CENTRALE LILLE

THÈSE

présentée en vue d'obtenir le grade de

DOCTEUR

en

Spécialité : Informatique

par

SALAZAR ZENDEJA Luis Alberto

DOCTORAT DELIVRE PAR CENTRALE LILLE

Titre de la thèse :

Modèles et Algorithmes pour le Problème d'Interdiction de l'Arbre Couvrant de Poids Minimal

Models and Algorithms for the Minimum Spanning Tree Interdiction Problem

---

Soutenue le 28 novembre 2022 devant le jury d'examen :

Présidente	Safia KEDAD-SIDHOUM	Professeure, CNAM
Rapporteur	François CLAUTIAUX	Professeur, Université de Bordeaux
Rapporteure	Yasmín Á. RÍOS SOLÍS	Professeure, Tecnológico de Monterrey
Examinatrice	Carmen GALÉ POLA	Professeure, Universidad de Zaragoza
Directeur de thèse	Frédéric SEMET	Professeur, Centrale Lille
Co-directrice de thèse	Martine LABBÉ	Professeure, Université Libre de Bruxelles
Co-Encadrant	Diego CATTARUZZA	MCF, Centrale Lille

Thèse préparée dans le laboratoire CRIStAL
École Doctorale MADIS-631 - Center Inria University of Lille - Centrale Lille

À mes parents,
à toute ma famille,
à mes professeurs,
et à mes chèr(e)s ami(e)s.

Acknowledgements

Many people were essential to the culmination of this four years adventure. I would like to thank my esteemed supervisors Frédéric Semet, Diego Cattaruzza and Martine Labbé for their invaluable supervision, support and tutelage during my PhD degree. Thank you Diego for the countless talks and your continuous advice, you have made me improve every time. Thank you Fédéric for pointing out the small details of the research and your great expertise, which has helped our work to progress. Thank you Martine for your insight and expertise, your theoretical observations were certainly invaluable. Thank you again my supervisors, all of you taught to become a good researcher and person. Working with you was one of the best experiences of my life.

My gratitude extends to the CONACyT and Université de Lille for the funding opportunity to undertake my studies at Centrale Lille.

I'd like to deeply thank my partner Valentin for all his support during the past couple of years. Without him, this would not have been possible. I would like to thank my friends, lab mates, colleagues and research team Tifaout, Matteo, Natalia, Juan, Wenjiao, Wenjuan, Yuan, Mathieu, Arnaud, Sylvain, Raphaël, Kaba, Francesco, Luce, Maxime, Helène, Pablo T., Pablo E., Pablo H., Pamela and Edouard for a cherished time spent together in the lab, and in social settings. Especially, to Tifaout for all our talks and advice and to Matteo for being a good friend and roommate. Lastly, I would like to thank my parents, my family and Valentin's family, whose endless support has given me the opportunity to focus on my work.

Every one of you made my life better. I'll cherish all these memories forever.

Villeneuve d'Ascq, France — *Luis Alberto SALAZAR ZENDEJA*
November, 2022

ACKNOWLEDGEMENTS

Contents

Acknowledgements i

Table of Contents ii

List of Figures vi

List of Tables viii

Abbreviations xi

1 Introduction 1

1.1 Context 1

1.2 Interdiction Games 1

1.3 Thesis Objective and Organization 3

2 Literature Review 7

2.1 Deterministic Network Interdiction Problems 7

2.2 Literature Review on IGs 11

2.2.1 Maximum Flow Interdiction Problem 11

2.2.2 Shortest Path Interdiction Problem 13

2.2.3 Minimum Spanning Tree Interdiction Problem 14

2.2.4 Variants and Extensions 16

2.2.5 Applications 18

3 Formulations for the MSTI Problem 21

3.1 Bazgan *et al.* (2012) Formulation for the MSTI Problem 22

3.2 MSTI Formulation Based on Martin (1991) MST Formulation . . 24

3.3 Reformulation using Optimality Constraints 26

3.4 Dual Formulations . . . 30
3.5 Total Interdiction Formulation . . . 32
3.6 Parameter Analyses and Illustrative Problem . . . 36
3.7 Valid Inequalities . . . 38
3.7.0.1 First Set of Valid Inequalities . . . 38
3.7.0.2 Second Set of Valid Inequalities . . . 40
3.8 Extension of the $\bar{B}$-edge Connected Certificate . . . 41
3.9 Computational Experiments . . . 42
3.9.1 Instance Specifications . . . 42
3.9.2 Tuning Phase for Valid Inequalities . . . 42
3.9.3 Final Experimentation . . . 44
3.10 Conclusions . . . 54

4 A Branch-and-Price Algorithm for the MSTI Problem 57
4.1 Dantzig-Wolfe Decomposition . . . 58
4.2 B&P Algorithm . . . 60
4.2.1 Branching . . . 60
4.2.2 Pricing Problem . . . 61
4.2.2.1 Strengthening of the Pricing Problem . . . 62
4.2.2.2 Heuristic for the Pricing Problem . . . 63
4.2.3 Lower Bound . . . 64
4.2.4 Overall presentation of the algorithm . . . 66
4.3 Computational Experiments . . . 69
4.4 Conclusions . . . 71

5 Benders Decomposition for the MSTI Problem 73
5.1 A Benders Decomposition . . . 74
5.2 Lifted Optimality Cuts . . . 76
5.3 Set Covering Constraints . . . 77
5.4 Initialization of $\hat{\mathcal{T}}$ and the initial solution . . . 78
5.5 Upper and Lower Bounds of [MP($\hat{\mathcal{T}}$)] . . . 78
5.6 Revised Benders Decomposition Algorithm . . . 79
5.6.1 Enhanced Master Problem . . . 79
5.6.2 Separation Procedure for the Lifted Optimality Cuts . . . 80
5.6.3 Implementation of the Revised BD Algorithm . . . 83

5.7 Computational Experiments 86
5.7.1 Instance Generation 86
5.7.2 Computational Results 87
5.8 Conclusions 95

6 Conclusion and Recommendations **97**
6.1 Thesis Summary 97
6.2 Future Research 99

References **102**

Résumé Etendu **112**

List of Figures

3.1 Comparison of formulation solutions 34
3.2 Illustrative example of $G = (4, 6)$ 37

5.1 Example of lifting procedure 82

LIST OF FIGURES

List of Tables

2.1 Main attributes of Interdiction Problems 20

3.1 Instance cases 36
3.2 Preliminary results of the first set of valid inequalities 43
3.3 Names of formulations with valid inequalities. 43
3.4 Preliminary results of the second set of valid inequalities 44
3.5 Computational results considering total interdiction and cardinality constraint. 48
3.6 Computational results considering partial interdiction and cardinality constraint. 49
3.7 Computational results considering total interdiction and knapsack constraint. 50
3.7 Computational results considering total interdiction and knapsack constraint. (continued) 51
3.8 Computational results considering partial interdiction and knapsack constraint. 52
3.8 Computational results considering partial interdiction and knapsack constraint. (continued) 53

4.1 Computational results for Cardinality Budget Constraint 70
4.2 Computational results for Knapsack Budget Constraint 71

5.1 Computational results for Type 1, 2 and 4 instances under total interdiction and cardinality budget constraint. 88
5.2 Computational results for Type 1, 2 and 4 instances under partial interdiction and cardinality budget constraint. 90

