiction cost increase by 2 to 3 times compared to the pre-interdiction cost against 37 times in the case of TCFLP.



(a) Design costs



(b) Post interdiction

Figure 5: The cost of facility interdiction

5.1.2 Impact of the Weight Attributed to The Disruption

To compare solutions for different value of ρ we would like to know how many days of disruption the network should be exposed to in order to observe potential savings. In this experiment, we consider three values for ρ and look at the difference in investment and robustness of the solution obtained for the problem *P1* studied in the previous section.

We define the relative investment ΔI as the difference between the annualized cost without disruption for R-TCFLP and TCFLP objective value. let $\bar{\boldsymbol{y}}^R$ be R-TCFLP optimal solution and $\bar{\boldsymbol{y}}$ TCFLP optimal solution then ΔI is given by Equation 30.

$$\Delta I = \Psi(\bar{\boldsymbol{y}}) + F(\bar{\boldsymbol{y}}) - [\Psi(\bar{\boldsymbol{y}}^R) + F(\bar{\boldsymbol{y}}^R)] \tag{30}$$

We also define relative post-interdiction Δr which is given by Equation 31.

$$\Delta r = -[r(\bar{\boldsymbol{y}}) - r(\bar{\boldsymbol{y}}^R)] \tag{31}$$

We are looking to evaluate the least recovery time such that selecting the robust solution is financially beneficial. In other words, if the disruption time over a year is higher than d, then the robust solution is profitable.

$$d = \frac{\Delta I}{\Delta r} \tag{32}$$

	ΔI \$	Δr \$k	d days
R-TCFLP $\rho = 0.80$	89,730	19,781	2
R-TCFLP $\rho = 0.90$	57,033	19,560	1
R-TCFLP $\rho = 0.99$	41,833	19,017	1

Table 10: Comparison of Investment for problem *P1*

Table 10 demonstrates that the investment in robust network creates saving if the disruption last a minimum of 1 or 2 days per year. We perform this analysis of the least recovery time before return on investment on all instances and summarises them in Table 11.The solutions generated in this example protects the network against the interdiction of two facilities (i.e., B=2), and we look at the performances of the network against worst case interdictions from 1 up to 4 facilities (B*). A solution obtain with R-TCFLP will be viable in less than two weeks if the enemy attacks a single facility, and less than two days if the enemy target more than one facility. Moreover, when two facilities are interdicted where the network is designed to be robust against an attacker with a budget of two the resulting value of d is about the same no matter the value of ρ.

Tables 12 and 13 in there first column displays the solution of R-TCFLP for each instance and different values of ρ. One should notice that we observe minor changes a ρ changes. The changes consist of adding an extra facility or replacing some facilities

B*	1			2			3			4		
ρ	0.99	0.90	0.80	0.99	0.90	0.80	0.99	0.90	0.80	0.99	0.90	0.80
49-node	6	11	14	1	2	3	2	2	2	1	2	3
88-node	6	9	15	1	1	2	1	1	1	1	1	6

Table 11: Average annual disruption time before return on investment

		B* (# facility interdicted)	0	1	2	3	4
	rho	solution	k$	%	%	%	%
P0	0.99	11, 28, 29, 38, 40, 42	1,316	14	55	1046	2026
	0.9	9, 28, 29, 38, 40, 42	1,310	10	42	1051	2036
	0.8	11, 18, 28, 38, 40, 42, 47	1,310	5	24	346	1317
P1	0.99	3, 9, 13, 15, 24, 30, 35	473	119	324	2849	5331
	0.9	3, 9, 13, 24, 25, 30, 35	466	88	213	1781	4379
	0.8	3, 9, 13, 15, 24, 30, 35, 41	455	90	170	1369	3950
P2	0.99	1, 7, 10, 21, 34, 45	1,235	99	122	1239	2272
	0.9	0, 10, 21, 22, 27, 34, 38, 45	1,188	12	39	406	1366
	0.8	0, 10, 21, 22, 27, 38, 45, 46	1,188	12	34	406	1366
P3	0.99	6, 12, 21, 31, 33, 36	1,524	5	78	861	1674
	0.9	6, 12, 15, 31, 33, 36, 38	1,518	6	24	480	1274
	0.8	6, 12, 18, 21, 31, 33, 36	1,512	4	20	344	1134
P4	0.99	6, 10, 19, 21, 23, 32, 41	569	55	644	2855	4931
	0.9	6, 10, 19, 21, 23, 32, 43, 47	573	21	123	1409	3323
	0.8	6, 10, 19, 21, 23, 32, 43, 47	573	21	123	1409	3323

Table 12: Comparison of solution depending on ρ - 49-node

while the total number of facilities remain the same. The remaining column of those tables represents first the cost of distributing the goods in the network under normal condition $\Psi(\bar{\boldsymbol{y}})$ then the percent post-interdiction increase when optimally targeting 1 to 4 facilities. We can see that some problem are more robust, because the percent increase is smaller (P0 and P3 for instance).

5.1.3 Impact of the Attacker Budget

In this subsection, we analyze the impact of choosing different attacker budgets when designing a network using R-TCFLP. In this experiment, we are only concerned by the 49-node datasets and ran the algorithm for B varying between 1 and 3. The parameter ρ takes the values 0.99, 0.90, 0.80. As a result, this experiment compares

	rho	B* (# facility interdicted) solution	0 k$	1 %	2 %	3 %	4 %
P5	0.99	1, 7, 8, 49, 54, 59, 75	2,619	5	30	833	1732
	0.9	1, 7, 48, 54, 59, 70, 73	2,608	6	24	760	1488
	0.8	1, 7, 8, 41, 49, 54, 59, 73, 75	2,612	4	9	30	211
P6	0.99	6, 8, 9, 15, 46, 66	2,686	4	51	909	1741
	0.9	6, 7, 8, 9, 15, 46, 66	2,657	4	18	52	920
	0.8	6, 8, 9, 15, 30, 46, 59, 66	2,677	4	8	42	583
P7	0.99	4, 11, 36, 41, 43, 52, 78	1,695	17	123	1243	2387
	0.9	11, 33, 48, 52, 56, 66, 75	1,639	13	81	1363	2715
	0.8	4, 11, 26, 33, 43, 46, 48, 52, 66, 75	1,632	7	32	73	131
P8	0.99	3, 22, 28, 31, 49, 71, 80	2,341	20	41	517	1380
	0.9	22, 28, 31, 36, 49, 71, 80	2,338	20	34	386	1260
	0.8	22, 28, 31, 36, 49, 71, 80	2,338	20	34	386	1260
P9	0.99	7, 11, 12, 25, 48, 85	2,916	16	47	821	1594
	0.9	7, 11, 30, 32, 48, 77, 85	2,876	2	21	538	1204
	0.8	7, 11, 25, 30, 32, 48, 51, 85	2,873	2	10	31	465

Table 13: Comparison of solution depending on ρ - 88-node

nine decisions corresponding to all possible pairs (B,ρ).

Figure 6 represents ratio of the relative investment ΔI to the optimal objective function value of TCFLP for each of the nine designs. It is clear that the higher B, the more significant the investment in robustness. Similarly, the smaller ρ, the lower the investment. On top of that, the investment seems to increase linearly for a given ρ.

Figure 7 compare the robustness of each solution when removing 1, 2, 3 or 4 facilities. It displays the ratio Δr to TCFLP solution's post-interdiction cost. Therefore if a point value equals to 1, then it means that the solution has the same perfomances as TCFLP.

When optimally removing one facility (Figure 7a) all the solution are performing equally and reduce $r(y)$ by about ten times.