LIST OF TABLES

5.3 Computational results for Type 1, 2 and 4 instances under total interdiction and knapsack budget constraint. 91
5.4 Computational results for Type 1, 2 and 4 instances under partial interdiction and knapsack budget constraint. 92
5.5 Computational results for Type 3 instances considering a cardinality budget constraint. 93
5.6 Computational results for Type 3 instances considering a knapsack budget constraint. 94

Abbreviations

IG	Interdiction Game
MFI	Maximum Flow Interdiction
SPI	Shortest Path Interdiction
MST	Minimum Spanning Tree
MSTI	Minimum Spanning Tree Interdiction
B&P	Branch-and-Price
BD	Benders Decomposition
MP	Master Problem

Chapter 1

Introduction

1.1 Context

Prevention is an important aspect of any operation to ensure its proper functioning. Some processes where prevention plays a key role are transportation, deliveries or production. Natural disasters and man-made disruptions are examples of what can cause them to malfunction which, in most cases, leads to undesired results or even to no results at all. For instance, a manufacturing company must know how to react when a machine failure compromises its supply chain. It must anticipate no production, delays, lost of profits, among others. An analysis in operations within an optimization framework is a must.

The modeling of these situations under an Interdiction Game (IG) setting is one of the most practical ways to guarantee the best results. An IG seeks a strategy that causes the most damage to the operations of a process. It is possible to decide whether to invest in additional security measures at critical points or to deploy alternate procedures once the worst-case scenario that operations may encounter is known.

1.2 Interdiction Games

An IG involves two non-cooperative entities interacting with each other. The first entity, called the *interdictor*, seeks to disrupt the optimization processes or settings of the second entity, called the *defender*. In the normal setting of an IG, the interdictor acts first then the defender reacts with full knowledge of the

interdictor's actions. The scope of the interdictor ranges from modifying the defender's assets to their total destruction in order to worsen the objective of the defender.

IGs have received a lot of attention because of their practical setting. Military operations (Wollmer (1964)), critical infrastructure of electric grids (Salmeron *et al.* (2004), Yuan *et al.* (2011)), law enforcement (Tsai *et al.* (2010)) and prevention of natural disasters (Yuan *et al.* (2016)) are some application domains in which the IGs are applied.

One of the most studied problems is the one presented in Morton *et al.* (2007). The authors model a situation in which an individual who carries nuclear material travels through a network between an origin and a destination. Each link of the network is associated with a probability of traverse it without being detected. The smuggler's objective is to traverse the network undetected while the nuclear plant security personnel attempts to intercept him. Detection is done by installing radiation sensors over the links of the network such that their probabilities to be traversed are diminished. This problem is often referred to in the literature as the most reliable path, as the defender tries to find the path that maximizes the chance of escape.

The work of Wood (1993) studies the Maximum Flow Interdiction (MFI) problem. The defender wishes to compute the maximum flow on a capacitated network. The objective of the interdictor is to minimize the maximum flow of the network by eliminating some of its arcs. However, the interdictor has limited resources to do so. This work is motivated by the direct application of the MFI problem to the degradation of drug flows in South America.

The Shortest Path Interdiction (SPI) problem is considered in Israeli & Wood (2002). The shortest path in a network between the source and sink nodes is computed by the defender while the interdictor, constrained by a budget, disrupts the network by altering the weight of the arcs. The proposed formulations can handle partial interdiction, i.e. an increment on the weight of the arc, as well as complete destruction of the arc, that is, total interdiction.

The Minimum Spanning Tree Interdiction (MSTI) problem is defined similarly as the two previous problems. The interdictor seeks to allocate the available resources in order to modify the network topology in which the defender computes the minimum spanning tree so that its weight has the largest increment.

The maximum number of edges the interdictor can interdict is imposed by a budget constraint. The budget constraint can be defined as cardinality constraint, the interdiction weight of each edge is the same, or a knapsack constraint, the interdiction weights are different. The first work related to the MSTI problem is Hsu *et al.* (1991). They consider a special case where the interdictor eliminates only one edge of the network. In Jan *et al.* (1992), the work is extended so that the interdictor removes an arbitrary number of edges. The most recent work on the MSTI problem is Wei *et al.* (2021) where different mathematical formulations are introduced and compared. The presented models take advantage of the MSTI problem structure. The authors also identify a family of valid inequalities which under a processing of the instance parameters provides an even stronger formulation. The articles addressing the MSTI problem provide a variety of models and resolution methods. However they all consider some special cases. For instance, Bazgan *et al.* (2012) introduces an efficient algorithm that solves the MSTI problem when the interdictor eliminates a fixed number of edges. Wei *et al.* (2021) focuses on the same case as well. The partial interdiction variant, i.e., the interdictor only increases the weight of the edges, has received little attention. The research in Frederickson & Solis-Oba (1999) and Baïou & Barahona (2008) propose algorithms that solve this variant but a study on mathematical formulations is missing.

1.3 Thesis Objective and Organization

To the best of our knowledge, no algorithm has been explicitly designed to solve the general MSTI problem. This thesis proposes models and algorithms for a generalization of MSTI problem. The latter is obtained by considering both types of interdiction, total and partial interdiction. In addition, the interdictior can be limited by a budget constraint, modeled as a knapsack constraint. A cardinality constraint, which is a special case, is also considered. The work presented in this thesis is structured as follows.

A literature review of the MFI, SPI and MSTI problems is presented in Chapter 2. We start by introducing the concept of Stackelberg games and bilevel optimization. Afterwards, their relation to IGs is explained. A general formulation for deterministic network interdiction problems is stated. The literature of the three

network problems considered is discussed. Mainly, the discussion focuses in the special cases that were treated and the different algorithms that were designed. Variants and applications of each of the network problems are also reviewed.

Chapter 3 introduces seven mathematical models for the MSTI problem. A first formulation retrieved form the literature is recalled. A second original model based on a linear formulation of the MST problem (Martin (1991)) is presented. The latter formulation is a $\max - \min$ non-linear problem. Therefore, it is reformulated and linearized to obtain a single-level linear problem. In addition, valid inequalities are developed to strengthen the model. Two additional based on the dual problem of the $\max - \min$ non-linear problem are proposed. All formulations handle both types of interdiction. A last model that only considers total interdiction case is presented. Instances are generated as those from the literature and used to compare the efficiencies of our formulations.

The MSTI problem is then solved by a Branch-and-Price algorithm in Chapter 4. The most compact formulation defined in Chapter 3 is used to define the restricted master problem and the pricing problem. The latter is obtained by applying a Dantzig-Wolfe decomposition. A family of constraints that rules out irrelevant interdiction strategies is developed and added to the pricing problem. Then, a column generation algorithm embedded in a Branch-and-Bound scheme is presented. Computational experiments are carried out on the instance set generated in Chapter 3.

The main contribution of this thesis is presented in Chapter 5. A Benders Decomposition algorithm is developed to tackle the MSTI problem. A MSTI formulation is decomposed into a Master Problem (MP) and a sub-problem. A family constraints for the MP, called optimality constraints, are introduced and lifted. A family of valid inequalities is also proposed to tighten the linear relaxation of the MP. The sub-problem, parameterized by the solution of the MP, consists only on the computation of a MST. The Benders Decomposition algorithm is implemented through a commercial solver. The MP is solved to optimality only once and the sub-problem is solved in user-defined methods that are given to the solver. Four types of instances are generated. Instances differentiate from each others in the range of the weight of the edges and the density of the network. The algorithm is assessed for each of the four possible configurations according to the type of interdiction and budget constraints.

The manuscript concludes in Chapter 6 with general conclusions of the thesis. Limitations and future works are also discussed.

Chapter 2

Literature Review

This chapter presents the literature related to Interdiction Games (IG). We begin by discussing Stackelberg Games, which provide a general structure of problems to which IGs belong. The concept of bilevel optimization is introduced and its relation to IGs is explained. A general structure for several network interdiction problems is presented. Results of the latter problems is discussed. The chapter concludes with applications of these problems.

2.1 Deterministic Network Interdiction Problems

The individual who has final authority over a decision process is called the decision-maker. A Stackelberg Game is a problem in which several decision-makers interact in a hierarchical structure (Von Stackelberg (1934)). In the most simple case of a Stackelberg Game, there are two parties, commonly referred to as leader and follower. They play a two-turn game in which the leader makes decisions first and the follower acts later. The leader's strategy is to anticipate the follower's actions whereas the follower is limited to react to the leader's decisions.

Bilevel programming or bilevel optimization are optimization paradigms in which only two decision makers play a Stackelberg Game (Dempe (2002)). The general bilevel optimization problem is defined as follows:

$$\min_{\mathbf{x}\in\mathcal{X},\mathbf{y}} \; F(\mathbf{x},\mathbf{y}) \tag{2.1a}$$

s.t.

$$G_h(\mathbf{x}, \mathbf{y}) \geq 0 \qquad \forall h \in \mathcal{H} \tag{2.1b}$$

$$\mathbf{y} \in \Psi(\mathbf{x}) \tag{2.1c}$$

where $n_{\mathbf{x}}$ denotes the number of variables of the leader and $n_{\mathbf{y}}$ the number of variables of the follower, $\mathbf{x} \in \mathbb{R}^{n_{\mathbf{x}}}$, $\mathbf{y} \in \mathbb{R}^{n_{\mathbf{y}}}$, $F : \mathbb{R}^{n_{\mathbf{x}}} \times \mathbb{R}^{n_{\mathbf{y}}} \mapsto \mathbb{R}$ and $G_h : \mathbb{R}^{n_{\mathbf{x}}} \times \mathbb{R}^{n_{\mathbf{y}}} \mapsto \mathbb{R}$ for all $h \in \mathcal{H}$. $\mathcal{X} \subseteq R^{n_{\mathbf{x}}}$ defines the domain of variables $\mathbf{x}$. $\Psi(\mathbf{x})$ is a set parameterized by $\mathbf{x}$. This set contains the optimal solutions of the next problem:

$$\min_{\mathbf{y} \in \mathcal{Y}} \; f(\mathbf{x}, \mathbf{y}) \tag{2.2a}$$

$$\text{s.t.} \;\; g_j(\mathbf{x}, \mathbf{y}) \geq 0 \qquad \forall j \in \mathcal{J}$$

where $f : \mathbb{R}^{n_{\mathbf{x}}} \times \mathbb{R}^{n_{\mathbf{y}}} \mapsto \mathbb{R}$, and $g_j : \mathbb{R}^{n_{\mathbf{x}}} \times \mathbb{R}^{n_{\mathbf{y}}} \mapsto \mathbb{R}$ for all $j \in \mathcal{J}$ and $\mathcal{Y} \subseteq R^{n_{\mathbf{y}}}$ denote the domain of variables $\mathbf{y}$. The leader controls variables $\mathbf{x}$ and the follower variables $\mathbf{y}$. Thus, the leader's (or upper level) and follower's (or lower level) problems are (2.1) and (2.2), respectively.

Bilevel problems can model a variety of hierarchical problems. Either the upper level or lower level can represent linear, mixed-integer, quadratic problems, etc. This makes bilevel problems complicated to solve and even in its simplest form, i.e. where both upper level and lower level are liner problems, are $\mathcal{NP}$-hard (Kleinert *et al.* (2021)). The reader is referred to Dempe *et al.* (2015),Vicente & Calamai (1994), Lachhwani & Dwivedi (2018) for surveys on bilevel optimization.

An IG is a special case of bilevel programming. The leader and the follower take on the roles of interdictor and defender, respectively. Both entities play a Stackelberg game in which the objective of both players is the same but optimized in opposite directions. During the first turn, the leader selects his interdiction plan, i.e. a set of decisions that will restrict the follower's actions. In the second turn, the follower knows the interdictor's actions and he reacts optimally. Throughout this manuscript, leader, upper level and interdictor are used interchangeably. Similarly, for follower, lower-level and defender.

The defender has no control over the actions of the interdictor. For example, a transportation company cannot control the closure of roads or highways by the government during a crisis. Moreover, the only purpose of the interdictor is to

worsen the defender's objective. This means that given a solution $\mathbf{x}$ of the upper level problem, any optimal solution of the lower level problem has the same effect on the leader's objective. Therefore, a general IG formulation is defined as:

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} \; f(\mathbf{x},\mathbf{y}) \tag{2.3}$$

where $\mathcal{X} := \{\mathbf{x} \in \bar{\mathcal{X}} \subseteq \mathbb{R}^{n_\mathbf{x}} \mid G_h(\mathbf{x},\mathbf{y}) \geq 0,\ \forall h \in \mathcal{H}\}$ and $\mathcal{Y} := \{\mathbf{y} \in \bar{\mathcal{Y}} \subseteq \mathbb{R}^{n_\mathbf{y}} \mid g_j(\mathbf{x},\mathbf{y}) \geq 0,\ \forall j \in \mathcal{J}\}$. Sets $\bar{\mathcal{X}} \subseteq \mathbb{R}^{n_\mathbf{x}}$ and $\bar{\mathcal{Y}} \subseteq \mathbb{R}^{n_\mathbf{y}}$ determine the nature of the variables, i.e. binary, integer, continuous, etc.

In this thesis, we tackle problems where formulation (2.3) is applied on networks. These problems are known as Network Interdiction problems and are the most studied type of IGs. Formally, the classical network interdiction problem is a game in which the defender optimizes a process over a network, i.e. the space $\mathcal{Y}$ defines a shortest path, maximum flow or minimum spanning tree problem and the interdictor, constrained by a budget, deletes or modifies the network topology to optimize the defender's objective in the opposite direction.

We now define the general notation for the following sections. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ represents the set of nodes and $\mathcal{E}$ the set of edges. Each edge $e \in \mathcal{E}$ is associated with:

- a non-negative weight/length/capacity indicated with c_e; this is the value of the edge e when it does not suffer from interdiction;
- an non-negative interdiction cost indicated with b_e, that has to be paid to interdict the corresponding edge;
- an interdiction augmentation (or decrement) indicated with d_e: if the edge e is interdicted, its weight/ length/capacity becomes $c_e + d_e$.

Total interdiction and partial interdiction are the two types of interdiction considered. Total interdiction is defined as the complete destruction of the edge while partial interdiction modifies the weight of the edge. If $d_e = \pm\infty$, the edge value becomes $\pm\infty$ and the interdiction action equals to edge removal. The interdiction on the edge implies that it does not exist anymore in the graph and cannot be used by the defender. If d_e is finite, the edge value is increased (or

decreased) by d_e units and is still present in the graph: it then might be utilized by the defender at a cost $c_e + d_e$.