Similarly, when optimally removing two facilities, the performances of solutions designed to sustain an attacker budget of two or three also reduce $r(y)$ by about ten times (Figure 7b). However, the performances of the solutions designed with $B = 1$ drop, but remain better than TCFLP. It is also important to notice that the results

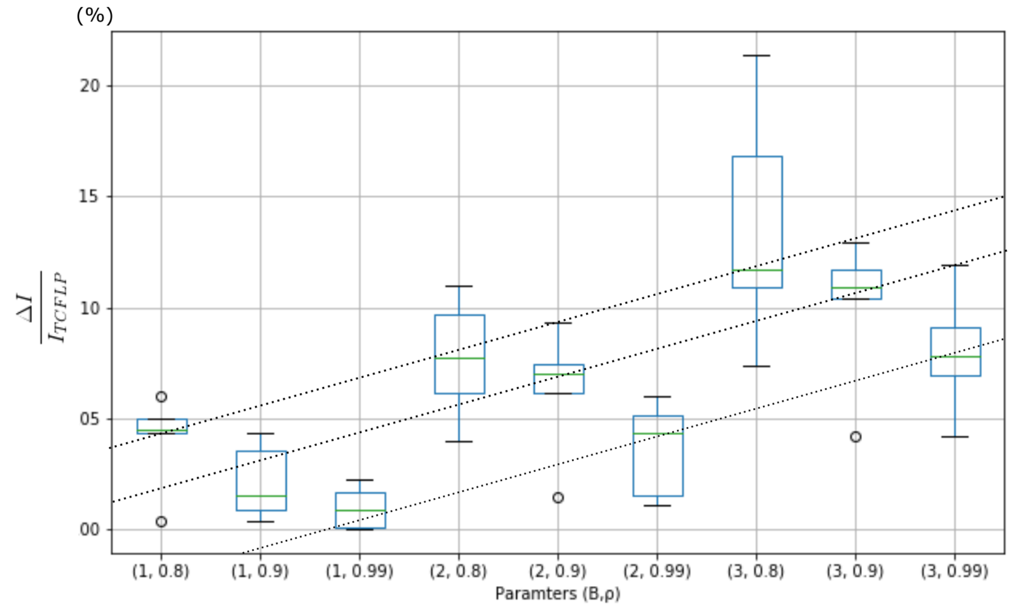
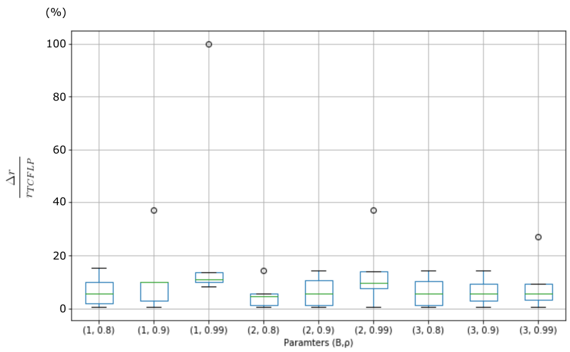


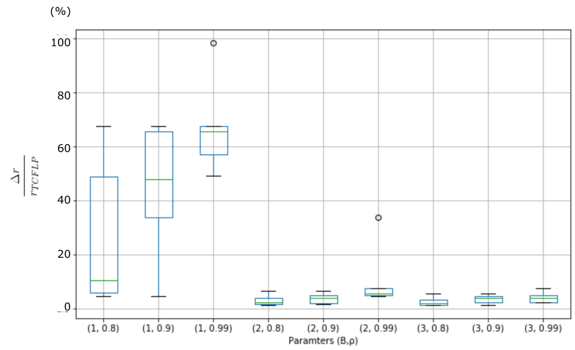
Figure 6: Relative investment for multiplies values of B and ρ

for (1,0.80), (1,0.90), (1,0.99) are extremely spread, they are going from 0.05 to 1 depending on the instances. This can be explain vy the fact that the initial design did not have enught slack to to accomodate interdictions greater than 1 facility

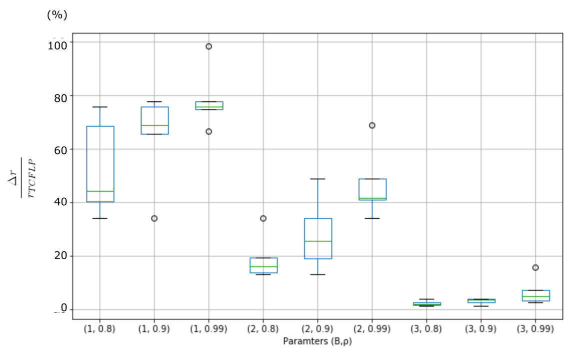
Figure 7c and Figure 7d are very similar and illustrates how the solution performs when optimally removing three or four facilities. In those case, we see a pattern similar to Figure 6: the performances linearly decreases when B increase and ρ decreases.



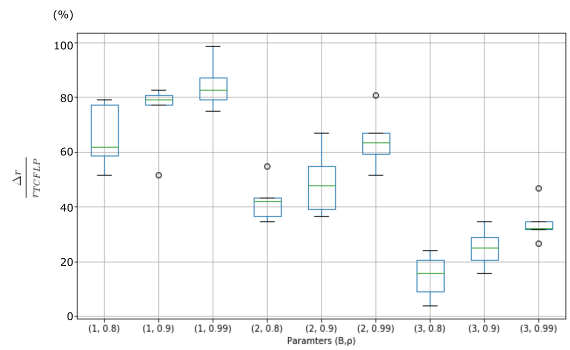
(a) 1 facility removed



(b) 2 facilities removed



(c) 3 facilities removed



(d) 4 facilities removed

Figure 7: Relative post-interdiction cost reduction for multiplies values of B and ρ

5.2 Comparison Between Solution Methods

In this section, we present result based on instances described in Table 6 and 7. We show in which extent the proposed methods improve the efficiency of the standard Bender Decomposition. We compare four types of algorithms listed in Table 14. We aim primarily to compare BD to MW and SVI; however, both use the linear relaxation of the master problem (LR). Therefore we also include in our study LR algorithm to better assess the performance of MW and SVI. Indeed, that way we know the impact of relaxing the integrability constraint in the master problem independently of the cuts added by the two other methods in the Bender decomposition.

Detailed result for each of the four algorithm are listed in Tables 16 and 17. Each table provides performances regarding CPU time (T (s)) and the number of iteration (Iter.) as well as gaps for both the Bender Decomposition (BD gap) and the overall problem R-TCFLP (overall gap). BD gap is computed using BD LB corresponds to the biggest value of Z_ϕ in the master problem, and BD UB that corresponds to

Abbr.	Designation
BD	Standard Bender Decomposition
LR	Bender Decomposition with linear relaxation of the master problem
MW	Bender Decomposition with Magnanti-Wong pareto-optimal cuts
SVI	Bender Decomposition with super-valid inequalities

Table 14: List and designation od compared algorithm

the smallest solution of the subproblem $\phi(y)$. Similarly, the overall gap is computed using the overall LB which corresponds to the biggest value of the objective function of R-TCFLP during the Bender Decomposition (see Equation 18a), while the UB is an evaluation of the objective function given the solution $\bar{y}$ (see Equation 2).

The renaming of this section provides a comparison between each of the four algorithms in terms of speed, quality of solution and convergence of the algorithm.

5.2.1 Comparison of CPU Time and Number of Iteration

The standard Bender decomposition (BD) solved 27 out 30 datasets within the 7 200-seconds time limit (i.e 90% of the problems), only three cases where $\rho = 0.8$ reached the time limit. The enhanced methods can solve the same number problem, therefore we can conclude that an instance that is hard for BD is also for LR, SVI, and MW. Table 15 summarize the average CPU time and the number of iteration per problem size and post-interdiction weight for each of our four algorithms. First of all, we can see in this table that the complexity of all algorithm increase for instances with a higher problem size $|V1|$ and a smaller value of ρ.