This thesis focuses on deterministic network interdiction problems with the following structure:

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} \sum_{e\in\mathcal{E}} (c_e + d_e x_e) y_e \tag{2.4}$$

The objective function of (2.4) calculates the cost or weight of the structure defined by $\mathcal{Y}$ on graph $\mathcal{G}$ with augmented weights. Throughout this manuscript, the feasible space of the interdictor is defined as $\mathcal{X} := \{\mathbf{x} \in \{0,1\}^{|\mathcal{E}|} \mid \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$ where B is the budget of the interdictor. An element $\mathbf{x} \in \mathcal{X}$ is referred to as interdiction plan or interdiction strategy. When $b_e = 1$ for all $e \in \mathcal{E}$, the constraint $\sum_{e\in\mathcal{E}} b_e x_e \leq B$ is referred as cardinality constraint, since it basically imposes an upper bound on the number of edges that can be interdicted. On the other hand, we refer to it as knapsack constraint when there exist at least one edge $\bar{e} \in \mathcal{E}$, such that $b_{\bar{e}}$ is different from 1 for its similarity with the capacity constraint in the knapsack problem.

The minimum cut value of $\mathcal{G}$ calculated with respect to b_e values is referred to as connectivity number. Large values of B, i.e. values greater or equal to the connectivity number, allow the interdictor to disconnect the network when assuming total interdiction. As a consequence, any of the aforementioned network problems have a solution and the instance becomes trivial. In the rest of the paper, we will assume that the budget is not large enough for this to happen when considering total interdiction.

The above notation is edge-based. To treat arc-based problems the notation is modified as follows. Let $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ define a digraph where $\mathcal{A}$ is a set of (i, j) arcs. For each arc $(i, j) \in \mathcal{A}$, parameters c_{ij}, d_{ij} and b_{ij} are defined analogously. Arcs of the form (i, i) do not exist in $\mathcal{D}$. Denote by $\delta^+(i)$ and $\delta^-(i)$ the sets of outgoing and incoming arcs of node i, respectively.

2.2 Literature Review on IGs

Three of the problems introduced in Chapter 1 fall under the modeling scheme (2.4), i.e. the Maximum Flow Interdiction (MFI), Shortest Path Interdiction (SPI) and the Minimum Spanning Tree Interdiction (MSTI) problems. The main results and variants of these problems are presented below. Whenever the problem contains a special node as a source (resp. sink), it will be indicated by s (resp. t).

We start by defining the special cases of network interdiction problems that are commonly found in the literature. The most basic case of interdiction problem (2.4), assuming total interdiction, is the most vital edge (or link) and it is obtained by setting $b_e = 1$ for all $e \in \mathcal{E}$ and by defining $B = 1$. Thus the interdictor looks to disrupt the edge that causes the largest increment (or decrement) in the objective function of the defender. A generalization of the most vital edge problem is the $\bar{B}$-most vital edges (links). It is defined by considering a cardinality constraint ($b_e = 1$ for all $e \in \mathcal{E}$) and a budget strictly greater than one, $\bar{B} = B > 1$. The goal is to remove $\bar{B}$ edges (links) of the network that worsen the most the objective value of the defender.

Partial interdiction is defined as a finite increase (or decrease) of c_e. Thus, the defender problem is redefined in a graph with modified weights. The latter can be modeled in two different ways: the discrete and continuous version. The discrete version consists in deciding whether to increase the edge weight exactly by d_e units or not. The weight of the edges in the continuous version of partial interdiction are defined as $c_e + x_e$ where $x_e \geq 0$. The latter version of partial interdiction is out of the scope of formulation (2.4), however this specific way to model is the one considered in the literature when dealing with partial continuous interdiction. In the following sections, we present how these variants are treated in the literature for the MFI, SPI and MSTI problems.

2.2.1 Maximum Flow Interdiction Problem

The MFI problem is a network interdiction problem in which the interdictor selects a set of edges to interdict in order to minimize the maximum network flow that is computed by the defender. Let $\mathcal{Y}$ define a maximum flow problem. The MFI problem cannot be directly model by (2.4) since it is a $\min - \max$

problem. Therefore, the senses of the objectives are simply interchanged since we can model a decrement on the flow of the edges by defining negative values of d_e. In addition, the MFI problem is usually formulated in an arc-based notation, therefore we consider the corresponding arc-based notation introduced above. The MFI problem is proven to be strongly $\mathcal{NP}$-Complete in Wood (1993) even when $b_{ij} = 1$ for all $(i, j) \in \mathcal{A}$. The works of Wollmer (1963) and Lubore *et al.* (1971) address the most vital link for the MFI. Lubore *et al.* (1971) improves the work of Wollmer (1963) by introducing a labeling algorithm that reduces the number of candidates to be the most vital link. Basically, their algorithms calculates the maximum flow on $\mathcal{D}_{ij} = (V, A \setminus \{(i, j)\})$ for each arc (i, j) in the candidate list.

The work of Wollmer (1964) tackles the $\bar{B}$-most vital links for the MFI problem. Their solving method reduces the $\bar{B}$-most vital links of the MFI problem to a problem related to the $\bar{B}$-most vital links of the SPI problem due to the equivalence between the minimum cut and the computation of the shortest path in the topological dual of the network. This approach is only valid for planar graphs. Ratliff *et al.* (1975) characterizes the $\bar{B}$ links to remove as a particular cut of $\mathcal{D}$. This cut is obtained by sequentially modifying the network based on disconnecting sets of $\mathcal{D}$ so that this cut becomes the one with the lowest capacity.

The research in McMasters & Mustin (1970) solves a version the MFI problem considering partial interdiction where the interdictor spends resources per unit of reduced arc capacity. The latter is done by redefining $\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^{|\mathcal{A}|} \mid \sum_{(i,j)\in\mathcal{A}} b_{ij}(c_{ij} - x_{ij}) \leq B,\ x_{ij} \geq 0\ \forall (i, j) \in \mathcal{A}\}$ and by modifying one of the defender's constraints to $y_{ij} \leq c_{ij} - x_{ij}$, i.e. the flow cannot exceed the reduced capacity of the arc. In Wood (1993) the decrement of an arc's capacity is determined by $(1 - x_{ij})c_{ij}$. The author in Phillips (1993) analyses the complexity of the latter problem. She proved the strongly $\mathcal{NP}$-Completeness of the problem for general graphs. For bandwidth-i graphs ($i \geq 3$), grid graphs, series-parallel graphs and Halin graphs the complexity becomes weakly $\mathcal{NP}$-Complete.

Formulation (2.4) where $\mathcal{Y}$ defines a maximum flow problem is proposed in Wood (1993). The formulation is then transformed into a linear single-level problem. The adaptability of this formulation to different variants, such as multiple sources and sinks, multiple commodities, undirected networks or multiple resources, is the main contribution of his work. Finally, he proposes valid inequal-

ities to strengthen the LP relaxation of the model. Two more families of valid inequalities are presented in Altner *et al.* (2010). In addition, they also show that the optimality gap of the MFI problem is not bounded from below by a constant. Two equivalent formulations and a family of valid inequalities under the cardinality constraint are presented in Afshari Rad & Kakhki (2017). Their work proved to provide better LP relaxations than those of Wood (1993) and Altner *et al.* (2010).

2.2.2 Shortest Path Interdiction Problem

In the SPI problem the defender calculates the shortest path between the source node s and the sink node t, whereas the interdictor, with her limited budget, modifies the topology of the graph to maximize the length of the shortest $s-t$ path. The work of Ball *et al.* (1989) shows that the $\bar{B}$-most vital links is $\mathcal{NP}$-hard, therefore the general SPI problem defined by (2.4) is $\mathcal{NP}$-hard as well.

A first algorithm of complexity $\mathcal{O}(|\mathcal{V}|^4)$ of the most vital link for the SPI problem is presented in Corley & Sha (1982). Implicitly, they calculate the shortest path on the graph $(\mathcal{V}, \mathcal{E} \setminus \{e\})$ for each e of the common edges of the shortest paths of $\mathcal{G}$. In Malik *et al.* (1989), the authors start by formally stating that the most vital link is one of the arcs of the shortest $s-t$ path. They use this result and Dijkstra's algorithm for the shortest path problem to develop an algorithm of complexity $\mathcal{O}(|\mathcal{V}|^2)$.

The exact algorithm proposed in Malik *et al.* (1989) for the $\bar{B}$-most vital links of the SPI problem consists in generating and ordering all $s-t$ paths in non-decreasing lengths. The interdiction is attempted on the graph formed by the union of the first l paths. The latter is done sequentially until the interdiction becomes infeasible. Then, the longest path generated is the optimal solution. The complexity of this algorithm is $\mathcal{O}(Cr)$ where C and r represent the number of generated paths and r the complexity of finding all paths from s to all other nodes, respectively. In Ball *et al.* (1989), it is shown that a problem related to the SPI problem, which is solved in polynomial time, provides a lower bound for the optimal value of the SPI problem.

The continuous version of the SPI problem is first addressed in Fulkerson & Harding (1977). The author provides a formulation of this version of the SPI problem and then, its equivalence to a minimum-cost flow problem is determined.

Golden (1978) treats a similar problem where the weight of the path is to be increased by a minimum of τ units. The objective function is to minimize the cost needed to obtain such increment since the increment is given. Again, this problem is shown to be reduced to a minimum-cost flow problem therefore it is solved in polynomial time.

In Israeli & Wood (2002), it is presented the formulation (2.4) for the SPI problem, i.e. the constraints in set $\mathcal{Y}$ are those of a shortest path problem. The authors present a Benders decomposition approach where the restricted master problem maximizes the lengths of paths that belong to a restricted collection of $s-t$ paths by interdicting at least one of the arcs of each path. Given a solution of the master problem, the sub-problem solves a shortest path problem parameterized by the interdiction plan defined by the master problem's solution. The master problem and sub-problem are iteratively solved until the lower and upper bounds are equal. A set covering based decomposition is also considered. The idea of this decomposition is that for a given feasible solution of the SPI problem with value v', the optimal interdiction plan has to interdict at least one arc on all $s-t$ paths that have a length less that v'. If one of these paths is not interdicted by the optimal interdiction plan then the defender can choose this path as his optimal solution which has a worst objective value. Super valid inequalities are introduced, i.e. valid inequalities that eliminate integer feasible solutions, except for optimal ones. Basically, as the Benders decomposition algorithm finds better lower bounds, super valid inequalities are defined to force certain interdiction variables to be positive in order to force the master problem to give a solution of at least the value of the lower bound. Finally, a enhanced Benders decomposition which includes both super valid inequalities and covering cuts is presented. The covering cuts follow the same idea of the set covering based decomposition in which paths that have a length less than the lower bound have to be interdicted.

2.2.3 Minimum Spanning Tree Interdiction Problem

In this section, the related literature with respect to the MSTI problem is presented. In this network problem, the defender seeks to calculate the Minimum Spanning Tree (MST) of $\mathcal{G}$. As in the previous problems, the interdictor uses a limited amount of resources to alter the topology of the network in order to worsen the defender's objective. The MSTI problem was proven to be $\mathcal{NP}$-Hard

in Lin & Chern (1993). As in the SPI problem, it is proven that the edge that, when removed, causes the largest increase in the weight of the MST is one of its own edges. Let T_0 be the edge set that defines the MST of $\mathcal{G}$. A replacement edge of an edge $e \in T_0$ is an edge $e' \in \mathcal{G} \setminus T_0$ such that $T_0 \setminus \{e\} \cup \{e'\}$ defines a spanning tree and this replacement induces the least increment on the weight of T_0. Two algorithms that use this result are proposed in Hsu *et al.* (1991) for the most vital edge problem. The first algorithm takes advantage of the construction of spanning forests embedded in Kruskal's algorithm to compute the replacement edges. The complexity of the latter algorithm is of $\mathcal{O}(|\mathcal{E}|log|\mathcal{E}|)$. Similarly, a second algorithm of complexity $\mathcal{O}(|\mathcal{E}|^2)$ but based on Prim's algorithm is as well presented. The work of Iwano & Katoh (1993) introduces two more algorithms that improve the previous complexities. One of them calculates the replacement edges in $\mathcal{O}(|\mathcal{E}|\alpha(|\mathcal{E}|,|\mathcal{V}|))$ where $\alpha(|\mathcal{E}|,|\mathcal{V}|)$ represents a functional inverse of Ackermann's function (Calude *et al.* (1979)). The second time-amortized algorithm of complexity $\mathcal{O}(|\mathcal{E}| + |\mathcal{V}|log|\mathcal{V}|))$ consists of reducing the set of edges where the replacement edges are searched.

An important property of the $\bar{B}$-most vital edges for the MSTI problem that prunes edges of $\mathcal{G}$ is first stated in Liang & Shen (1997). Let T_0 be the edge set of the MST of $\mathcal{G}$ and T_i the MST of $\mathcal{G}_i = \mathcal{G} \setminus \cup_{j=0}^{i-1} T_j$. Let $\mathcal{G}_{\bar{B}} = \cup_{i=0}^{\bar{B}} T_i$, the union of the spanning trees T_i, $i = 0, ..., \bar{B}$. The reduced graph $G_{\bar{B}}$ is named the sparse weighted $\bar{B}$-edge connected certificate of $\mathcal{G}$. A property of this certificate reads as follows:

Lemma 2.1. *(Liang & Shen (1997)) For any edge set $\mathcal{S}$, $\mathcal{S} \subseteq \mathcal{E}$ and $|\mathcal{S}| < \bar{B}$, the edges of MST of $\mathcal{G}(\mathcal{V}, \mathcal{E} \setminus \mathcal{S})$ belong to $\mathcal{G}_{\bar{B}}$.*

The MSTI problem is then equivalently defined on $\mathcal{G}_{\bar{B}}$ instead of $\mathcal{G}$. Liang & Shen (1997) also introduce a sequential algorithm that runs in $\mathcal{O}(|\mathcal{V}|^{\bar{B}+1})$. The algorithm consists of interdicting a subset of edges of T_0 and then removing this edge set of $\mathcal{G}$ and recalculating the MST on the new graph, then the algorithm interdicts another subset of edges of the new tree and starts over. Once the $\bar{B}$ edges are interdicted, the algorithm verifies if the interdiction plan and the corresponding tree provide a better increment on the weight of the tree. The latter steps are done on all possible subsets of edges of T_0.