		T (s)				Iter.			
		BD	LR	SVI	MW	BD	LR	SVI	MW
$\|V1\|$	ρ								
49	0.8	225	123	145	74	75	84	86	45
	0.9	105	57	49	32	48	57	56	35
	0.99	21	13	15	13	14	28	28	19
88	0.8	5,843	5,937	5,342	3,897	100	129	123	72
	0.9	2,045	1,097	879	618	69	87	80	45
	0.99	91	56	54	46	14	29	26	18
Avg. var.			-35.8%	-36.9%	-54.1%		+49.6 %	+43.3%	-9.1%

Table 15: CPU time and number of iterations

We observe that relaxing the integrability condition on the master problem (LR) already improve significantly the performance in terms of CPU time, while the number of iteration increases. The better performance of LR can be partially explained by the fact that the *relaxed* MP is easier to solve, in consequence, we generate a greater number of cut in a smaller amount of time. According to table 15, LR requires on average about 50% more iterations while performing 36% better in terms of CPU time (15). Another explanation to LR performances is that more decision variables covered at each iteration. Indeed relaxing the integrality constraint in MP allow the decision variable y_{j1} associated to open a facility to can take any real value between 0 and 1. This directly impact the fixed cost associated to a facility j1 equals to $f_{j1} \times y_{j1}$ and its capacity equals to $\alpha_{j1} \times y_{j1}$. In consequence, we can choose the capacity we need for a facility and at the same time reduce the cost associated, which encourage the designer to open more facilities closer to the customer. As a result, the *relaxed* MP solution contains more non zero value for the vector y, then SP provides tighter coefficient for those non zero values, thus the Bender cut generated covers more variables. As Saharidis and Ierapetritou, 2010 emphasizes it is important for the Bender Decomposition to cover more decision variables to better restrict the solution space of the decomposed problems.

The SVI algorithm performances are similar to the LR algorithm, meaning that the inequalities generated do not improve the efficiency of this algorithm significantly on the problem studied. We only observe on average 1% improvement in terms of CPU time compared to LR. We believe that this is due to the emergency facility contained in the problem. Indeed because of this emergency facility, every Type-I SVI will imply that at least one facility is open and the emergency facility is always the best candidate because it is inexpensive. On top of that it is not always possible to find a lifted Type-I SVI, therefore some iteration results in weak SVI. On the other hand, MW demonstrates a good improvement in reducing both the CPU time (54.1% average CPU time reduction) and the number of iteration (9.1% average number of iteration reduction).

5.2.2 Comparison of Bounds

When the problem is solved with an ϵ-optimal solution while $\epsilon = 0.05$, we obtain very good overall bounds. Table 18 outline the average performance regarding the

		BD				LR				MW				SVI			
	rho	BD gap %	overall gap %	T s	Iter.	BD gap %	overall gap %	T s	Iter.	BD gap %	overall %	gap T s	Iter.	BD gap %	overall gap %	T s	Iter.
P0	0.99	0.0	0.0	6	10	0.0	0.0	7	27	0.0	0.0	7	18	0.0	0.0	7	25
	0.90	0.7	0.1	31	31	1.7	0.2	24	41	0.8	0.1	25	35	0.0	0.0	32	49
	0.80	4.5	0.8	105	56	3.4	0.6	88	83	3.2	0.6	60	46	0.2	0.0	165	97
P1	0.99	0.0	0.0	5	7	0.0	0.0	4	19	0.0	0.0	5	15	0.0	0.0	4	17
	0.90	0.9	0.1	30	22	0.0	0.0	26	47	0.0	0.0	14	23	1.0	0.2	23	40
	0.80	3.4	0.8	115	43	2.8	0.6	84	64	1.4	0.3	69	40	2.4	0.5	85	68
P2	0.99	0.0	0.0	84	36	0.0	0.0	36	38	0.0	0.0	34	27	0.0	0.0	48	37
	0.90	3.5	0.3	291	98	3.5	0.3	147	79	4.1	0.4	66	47	3.6	0.3	113	75
	0.80	4.6	0.8	570	134	3.6	0.7	275	109	0.9	0.2	125	54	4.9	0.9	205	87
P3	0.99	0.0	0.0	3	9	0.9	0.0	5	25	4.8	0.1	5	14	1.4	0.0	7	34
	0.90	0.0	0.0	102	54	3.8	0.4	45	55	4.4	0.4	27	31	2.3	0.2	38	62
	0.80	1.6	0.3	175	82	4.9	0.9	80	77	1.0	0.2	44	38	3.4	0.6	109	78
P4	0.99	2.1	0.1	8	6	0.0	0.0	12	29	0.0	0.0	15	22	0.0	0.0	12	25
	0.90	4.2	0.5	74	33	4.0	0.4	43	62	3.0	0.3	29	37	1.0	0.1	40	52
	0.80	4.2	0.9	160	59	1.5	0.3	87	89	1.6	0.3	74	49	1.0	0.2	159	100

Table 16: 49-node experimental results

		BD				LR				MW				SVI			
	rho	BD gap %	overall gap %	T s	Iter.	BD gap %	overall gap %	T s	Iter.	BD gap %	overall %	gap T s	Iter.	BD gap %	overall gap %	T s	Iter.
P5	0.99	0.6	0.0	26	23	0.0	0.0	29	8	0.3	0.0	26	12	0.0	0.0	34	20
	0.90	2.1	0.2	469	73	4.0	0.4	1,519	73	2.5	0.3	459	42	5.0	0.6	520	73
	0.80	3.5	0.6	5,346	133	4.0	0.7	4,863	105	5.0	0.9	2,225	64	19.1	1.0	7,219	135
P6	0.99	3.3	0.0	23	29	0.0	0.0	83	16	0.0	0.0	26	15	4.0	0.1	21	24
	0.90	0.4	0.0	820	82	3.3	0.3	1,012	55	4.7	0.5	230	36	2.1	0.2	587	80
	0.80	4.2	0.8	2,221	102	4.3	0.8	2,415	72	2.2	0.4	1,883	69	3.9	0.7	1,253	90
P7	0.99	0.0	0.0	117	28	0.0	0.0	170	16	0.0	0.0	90	22	0.0	0.0	135	27
	0.90	1.8	0.2	2,463	105	4.1	0.6	4,600	99	4.5	0.6	1,289	48	4.6	0.6	2,053	93
	0.80	10.3	8.9	7,395	128	28.2	5.4	7,341	90	42.0	2.6	7,262	79	42.9	2.0	7,275	115
P8	0.99	0.0	0.0	30	28	0.0	0.0	27	11	0.0	0.0	32	21	0.0	0.0	28	27
	0.90	4.0	0.4	296	74	0.0	0.0	564	41	0.0	0.0	287	39	0.0	0.0	305	70
	0.80	49.6	7.0	7,326	141	2.4	0.2	7,277	116	4.3	0.9	877	54	0.0	0.0	3,741	133
P9	0.99	0.1	0.0	83	35	0.0	0.0	144	18	0.1	0.0	58	22	0.1	0.0	53	31
	0.90	3.9	0.4	1,438	102	1.2	0.1	2,531	77	1.1	0.1	827	59	1.0	0.1	932	83
	0.80	19.6	7.7	7,397	142	26.0	2.7	7,318	117	0.4	0.1	7,239	96	15.7	1.1	7,224	142

Table 17: 88-node experimental results

optimality gap and the overall gap for instances of the same size and value ρ for each of the tested algorithms. All that dataset are solved to optimality except for three cases for 88-node and $\rho = 0.8$, therefore for the problem solve to optimality Table 18 shows that the average overall gap is less than 1% even though the optimality tolerance of the Bender decomposition is 5%.

		Bender Gap				Overall Gap			
		BD	LR	MW	SVI	BD	LR	MW	SVI
cities	rho	%	%	%	%	%	%	%	%
49	0.8	3.6	3.3	1.6	2.4	0.7	0.6	0.3	0.5
	0.9	1.9	2.6	2.4	1.6	0.2	0.3	0.2	0.2
	0.99	0.4	0.2	1.0	0.3	0.0	0.0	0.0	0.0
88	0.8	13.0	17.4	10.8	16.3	2.0	5.0	1.0	1.0
	0.9	2.5	2.4	2.6	2.5	0.3	0.3	0.3	0.3
	0.99	0.0	0.8	0.1	0.8	0.0	0.0	0.0	0.0

Table 18: Bender Optimality Gap and Overall Gap

Of the three problems not solved to optimality by BD (cf. problem *P7*, *P8*, *P9*) even though the bender gap seems important (up to 49 %), once again the overall gap is very good (less than 5%) for both improved methods. Table 19 summarize the results in terms bound for the two problems we are discussing. We also must notice that MW and SVI are able to solve *P8* problem to optimality, and *P9* can be solved using MW.