In Shen (1999), the author proposes two-random algorithms. The first one has a time complexity of $\mathcal{O}(|\mathcal{E}||\mathcal{V}|)$ with $\exp(-\frac{\sqrt{2B}}{B-2})$ of probability of success. The second one is in fact an approximation algorithm that runs in $O(|\mathcal{E}||\mathcal{V}|+B2|\mathcal{V}|\log|\mathcal{V}|)$ time complexity with $\exp(-\frac{\sqrt{2B}}{B-2})$ probability of success; it provides solutions within a factor of 2 with respect to the optimal solution. The randomness of the algorithms comes from the fact that the replacement edges for the edges in T_0 are chosen randomly and uniformly from $\mathcal{G} \setminus T_0$.

The work of Frederickson & Solis-Oba (1999) introduces a greedy algorithm of $\mathcal{O}(|\mathcal{E}|^2|\mathcal{V}|^3\log(|\mathcal{V}|^2/|\mathcal{E}|))$ time complexity for the continuous version. The interdictor consumes resources per unit of weight augmentation of each edge. The coverage of an edge set $\mathcal{S}$ is defined as the minimum amount of edges that any MST has to share with the edge set $\mathcal{S}$. The greedy algorithm begins by choosing an edge set $\mathcal{S}$, then the weights of the edges of $\mathcal{S}$ are increased so that its coverage decreases. Once this happens, a new edge set $\mathcal{S}$ is chosen, and the process is repeated until the budget is consumed. Baïou & Barahona (2008) extends this version by using piece-wise linear functions for weight augmentations. The authors adapt the algorithm of Frederickson & Solis-Oba (1999) and also present a combinatorial mathematical formulation for this extension.

A Branch-and-Bound-like enumerative algorithm for the $\bar{B}$-most vital edges is introduced in Bazgan *et al.* (2012). The authors take advantage of the particular structure provided by Lemma 2.1 to develop a Branch-and-Bound scheme that only enumerates the relevant interdiction plans and MSTs associated to them. The algorithm is improved by defining upper and lower bounds for each node of the Branch-and-Bound search tree, so that other nodes can be fathomed. In addition, mathematical formulations for the $\bar{B}$-most vital edges are proposed. However, they proved to be inefficient when compared to their enumerative algorithm.

2.2.4 Variants and Extensions

In Section 2.1, the most pertinent literature of the MFI, SPI and MSTI problems is presented. Two types of interdiction and two definitions of the interdictor's budget constraint are identified as a key aspect when it comes to proposing a resolution method. However, the problem is essentially the same, i.e. the interdiction of links as to worsen the network operator's objective. In the current

section, we present variants and extensions of the three considered interdiction problems and point out the main differences with respect to the classical network interdiction problems.

Wood (1993) and Israeli & Wood (2002) present a straightforward extension to deal with undirected networks in the MFI and SPI problems, respectively. The authors reduce the problems to work with a directed network by defining the problems on the sets $\mathcal{A}' = \{(i,j) \,|\, \{i,j\} \in \mathcal{E}\}$ and $\mathcal{A}'' = \{(j,i) \,|\, (i,j) \in \mathcal{A}'\}$. Furthermore, the research in Wood (1993) and Israeli & Wood (2002) also extend the MFI and SPI problems to consider multiple interdiciton resources. Multiple resource extension is defined in two different ways. We can consider that a link can be interdicted by any one of the resources available to the interdictor. Let $\mathcal{R}$ be the number of different resources and x_{ijr} a decision variable that is equal to one if arc (i,j) is interdicted using resource r and zero, otherwise. The feasible set of the interdictor then extends to $\mathcal{X} := \{\mathbf{x} \in \{0,1\}^{|\mathcal{A}|\times|\mathcal{R}|} \mid \sum_{(i,j)\in\mathcal{A}} b_{ijr}x_{ijr} \leq B_r\}$. We may consider as well that an edge needs a number of different resources in order to be interdicted. In that case, we define $\mathcal{X} := \{\mathbf{x} \in \{0,1\}^{|\mathcal{E}|} \mid \sum_{e\in\mathcal{E}} b_{er}x_e \leq B_r, \forall r \in \mathcal{R}\}$.

The removal of a node together with all its incident links is named node interdiction. The MFI and SPI problems subject to node interdiction are considered in Corley & Chang (1974) and Corley & Sha (1982), respectively. Both articles use the same approach. It is proven that node interdiction can be solved as a $\bar{B}$-most vital links problem in an augmented network. The construction of the augmented graph consists in replacing each node $i \in \mathcal{N}$, except for the source and sink nodes, by two new nodes i' and i'' along with a dummy arc (i'', i'). An arc (i', j'') is defined with capacity or length $c_{i'j''} = c_{ij}$ for each arc in $(i,j) \in \mathcal{A}$. For the dummy arcs, $c_{i''i'} = 0$ in the SPI problem or $c_{i''i'} = \min\{\sum_{(i,j)\in\delta^+(i)} c_{ij}, \sum_{(i,j)\in\delta^-(i)} c_{i,j}\}$ for the MFI problem. Node interdiction is then reduced to find the dummy arcs whose removal maximizes/minimizes the defender's objective. A second type of node interdiction, named node-wise limited interdiction, is defined in Khachiyan *et al.* (2006) and Khachiyan *et al.* (2008). In this variant, a fixed number of k outgoing arcs are to be removed for each node in the network. Node removal in the MSTI problem is not considered since the removal of a node disconnects the graph which disables the defender of finding a MST.

A multi-objective problem consists in optimizing several objective functions simultaneously. The authors in Royset & Wood (2007) and Rocco S. *et al.* (2010) treat the bi-objective variants of the MFI and SPI problems, i.e. the minimization of the interdiction costs and the maximization of the maximum flow or the minimization of the shortest path are optimized at the same time. The research in Rocco S. & Ramirez-Marquez (2010) considers the same problem. In addition, the authors of the article introduce a new follower behavior in which the follower wishes to restore the flow lost after interdiction over time, then a tri-objective MFI problem is defined where the third objective maximizes flow restoration.

The fortification of network links drawn attention from the scholars in the last decade (Brown *et al.* (2006), Sadeghi *et al.* (2017), Wu & Conejo (2017), Wu *et al.* (2021)). The network user, limited by a budget, selects a subset of edges to reinforce, i.e. edges that are impossible to destroy by the interdictor. This variant makes it possible to protect some of the most important links for the follower.

Stochastic optimization is the domain of optimization research where randomness is present in functions or variables. Cormican *et al.* (1998) and Janjarassuk & Linderoth (2008) consider that the leader's decision to interdict a given arc may result in failure. For each link in the network, there is an independent random variable that is equal to one with given probability, i.e. the arc is destroyed, and equal to zero with the complement, i.e. the arc remains intact. The latter is called uncertain interdiction and is one of several stochastic variants of interdiction problems. Asymmetric information (Sullivan *et al.* (2014)), uncertainty of arc capacities (Cormican *et al.* (1998)), stochastic networks (Hemmecke *et al.* (2003), Held *et al.* (2005a)) and uncertainty of source/destinations nodes (Pan (2005), Morton (2010)) are some more examples of such variants.