	Bender Gap				Overall Gap			
	BD	LR	MW	SVI	BD	LR	MW	SVI
	%	%	%	%	%	%	%	%
P7	10.3	28.2	42.0	42.9	8.9	5.4	2.6	2.0
P8	49.6	2.4	4.3	0.0	7.0	0.2	0.9	0.0
P9	19.6	26.0	0.4	15.7	7.7	2.7	0.1	1.1

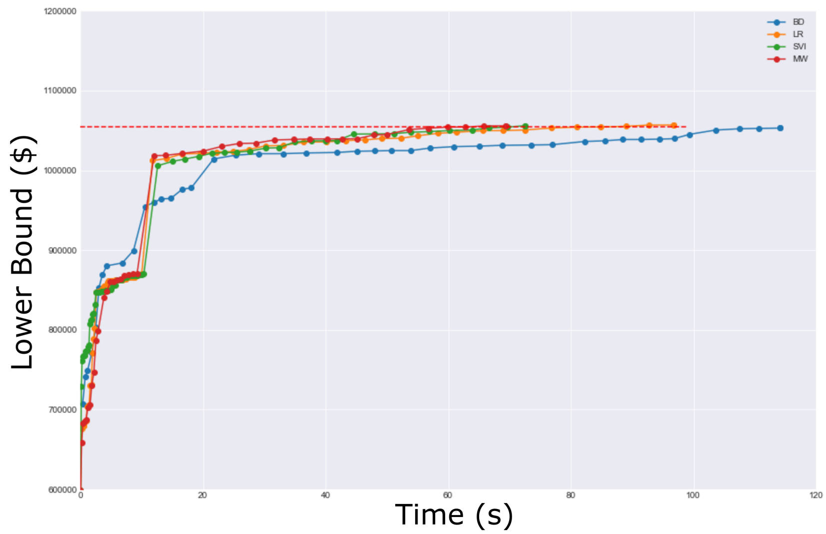
Table 19: Bender Optimality Gap and Overall Gap for problems *P7*, *P8*, *P9* with $\rho = 0.80$

We saw in the previous section that the bigger ρ, the less impact the post-interdiction as on the objective value. As a result, it is also true that a higher value of ρ help to close the overall gap even though the bender decomposition has not yet reach optimality.

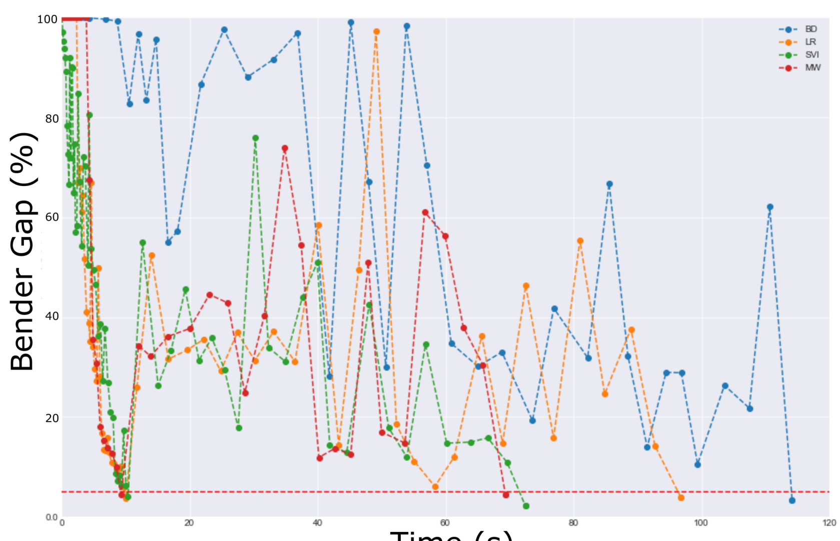
5.2.3 Comparison of the Convergence

8 shows the convergence of the four algorithm for the problem *1277* with $\rho = 0.8$ and $B = 2$, while 9 shows the same figure under a time limit of 20s . As expected, the standard Bender decomposition perform badly in terms of closing the optimality gap. Relaxing the integrability constraint in the master problem (LR) help the algorithm to constantly improve the optimality gap. All methods are characterized in terms of objective value overtime by a fast convergence under 20s and a very slow slope until optimality is reach. We also observe that the optimality gap evolution is very chaotic and has clearly no convergence property. This can be explained by the fact that the restrict solution space is define as the algorithm progresses and some poor solution in terms of optimality gap are selected because they are not yet covered in the master problem, therefore nothing penalize that solution to be selected yet.

We clearly see the end of the fist phase (integrality relaxation of the MP) after about 10s for the LR, SVI, and MW enhancement methods as we can see in Figure 9b. At the same time we observe a plateau on Figure 9a between 2.5s and 10s that also correspond at end of the first phase.

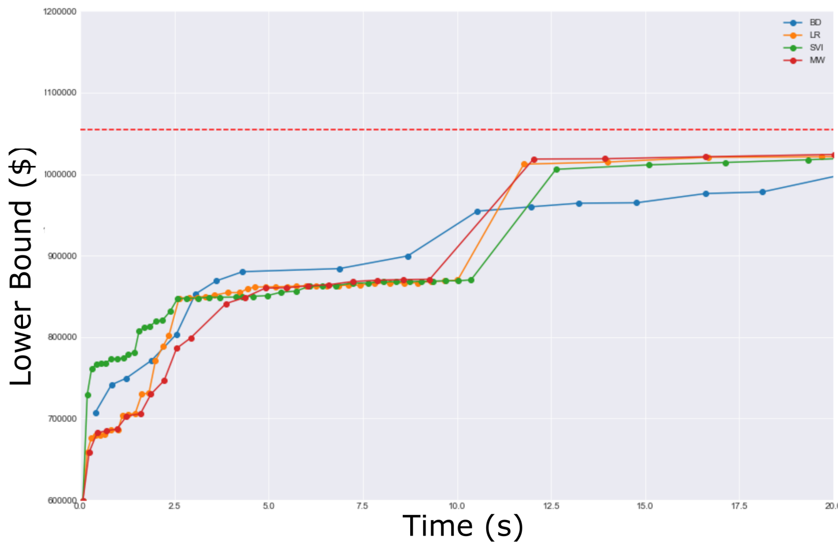


(a) Lower Bound over Time

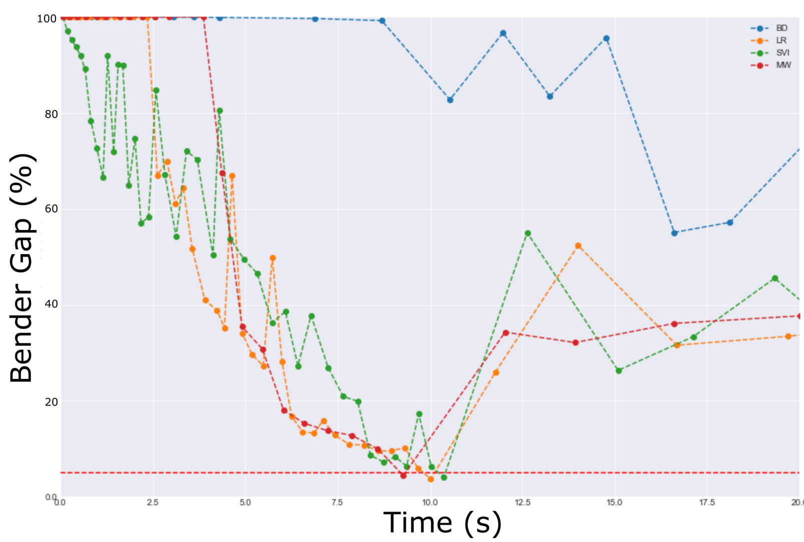


(b) Bender Optimality Gap over Time

Figure 8: Convergence Plot



(a) Lower Bound over Time



(b) Bender Optimality Gap over Time

Figure 9: Convergence Plot under 20s

Chapter 6

Conclusion

In this research, we present a model that takes into account the risk of a potential attack on the classical two-echelon facility location problem. We saw that in the original model an intelligent attack has a severe financial impact on the distribution network. Our model offers decisions makers a tool to design a distribution network that will diminish the cost of distributing products after a disruptive event by investing in a more robust network topology. In particular, the designer is given a choice on how much importance he wishes to assign into reducing the post-interdiction cost.

We implemented a Bender Decomposition approach to address this porblem, and we provided two accelerating approaches namely super-valid inequalities and Pareto-optimal cut. Both solutions offer significant CPU time decrease compared to the standard Bender Decomposition. However, the Pareto-optimal cut generation approach is the fastest, and we recommend this technique to solve the studied problem.

We solved scenarios with 49-node and 88-node customers and potential intermediate facilities. The results show that an intentional attacker can quickly destabilize the network,and customers may not be served by the distribution network post-interdiction resulting in a significant financial loss. The formulation we developed offers flexibility concerning the weight of the post-interdiction cost in the design of a distribution network. However, the robustness of a solution is only slightly affected by the weight of the post-interdiction. The premium paid for designing a robust network is recouped after a couple of days of disruption no matter the weight chosen for the post-interdiction. On the other hand, the enemy budget is an important parameter to consider. Anticipating the number of potential simultaneous targets is critical.

For future work, multiple possible extensions are available given the fact that we based this research on the most simple multi-echelon facility location problem to open the work on this area of research that considers deterministic network interdiction on hierarchical network design. We believe that the model and methodology studied in this research is general enough to be extended to other classical models. A more straightforward extension would be to look at the location and interdiction of suppliers that have not been tackled in this study. It would also be very beneficial to integrate other types of decision which could be strongly impacted by disruptive events such as inventory management and routing decision.

Appendix A

Bender Master Problem

```
from operator import itemgetter, attrgetter
# first import the Model class from docplex.mp
from docplex.mp.model import Model

def DOcplex_tcnip_master_problem(pb,rho):
    """ Initial Two echelon capacitated network interdiction problem

    arguments:
    pb -- Dictionary representing the problem (see problem documentation)
    """

    ###################
    # extract data from the problem
    d,f,c,a1,a2,I,V1,V2 = itemgetter('d','f','c','a1','a2','I','V1','V2')(pb)

    ###################
    # create one model instance
    m = Model(name='Two_echelon_capacitated_facility_location')

    ###################
    # Define variables

    # x(i,j) is the flow going out of node i to node j
    X1 = {(i,j1): m.continuous_var(name='x1_{0}_{1}'.format(i,j1))
         for i in I for j1 in V1}
    X2 = {(j1,j2): m.continuous_var(name='x2_{0}_{1}'.format(j1,j2))
         for j1 in V1 for j2 in V2}
    Y = {(j): m.binary_var(name='Y_{0}'.format(j)) for j in V1}
    Z = m.continuous_var(name='Z')

    ###################
    # Define constraints

    # constraint #1: demande constraint
    for i in I:
        m.add_constraint(m.sum(X1[i,j1] for j1 in V1) >= d[i], ctname='demande_%s' % i)

    # constraint #2: Flow conservation constraint
    for j1 in V1:
        m.add_constraint(m.sum(X2[j1,j2] for j2 in V2) >= m.sum(X1[i,j1] for i in I), ctname='
            flow_%s' % j1)

    # constraint #3: Capacity constraint on facility V2
    for j2 in V2:
        m.add_constraint(m.sum(X2[j1,j2] for j1 in V1) <= a2[j2], ctname='capa_lvl2_%s' % j2)
```

```
    # constraint #4: Capacity constraint on facility V1
    for j1 in V1:
        m.add_constraint(m.sum(X2[j1,j2] for j2 in V2) <= a1[j1] * Y[j1], ctname='capa_lvl1_%s' %
            j1)

    ###################
    # Define Objective
    m.minimize(m.sum(rho * X1[i,j1] * c[i][j1] for i in I for j1 in V1) \
               + m.sum(rho * X2[j1,j2] * c[j1][j2] for j1 in V1 for j2 in V2) \
               + (1-rho) * Z \
               + m.sum(Y[j1] * f[j1] for j1 in V1))

    return m
```

Appendix B

Bender Sub Problem

```
from operator import itemgetter, attrgetter
# first import the Model class from docplex.mp
from docplex.mp.model import Model

def DOcplex_two_echelon_capacitated_interdiction_problem(pb,B, dummy=False,m_dummy=10):
    """ Solve the facility location problem

    arguments:
    pb -- Dictionary representing the problem (see problem documentation)
    B  -- Number of facility to interdict

    Note:
    V1 -- Set of open ficility. There is no design decision in this problem
    """

    ###################
    # extract data from the problem
    d,f,c,a1,a2,I,V1,V2,b = itemgetter('d','f','c','a1','a2','I','V1','V2','b')(pb)

    Targets = V1
    # dummy facility
    if dummy:
        V1 = list(V1) + [len(V1)]
        a1 = list(a1) + [sum(d)]

        c_temp = []
        for i in I:
            c_temp.append(c[i] + [max(c[i])*m_dummy])

        c_temp.append([max(c_temp[i]) for i in I] + [0])

        c        = c_temp
        Targets  = Targets[:-1]

    ###################
    # create one model instance
    m = Model(name='Two_echelon_capacitated_interdiction_problem')

    ###################
    # Define variables

    beta     = {i:  m.continuous_var(name='beta_{0}'.format(i)) for i in I}
    gamma    = {j1: m.continuous_var(name='gamma_{0}'.format(j1)) for j1 in V1}
    delta    = {j2: m.continuous_var(name='delta_{0}'.format(j2)) for j2 in V2}

    Tepsilon = {j1: m.continuous_var(name='Tepsilon_{0}'.format(j1)) for j1 in V1}
```

```
    T        = {j1: m.binary_var(name='T_{0}'.format(j1)) for j1 in V1}
    epsilon  = {j1: m.continuous_var(name='epsilon_{0}'.format(j1)) for j1 in V1}

    ##################
    # Define constraints

    # constraint #1: Shadow cost constraint
    for i in I:
        for j1 in V1:
            m.add_constraint(beta[i] - gamma[j1] <= c[i][j1], ctname='shadow_cost1_%s_%s' % (i,j1)
                )

    # constraint #2:
    for j1 in V1:
        for j2 in V2:
            m.add_constraint(gamma[j1] - delta[j2] - epsilon[j1] <= c[j1][j2], ctname='
                shadow_cost2_%s_%s' % (j1,j2))

    # constraint #3: Interdiction budget
    m.add_constraint(m.sum(b[j1] * (1 - T[j1]) for j1 in Targets) <= B, ctname='
        interdiction_budget')

    epsilon_bar = {j1: 2*max(c[j1]) for j1 in V1}

    # constraint #4: linearisation 1
    for j1 in V1:
        m.add_constraint(Tepsilon[j1] - epsilon_bar[j1]*T[j1] <= 0, ctname='lin1_%s' % j1)

    # constraint #5: linearisation 2
    for j1 in V1:
        m.add_constraint(Tepsilon[j1] - epsilon[j1] <= 0, ctname='lin2_%s' % j1)

    # constraint #6: linearisation 3
    for j1 in V1:
        m.add_constraint(Tepsilon[j1] - epsilon[j1] - epsilon_bar[j1]*T[j1] >= - epsilon_bar[j1],
            ctname='lin3_%s' % j1)

    ##################
    # Define Objective
    m.maximize(m.sum(d[i]*beta[i] for i in I) \
            - m.sum(a2[j2]*delta[j2] for j2 in V2) \
            - m.sum(a1[j1]*Tepsilon[j1] for j1 in V1))

    return m
```