2.2.5 Applications

The main application of IGs is to identify critical infrastructure in a system that would cause considerable losses if destroyed. Vulnerability analyses of electrical grids is carried out in Salmeron *et al.* (2009), Delgadillo *et al.* (2010) and Yuan *et al.* (2014). The research of Wood (1993), Jabarzare *et al.* (2019) and Malaviya *et al.* (2012) plays a role in the combat against drug trafficking. Protection of crucial sites or infrastructure against terrorist attacks or natural disasters are the

research objective of Salmeron *et al.* (2004), Powell (2007) and Zhuang & Bier (2007).

The radio communication interdiction problem is addressed in Tanergüçlü *et al.* (2019) where, in a war-like context, the objective is to place transmitters on the battlefield to secure communication and thus mitigate the damage caused by radio jamming attacks committed by enemy forces.

The vulnerability of cyber-physical networks or systems is the scope of Baycik *et al.* (2018) and Etesami & Başar (2019). Internet and smart grids became an intrinsic part of communication networks. Therefore, the authors model this type of systems in order to protect them against disruption such as hackers, terrorism or information leakage.

The research in Konrad *et al.* (2017) analyze solutions to tackle human trafficking from the perspective of the discipline of operation research. The contribution of their work is the analysis of the issue of human trafficking and the different techniques, such as network interdiction, that can be employed.

To end this section, we present in Table 2.1 a summary of the most pertinent articles. Additionally, the literature is classified by the main assumptions and similarities of interdiction problems.

This thesis focuses on the MSTI problem. The works on the interdiction version of the MST are fairly scarce compared to those of SPI and MFI problems. Furthermore, most of them focus in specific cases such as the $\bar{B}$-most vital edges. The work presented in this manuscript is located among the literature in Table 2.1 as follows. The mathematical formulations and algorithms presented in the following sections are all capable of solving instances with any type of edge interdiction and with any budget constraint. The majority of studies have neglected both partial interdiction and knapsack budget constraint assumptions for the MSTI problem. Therefore, up to some extent, our resolution methods are able to solve the generalization of the MSTI problem.

2. LITERATURE REVIEW

Table 2.1: Main attributes of Interdiction Problems

Problem	Reference	Type of Interdiction			Type of Network		Approach followed		Budget constraint		Underlying Formulation				
		ER	EA	NR	Di	U	D	S	C	K	LP	MIP	BP	MO	N/A
Shortest Path	Wollmer (1964)	★				★	★		★						★
	Corley & Chang (1974)			★		★	★		★						★
	Wood (1993)	★	★		★	★	★		★	★		★			
	Cormican *et al.* (1998)	★			★			★		★		★			
	Burch *et al.* (2003)	★	★			★	★			★		★			
	Brown *et al.* (2005)	★	★		★		★			★		★			
	Lim & Smith (2007)	★	★		★		★			★		★			
	Bertsimas *et al.* (2016)	★			★			★	★			★			
Maximum Flow	Fulkerson & Harding (1977)		★			★	★			★	★				
	Ball *et al.* (1989)		★		★		★			★					★
	Israeli & Wood (2002)	★	★		★		★			★		★			
	Hemmecke *et al.* (2003)	★	★	★	★			★		★		★			
	Bayrak & Bailey (2008)	★	★		★		★			★		★	★		
	Khachiyan *et al.* (2008)		★	★	★		★		★				★		
	Rocco S. & Ramirez-Marquez (2010)	★	★			★	★							★	
	Sefair & Smith (2016)		★			★	★		★						★
	Sadeghi *et al.* (2017)	★	★		★		★			★		★			
	Pay *et al.* (2019)	★	★			★		★		★		★			
Minimum Spanning Tree	Hsu *et al.* (1991)	★				★	★		★						★
	Lin & Chern (1993)	★				★	★		★						★
	Liang & Shen (1997)	★				★	★		★						★
	Frederickson & Solis-Oba (1999)	★	★			★	★		★						
	Bazgan *et al.* (2012)	★				★	★		★			★			
	Wei *et al.* (2021)	★				★	★		★	★		★			
Others	Pan *et al.* (2003)		★		★		★	★		★		★	★		
	Church *et al.* (2004)			★			★		★			★			
	Held *et al.* (2005b)	★	★		★			★		★		★			
	Morton *et al.* (2007)		★		★			★		★		★			
	Scaparra & Church (2008)			★	★		★		★				★		
	Dinitz & Gupta (2013)	★				★	★			★	★	★			
	Yates & Sanjeevi (2013)		★		★		★			★		★	★		
	Furini *et al.* (2019)			★		★	★		★		★	★			

ER: Edge Removal EA: Edge Augmentation NR: Node Removal Di: Directed
U: Undirected D: Deterministic S: Stochastic C: Cardinality
BP: Bilevel Programming LP: Linear Problem K: Knapsack MO: Multiobjective
MIP: Mixed Integer Problem

Chapter 3

Formulations for the MSTI Problem

For the MSTI problem, the bulk of research presented in Chapter 2 have disregarded both the partial interdiction and knapsack budget constraint assumptions. This chapter presents different mathematical formulations of the MSTI problem that generalizes it by being able to consider both types of edge interdiction and both types of budget constraints.

The first formulation presented in this chapter is proposed in Bazgan *et al.* (2012) and is based on the directed multi-commodity flow model of the MST. The second model uses the linear MST formulation of Martin (1991) to propose a max − min model for the MSTI problem. Constraints that characterize optimal spanning trees given the comparison of every two edges in $\mathcal{E}$ and the reaction with respect to interdiction are defined. We called them optimality constraints and are used to redefine the latter model into a single-level problem.

The max − min formulation is redefined by computing the dual of the inner minimization problem which allows to define a single-level maximization problem. Additionally, a more compact version of the max − min formulation is obtained by modifying and removing unnecessary constraints. This gives place to define a more compact version of the single-level maximization problem.

A model in which only total interdiction is allowed is proposed as well. The latter model is equivalent to the rest of the formulations only under a specific parameter setting. The chapter ends with numerical experiments on instances defined on graphs of different sizes, as well as with all types of interdiction and budget constraints.

3.1 Bazgan *et al.* (2012) Formulation for the MSTI Problem

Two mathematical formulations for the $\bar{B}$-most vital edges interdiction problem are presented in Bazgan *et al.* (2012). We begin this chapter by reintroducing them. The present formulation is based on the directed multi-commodity flow model (Magnanti & Wolsey (1995)) for the MST. In this MST formulation, node 1 is chosen as root node and a commodity is associated with any other node l. One unit of commodity l is sent from the root node to each other node l. The flow starts in node 1 and travels to all commodities, i.e. each node $l \neq 1$. Let variables f_{ij}^l represent the flow of commodity l in arc (i,j). The spanning tree is defined by variables y_{ij}. Binary variables x_{ij} define the interdiction plan. Lastly, let $n = |\mathcal{V}|$. The mixed-integer formulation that describes the MSTI problem reads as follows:

$$[\text{MSTI}_{\text{Ba}}] \quad \max_{\mathbf{x}} \min_{\mathbf{f},\mathbf{y}} \sum_{\{i,j\}\in\mathcal{E}} (c_{ij} + d_{ij}x_{ij})(y_{ij} + y_{ji}) \tag{3.1a}$$

s.t.