Appendix C

Bender Decomposition Algorithm

```
import pkg
import cplex

from timeit import default_timer as timer

class DOcplex_tcnip:
    """Generator for looping over the bender cut algorithm"""

    def __init__(self,pb,B,rho,
                 epsilon=0.05, max_time=7200, relaxing_m=False,
                                 relaxing_s=False,
                 debug=False, verbose=False, UB=True, **kw):
        """
        pb:
        d[i]      demand of customer i                     - 1D array
        f[j]      cost of opening facility j               - 1D arrays
        c[a][b]   transportation cost between node a and b - 2D arrays
        a[j]      capacity of facility j                   - 1D arrays
        b[j]      cost of destroying facility j            - 1D arrays

        I,V1,V2  set of index for each group offacility

        epsilon:      acceptable optimaily gap
        max_time:     terminate the algorithm after max_time (s)
        """
        self.__dict__.update(kw)
        self.__dict__.update(locals())

        self.initialisation()

    def initialisation(self):
        """ Create the problem """
        self.stop = False; self.s_status = 0 ; self.m_status = 0;
        self.loop = 0; self.time = 0; self.obj = 1;
        self.best_objective = 0; self.best_solution = []
        self.best_stop_gab = 0.5;  self.best_objective = float('Inf');
        self.best_phi = float('Inf')

        # Generate the opl problem with Docplex
        m_prob = pkg.DOcplex_tcnip_master_problem(self.pb,self.rho)
        s_prob = pkg.DOcplex_two_echelon_capacitated_interdiction_problem(self.pb,self.B)

        # Write the opl
        m_prob.export_as_lp('temp_DOcplex_tcnip_m_prob.lp')
        s_prob.export_as_lp('temp_DOcplex_tcnip_s_prob.lp')
```

```
    # Create Cplex object
    self.m_prob = cplex.Cplex('temp_DOcplex_tcnip_m_prob.lp')
    self.s_prob = cplex.Cplex('temp_DOcplex_tcnip_s_prob.lp')

    # Turn on/off Cplex log (activated the log will reduce the speed of the algorithm)
    if not self.verbose:
        self.m_prob.set_log_stream(None)     ; self.s_prob.set_log_stream(None)
        self.m_prob.set_error_stream(None)   ; self.s_prob.set_error_stream(None)
        self.m_prob.set_warning_stream(None); self.s_prob.set_warning_stream(None)
        self.m_prob.set_results_stream(None); self.s_prob.set_results_stream(None)

    # need to use traditional branch-and-cut to allow for control callbacks
    self.m_prob.parameters.mip.strategy.search.set(
        self.m_prob.parameters.mip.strategy.search.values.traditional)

    # The problem will be solved several times, so turn off advanced start
    self.m_prob.parameters.advance.set(0)
    self.s_prob.parameters.advance.set(0)

    # We want to prevent the branch and cut to consume to much time
    self.s_prob.parameters.timelimit.set(self.max_time / 10)
    self.m_prob.parameters.timelimit.set(self.max_time / 10)

    if self.relaxing_m:
        ###Relaxing the integrality constraint
        # Keeping the information on integrality of some variables might allow the
                    # MIP presolve to fix some variable or tigthen the bounds of some
                        variables
        # Also if cut generation is kept active, new cuts will be added to the root node.
        # All that will allow you to get a tighter (thus better) relaxation of your MIP.
        self.m_prob.parameters.mip.tolerances.integrality.set(0.5)
        self.phase2 = False

    if self.relaxing_s:
        self.s_prob.parameters.mip.tolerances.integrality.set(0.5)
        self.phase2 = False

    if not self.relaxing_m and not self.relaxing_s:
        self.phase2 = True

    #### Algorithm Steps ####

def add_bender_cut(self):
    """ Add cut to the master problem """

    self.bdc_r   = [self.pb['a1'][j1] * \
                                        self.s_prob.solution.get_values('Tepsilon_%s' %j1)
                                              for j1 in self.pb['V1']]
    self.bdc_rhs = sum([self.pb['d'][i] * \
                                        self.s_prob.solution.get_values('beta_{0}'.format(
                                            i))
                          for i in self.pb['I']]) \
                  - sum([self.pb['a2'][j2]*self.s_prob.solution.get_values('delta_{0}'.format(
                      j2))
                           for j2 in self.pb['V2']])

    cut = {
        "lin_expr": [cplex.SparsePair(ind = ["Z"] + ["Y_%s" % j1 for j1 in self.pb['V1']],
                                      val = [1] + self.bdc_r)],
        "senses"  : ["G"],
        "rhs"     : [self.bdc_rhs]
    }

    self.m_prob.linear_constraints.add(lin_expr=cut["lin_expr"],
                                       senses=cut["senses"],
                                       rhs=cut["rhs"])

def step1(self):
    """ Solve master problem """
```

```
        # processing
        start = timer()
        self.m_prob.solve()
        end = timer()

        # post-processing
        self.F1         = self.m_prob.solution.get_values(
                            ['Y_%s' % j1 for j1 in self.pb['V1']])

        # Update the Sub-problem objective function
        self.s_prob.objective.set_linear(
            [("Tepsilon_%s" % j1,-self.pb['a1'][j1]*item)for j1,item in enumerate(self.F1)])

        return end - start

    def step2(self):
        """ Solve sub problem """

        # processing
        start = timer()
        self.s_prob.solve()
        end = timer()

        return end - start

    def step3(self):
        """ Summary/ Reporting """

        self.loop += 1
        self.s_status   = self.s_prob.solution.get_status()
        self.m_status   = self.m_prob.solution.get_status()

        # Stats
        self.obj        = self.m_prob.solution.get_objective_value()
        self.Z          = self.m_prob.solution.get_values('Z')
        self.phi        = self.s_prob.solution.get_objective_value()
        self.fixed_cost = sum(self.pb['f'][j1] * item for j1,item in enumerate(self.F1))

        self.psi        = (self.obj - self.fixed_cost - (1-self.rho) * self.phi) / self.rho
        self.solution   = [j1 for j1,item in enumerate(self.F1) if item > 0.5]

        # Stoping craterias
        self.stop_gab = abs(1 - self.Z / self.phi)
        self.stop = self.stop or self.stop_gab < self.epsilon or self.time > self.max_time

        # Terminating LP relaxation
        if self.stop and not self.phase2:
            self.stop   = False
            self.phase2 = True
            self.m_prob.parameters.mip.tolerances.integrality.set(0.00001)
            self.s_prob.parameters.mip.tolerances.integrality.set(0.00001)
            self.report["LR_iter."] = self.loop
            if self.debug:
                print("---_MIP_---")

        #Summary
        report = {
            "iteration"         : self.loop,
            "objective"         : self.obj,
            "solution"          : self.solution,
            'cpu_time'          : self.time,
            "optimality_gap"    : self.stop_gab,
            }

        if self.debug: # details
            report.update({
                "psi"           : self.psi,
                "phi"           : self.phi,
                "Z_phi"         : self.Z,
```

```
            "fixed_cost"    : self.fixed_cost,
            "time_s"        : self.time_s,
            "time_m"        : self.time_m,
        })

        report.update({
            "overall_UB"    : self.best_objective,
            "overall_LB"    : self.obj,
            "overall_gap"   : abs((self.best_objective / self.obj) - 1),
            "Phase"         : 2 if self.phase2 else 1
        })

    return report

#### Collecting Info ####

def get_Tepsilon(self):
    return [self.s_prob.solution.get_values('Tepsilon_%s' %j1)
            for j1 in self.pb['V1']]
def get_beta(self):
    return [self.s_prob.solution.get_values('beta_%i' % i)
            for i in self.pb['I']]
def get_delta(self):
    return [self.s_prob.solution.get_values('delta_%j2' % j2)
            for j2 in self.pb['V2']]

def save_best_solution(self):
    """Calculate an Upper bound for Master of we obtain a better Upper bound
    for the Subproblem"""
    # If the Algorithm doesn't terminate we want a good approximate solution
    # On top of that the DOcplex_tcnip_obj is very fast to solve

    if self.phi < self.best_phi: # min SP => actual < best
        psi, phi, fixed_cost = pkg.DOcplex_tcnip_obj(self.pb,self.solution,self.B,log_output=
            False)

        if psi is not None:
            objective = self.rho * psi + (1-self.rho)*phi + fixed_cost
            if objective < self.best_objective: #better UB => actual < best
                self.best_gab       = abs(1 - (objective / self.obj))
                self.best_solution  = self.solution
                self.best_objective = objective
                self.best_phi       = self.phi

#### Iteration Logic ####

def terminate(self):
    if self.time > self.max_time: #time out
        self.report["objective"]     = self.best_objective
        self.report["solution"]      = self.best_solution

    # Overall LB-UB and gap
    # Note: require save_best_solution to be called to opdate bounds
    self.report["overall_gap"]     = abs((self.best_objective / self.obj) - 1)
    self.report["overall_UB"]      = self.best_objective
    self.report["overall_LB"]      = self.obj
    # Bender LB-UB and gap
    self.report["BD_gap"]          = self.stop_gab
    self.report["BD_UB"]           = self.phi
    self.report["BD_LB"]           = self.Z
    # Problem parameters
    self.report["B"]               = self.B
    self.report["rho"]             = self.rho

    return self.report

def run(self):
    """ Generetor for iterating over the problem"""
    #self.initialisation()
```

```
        while(not self.stop):
            self.step()
            yield self.report

        self.terminate()

        yield self.report

    def step(self):
        # Bender
        self.temp_obj = self.obj
        self.time_m = self.step1()
        self.time_s = self.step2()
        # Reporting
        self.time += self.time_m + self.time_s
        self.report = self.step3()

        self.add_bender_cut()

        return self.report
```

Appendix D

Bender Decomposition and Super-valid Inequalities Algorithm

```
import pkg
import cplex

class DOcplex_tcnip_2(pkg.DOcplex_tcnip):

    def initialisation(self):
        """Add initialisation relative to SVI"""
        self.best_LB_Z = 0; self.nbr_svi = 0; self.UB_phi = 0;
        super().initialisation()

    def add_SVI(self):
        """The Post-interdiction must be at least equal to the pre-interdiction cost"""
        LB_Z = self.obj - self.fixed_cost

        ### SVI 1
        # Add the cut only if it further constrain the algorithm
        if LB_Z > self.best_LB_Z:
            # Add Z svi
            cut = {
                "lin_expr": [cplex.SparsePair(ind = ["Z"],
                                              val = [1])],
                "senses"  : ["G"],
                "rhs"     : [self.obj - self.fixed_cost]
            }

            self.nbr_svi += 1
            self.report["svi"] = self.obj - self.fixed_cost
            self.m_prob.linear_constraints.add(lin_expr=cut["lin_expr"],senses=cut["senses"], rhs=
                cut["rhs"])


        ### SVI 2
        # Add the cut only if it further constrain the algorithm

        # lifting procedure
        r = sorted(self.bdc_r, reverse=True)
        for i in range(1,len(self.pb["V1"])):
            if self.bdc_rhs - sum(r[:i]) - self.phi < 0:
                break

        cut = {
            "lin_expr": [cplex.SparsePair(ind = ["Y_%s" % j1 for j1 in self.pb['V1']],
```

```
                                          val = [1 if Tepsilon > 0 else 0 for Tepsilon in self.
                                              get_Tepsilon()])],
            "senses"  : ["G"],
            "rhs"     : [i]
        }

        self.m_prob.linear_constraints.add(lin_expr=cut["lin_expr"],senses=cut["senses"], rhs=cut[
            "rhs"])

    def step(self):
        super().step()

        # SVI
        self.add_SVI()

        if self.debug:
            self.report["nbr_svi"]  = self.nbr_svi

        return self.report
```

Appendix E

Bender Decomposition and Pareto Optimal Cut Algorithm

```
%%file "./pkg/DOcplex_tcnip_3.py"

import pkg
import cplex

from timeit import default_timer as timer

class DOcplex_tcnip_3(pkg.DOcplex_tcnip):

    def initialisation(self):
        super().initialisation()
        self.mw = True; self.mw_cut = 0

    def step_mw(self):
        """ Solve independant Magnanti-Wong Problem """
        # Update core point
        self.Y0 = [0.5 * self.Y0[j1] + 0.5 * self.F1[j1] for j1 in self.pb["V1"]]

        # Update Objective function
        self.s_prob.objective.set_linear(
            [("Tepsilon_%s" % j1,-self.pb['a1'][j1]*self.Y0[j1]) for  j1 in self.pb["V1"]])

        # processing
        start = timer()
        self.s_prob.solve()
        self.s_status = self.m_prob.solution.get_status()

        # cuts
        cut = {
            "lin_expr": [cplex.SparsePair(ind = ["Z"] + ["Y_%s" % j1 for j1 in self.pb['V1']],
                                          val = [1] + [self.pb['a1'][j1]*self.s_prob.solution.
                                                get_values('Tepsilon_%s' %j1)
                                                        for j1 in self.pb['V1']])],
            "senses"  : ["G"],
            "rhs"     : [sum([self.pb['d'][i]*self.s_prob.solution.get_values('beta_{0}'.format(i)
                 )
                             for i in self.pb['I']]) \
                        - sum([self.pb['a2'][j2]*self.s_prob.solution.get_values('delta_{0}'.
                             format(j2))
                               for j2 in self.pb['V2']])]
        }
```

```
        self.m_prob.linear_constraints.add(lin_expr=cut["lin_expr"],senses=cut["senses"], rhs=cut[
            "rhs"])

        end = timer()

        return end - start

    def step(self):
        # Magnati-Wong
        if self.loop is not 0 and self.mw:
            self.time_mw = self.step_mw()
            self.mw_cut += 1
        else:
            self.time_mw  = 0
            self.obj = 1

        self.time += self.time_mw

        super().step()

        #debug
        if self.debug:
            self.report["time_mw"] = self.time_mw
            self.report["mw_cut"]  = self.mw_cut

        #Improvement
        if self.loop is 1:
            self.Y0 = list(self.F1)

        #Amelioration of solution if slow improvement
        #         if abs(1 - self.obj / self.temp_obj) < 0.01: # no evolution in the obj
        #             tolerances = self.m_prob.parameters.mip.tolerances.integrality.get()
        #             if tolerances > 0.00001:
        #                 self.m_prob.parameters.mip.tolerances.integrality.set(
        #                     tolerances / 2
        #                 )

        return self.report
```

Bibliography

Aardal, Karen (1992). "On the solution of one and two-level capacitated facility location problems by the cutting plane approach". In: *Unpublished Ph. D. Thesis, Université Catholique de Louvain, Faculté des Sciences Appliquées.*

— (1998). "Reformulation of capacitated facility location problems: How redundant information can help". In: *Annals of Operations Research* 82, pp. 289–308.

An, Yu et al. (2014). "Reliable p-median facility location problem: two-stage robust models and algorithms". In: *Transportation Research Part B: Methodological* 64, pp. 54–72.

Anandalingam, G and Victor Apprey (1991). "Multi-level programming and conflict resolution". In: *European Journal of Operational Research* 51.2, pp. 233–247.

Assimakopoulos, Nikitas (1987). "A network interdiction model for hospital infection control". In: *Computers in biology and medicine* 17.6, pp. 413–422.

Azad, Nader et al. (2013). "Strategies for protecting supply chain networks against facility and transportation disruptions: an improved Benders decomposition approach". In: *Annals of Operations Research* 210.1, pp. 125–163.

Barkley, Timothy R (2008). *An attacker-defender model for IP-based networks.* Tech. rep. Naval postgraduate school monterey ca.

Bayrak, Halil and Matthew D Bailey (2008). "Shortest path network interdiction with asymmetric information". In: *Networks* 52.3, pp. 133–140.

Benders, Jacques F (1962). "Partitioning procedures for solving mixed-variables programming problems". In: *Numerische mathematik* 4.1, pp. 238–252.

Brown, Gerald et al. (2005). "A two-sided optimization for theater ballistic missile defense". In: *Operations research* 53.5, pp. 745–763.

Brown, Gerald et al. (2006). "Defending critical infrastructure". In: *Interfaces* 36.6, pp. 530–544.

Brown, Gerald G et al. (2009). “Interdicting a nuclear-weapons project”. In: *Operations Research* 57.4, pp. 866–877.

BSI, British Standards Institution (2017). *Impact of supply chain attacks hits highest rate ever*. Accessed: 2018-03-26.

Bundschuh, Markus, Diego Klabjan, and Deborah L Thurston (2003). “Modeling robust and reliable supply chains”. In: *Optimization Online e-print*.

Church, Richard L, Maria P Scaparra, and Richard S Middleton (2004). “Identifying critical infrastructure: the median and covering facility interdiction problems”. In: *Annals of the Association of American Geographers* 94.3, pp. 491–502.

Cormican, Kelly J, David P Morton, and R Kevin Wood (1998). “Stochastic network interdiction”. In: *Operations Research* 46.2, pp. 184–197.

Cornuéjols, Gérard, George L Nemhauser, and Lairemce A Wolsey (1983). *The uncapacitated facility location problem*. Tech. rep. Carnegie-mellon univ pittsburgh pa management sciences research group.

Cote, Gilles and Michael A Laughton (1984). “Large-scale mixed integer programming: Benders-type heuristics”. In: *European Journal of Operational Research* 16.3, pp. 327–333.

Daskin, Mark S (2008). “What you should know about location modeling”. In: *Naval Research Logistics (NRL)* 55.4, pp. 283–294.

Drezner, Zvi (1987). “Heuristic solution methods for two location problems with unreliable facilities”. In: *Journal of the Operational Research Society* 38.6, pp. 509–514.

Fulkerson, Delbert Ray and Gary C Harding (1977). “Maximizing the minimum source-sink path subject to a budget constraint”. In: *Mathematical Programming* 13.1, pp. 116–118.

Geoffrion, Arthur M and Glenn W Graves (1974). “Multicommodity distribution system design by Benders decomposition”. In: *Management science* 20.5, pp. 822–844.

Ghare, PM, Douglas C Montgomery, and WC Turner (1971). “Optimal interdiction policy for a flow network”. In: *Naval Research Logistics (NRL)* 18.1, pp. 37–45.

Granata, Donatella, Gregory Steeger, and Steffen Rebennack (2013). “Network interdiction via a critical disruption path: Branch-and-price algorithms”. In: *Computers & Operations Research* 40.11, pp. 2689–2702.

Hakimi, S Louis (1964). "Optimum locations of switching centers and the absolute centers and medians of a graph". In: *Operations research* 12.3, pp. 450–459.

Israeli, Eitan and R Kevin Wood (2002). "Shortest-path network interdiction". In: *Networks* 40.2, pp. 97–111.

Li, Qingwei, Bo Zeng, and Alex Savachkin (2013). "Reliable facility location design under disruptions". In: *Computers & Operations Research* 40.4, pp. 901–909.

Lim, Churlzu and J Cole Smith (2007). "Algorithms for discrete and continuous multicommodity flow network interdiction problems". In: *IIE Transactions* 39.1, pp. 15–26.

Magnanti, Thomas L and Richard T Wong (1981). "Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria". In: *Operations research* 29.3, pp. 464–484.

McDaniel, Dale and Mike Devine (1977). "A modified Benders' partitioning algorithm for mixed integer programming". In: *Management Science* 24.3, pp. 312–319.

McMasters, Alan W and Thomas M Mustin (1970). "Optimal interdiction of a supply network". In: *Naval Research Logistics (NRL)* 17.3, pp. 261–268.

Melo, M Teresa, Stefan Nickel, and Francisco Saldanha-Da-Gama (2009). "Facility location and supply chain management–A review". In: *European journal of operational research* 196.2, pp. 401–412.

Morton, David P, Feng Pan, and Kevin J Saeger (2007). "Models for nuclear smuggling interdiction". In: *IIE Transactions* 39.1, pp. 3–14.

Ortiz-Astorquiza, Camilo, Ivan Contrerasn, and Gilbert Laporte (2017). "Multi-level facility location problems". In: *European Journal of Operational Research*.

Pan, Feng and David P Morton (2008). "Minimizing a stochastic maximum-reliability path". In: *Networks* 52.3, pp. 111–119.

Papadakos, Nikolaos (2008). "Practical enhancements to the Magnanti–Wong method". In: *Operations Research Letters* 36.4, pp. 444–449.

Peng, Peng et al. (2011). "Reliable logistics networks design with facility disruptions". In: *Transportation Research Part B: Methodological* 45.8, pp. 1190–1211.

Sadeghi, Somayeh, Abbas Seifi, and Elnaz Azizi (2017). "Trilevel shortest path network interdiction with partial fortification". In: *Computers & Industrial Engineering* 106, pp. 400–411.

Saharidis, Georges K and Marianthi G Ierapetritou (2009). “Resolution method for mixed integer bi-level linear problems based on decomposition technique”. In: *Journal of Global Optimization* 44.1, pp. 29–51.

Saharidis, Georgios KD and Marianthi G Ierapetritou (2010). “Improving Benders decomposition using maximum feasible subsystem (MFS) cut generation strategy”. In: *Computers & chemical engineering* 34.8, pp. 1237–1245.

Şahin, Güvenç and Haldun Süral (2007). “A review of hierarchical facility location models”. In: *Computers & Operations Research* 34.8, pp. 2310–2331.

Sheffi, Yossi and James B Rice Jr (2005). “A supply chain view of the resilient enterprise”. In: *MIT Sloan management review* 47.1, p. 41.

Shishebori, Davood, Lawrence V Snyder, and Mohammad Saeed Jabalameli (2014). “A reliable budget-constrained FL/ND problem with unreliable facilities”. In: *Networks and Spatial Economics* 14.3-4, pp. 549–580.

Smith, J Cole, Churlzu Lim, and Fransisca Sudargho (2007). “Survivable network design under optimal and heuristic interdiction scenarios”. In: *Journal of global optimization* 38.2, pp. 181–199.

Snyder, Lawrence V and Mark S Daskin (2004). *Data Sets for Reliability Models for Facility Location: The Expected Failure Cost Case*. URL: `http://coral.ise.lehigh.edu/larry/research/data-sets-for-reliability-models-for-facility-location-the-expected-failure-cost-case` (visited on 02/19/2018).

— (2005). “Reliability models for facility location: the expected failure cost case”. In: *Transportation Science* 39.3, pp. 400–416.

Snyder, Lawrence V et al. (2006). “Planning for disruptions in supply chain networks”. In: *Tutorials in operations research* 2, pp. 234–257.

Snyder, Lawrence V et al. (2016). “OR/MS models for supply chain disruptions: A review”. In: *IIE Transactions* 48.2, pp. 89–109.

Washburn, Alan and Kevin Wood (1995). “Two-person zero-sum games for network interdiction”. In: *Operations research* 43.2, pp. 243–251.

Wollmer, Richard (1964). “Removing arcs from a network”. In: *Operations Research* 12.6, pp. 934–940.

Wood, R Kevin (1993). “Deterministic network interdiction”. In: *Mathematical and Computer Modelling* 17.2, pp. 1–18.

Wood, R Kevin (2011). “Bilevel network interdiction models: Formulations and solutions”. In: *Wiley encyclopedia of operations research and management science*.

Zakeri, Golbon, Andrew B Philpott, and David M Ryan (2000). “Inexact cuts in Benders decomposition”. In: *SIAM Journal on Optimization* 10.3, pp. 643–657.