$$\sum_{(j,1)\in\mathcal{A}} f_{j1}^l - \sum_{(1,j)\in\mathcal{A}} f_{1j}^l = -1 \qquad : \{\alpha_1^l\} \qquad \forall l \in \mathcal{V}\setminus\{1\} \tag{3.1b}$$

$$\sum_{(j,i)\in\mathcal{A}} f_{ji}^l - \sum_{(i,j)\in\mathcal{A}} f_{ij}^l = 0 \qquad : \{\alpha_i^l\} \qquad \forall i,l \in \mathcal{V}\setminus\{1\}, i \neq l \tag{3.1c}$$

$$\sum_{(j,l)\in\mathcal{A}} f_{jl}^l - \sum_{(l,j)\in\mathcal{A}} f_{lj}^l = 1 \qquad : \{\alpha_l^l\} \qquad \forall l \in \mathcal{V}\setminus\{1\} \tag{3.1d}$$

$$f_{ij}^l \leq y_{ij} \qquad : \{\sigma_{ij}^l\} \qquad \forall (i,j) \in \mathcal{A}, \forall l \in \mathcal{V}\setminus\{1\} \tag{3.1e}$$

$$\sum_{(i,j)\in\mathcal{A}} y_{ij} = n - 1 \qquad : \{\mu\} \tag{3.1f}$$

$$\sum_{\{i,j\}\in\mathcal{E}} b_{ij}x_{ij} \leq B \tag{3.1g}$$

$$f_{ij}, y_{ij} \geq 0 \qquad \forall (i,j) \in \mathcal{A} \tag{3.1h}$$

$$x_{ij} \in \{0,1\} \qquad \forall \{i,j\} \in \mathcal{E} \tag{3.1i}$$

The maximization of the total cost of the MST by augmenting the weight of the arcs is represented by (3.1a). Constraints (3.1b)-(3.1d) define flow balance for

each node. If $y_{ij} = 0$, constraint (3.1e) forces the flow on the corresponding arc to be zero. Constraint (3.1f) imposes that the MST is composed by exactly $n-1$ arcs. Therefore, solutions given by (3.1b)-(3.1f) represent a spanning tree. The interdiction budget constraint is expressed by (3.1g) which is originally stated in Bazgan *et al.* (2012) as a cardinality constraint. In this manuscript, it is replaced by a knapsack constraint to generalize the formulation. Positive and integrality requirements are expressed in (3.1h) and (3.1i). [MSTI_{Ba}] is a $\max - \min$ problem with a quadratic objective function.

We refer to the inner problem or inner part of a $\max - \min$ formulation as the reminder of the model when the values of the variables with respect to the maximization problem are fixed. The inner problem of [MSTI_{Ba}] when fixing variables x_{ij} is defined by (3.1a)-(3.1f) and (3.1h).

To establish a single-level model, the inner part of [MSTI_{Ba}] is substituted with its dual formulation. The dual variables are shown next to their respective constraints in model (3.1). The dual formulation is the following:

$$[\text{MSTI}_{\text{DBa}}] \quad \max_{\mathbf{x},\boldsymbol{\sigma},\boldsymbol{\alpha},\boldsymbol{\mu}} \sum_{\substack{l \in \mathcal{V} \\ l \neq 1}} (\alpha_l^l - \alpha_1^l) + (n-1)\mu \tag{3.2a}$$

s.t.

$$\sigma_{ij}^l \geq \alpha_j^l - \alpha_i^l \qquad \forall (i,j) \in \mathcal{A}, \forall l \in \mathcal{V} \setminus \{1\} \tag{3.2b}$$

$$\sum_{l \neq 1} \sigma_{ij}^l + \mu \leq c_{ij} + d_{ij} x_{ij} \qquad \forall \{i,j\} \in \mathcal{E} \tag{3.2c}$$

$$\sum_{l \neq 1} \sigma_{ji}^l + \mu \leq c_{ij} + d_{ij} x_{ij} \qquad \forall \{i,j\} \in \mathcal{E} \tag{3.2d}$$

$$\sum_{\{i,j\} \in \mathcal{E}} b_{ij} x_{ij} \leq B \tag{3.2e}$$

$$x_{ij} \in \{0,1\} \qquad \forall \{i,j\} \in \mathcal{E} \tag{3.2f}$$

$$\sigma_{ij}^l \geq 0 \qquad \forall (i,j) \in \mathcal{A}, \forall l \in \mathcal{V} \setminus \{1\} \tag{3.2g}$$

$$\alpha_i^l \geq 0 \qquad \forall i \in \mathcal{V}, \forall l \in \mathcal{V} \setminus \{1\} \tag{3.2h}$$

$$\mu = \textit{unrestricted} \tag{3.2i}$$

The quadratic terms in [MSTI_{Ba}] objective function (3.1a) are no longer present in formulation (3.2). Thus, we obtain a mixed-integer linear formula-

tion.

3.2 MSTI Formulation Based on Martin (1991) MST Formulation

In Martin (1991), the author proposes a formulation for the MST that meets the condition of Total Dual Integrality. This condition proves the integrality of the polyhedron defined by the constraints of the model. An integral polyhedron is a polyhedron whose vertices are all integer (Cook *et al.* (1997)). Thus the model provides an integer solution even when the integrality requirement on the variables respect to the edge selection are removed. To present the model, first, let us define $\mathcal{A}_m = \{(m, j) \in \mathcal{A}, \mid j \in \mathcal{V}\}$ for all $m \in \mathcal{V}$ as the set of all arcs with node m as origin. Let y_e be a variable that is equal to 1 if edge e is part of the MST, and 0 otherwise. Each edge e is disaggregated into k-arcs, i.e. (k, i, j) and (k, j, i) for all $k \in \mathcal{V}$ and for all $e = \{i, j\} \in \mathcal{E}$. Let z_{kij} be 1 if arc (i, j) is chosen as part of the directed MST, and 0 otherwise. Let $\mathbf{y}$-tree and $\mathbf{z}_k$-tree denote the trees defined by y_e and z_{kij} variables, respectively. The MST formulation reads as follows:

$$[\text{MST}] \quad \min_{\mathbf{y},\mathbf{z}} \sum_{e \in \mathcal{E}} c_e y_e \tag{3.3a}$$

s.t.

$$\sum_{e \in \mathcal{E}} y_e = n - 1 \tag{3.3b}$$

$$z_{kij} + z_{kji} = y_e \qquad \forall k \in \mathcal{V}, e = \{i, j\} \in \mathcal{E} \tag{3.3c}$$

$$\sum_{(m,j) \in \mathcal{A}_m} z_{kmj} \leq 1 \qquad \forall k, m \in \mathcal{V}, k \neq m \tag{3.3d}$$

$$\sum_{(k,j) \in \mathcal{A}_k} z_{kkj} \leq 0 \qquad \forall k \in \mathcal{V} \tag{3.3e}$$

$$y_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{3.3f}$$

$$z_{kij} \geq 0 \qquad \forall k, i, j \in \mathcal{V}, i \neq j \tag{3.3g}$$

Objective function (3.3a) maximizes the weight of the $\mathbf{y}$-tree. Exactly $n - 1$ edges are selected to define a MST thanks to (3.3b). The variables z_{kij} represent

k directed trees induced by variables y_e, whose arcs define paths from any node towards node k. Constraints (3.3c)-(3.3e) define cycle elimination constraints for the $\mathbf{z}_k$-trees. In turn, they eliminate the cycles of the $\mathbf{y}$-tree. If there were an undirected cycle in the $\mathbf{y}$-tree that contains a given node k, each $\mathbf{z}_k$-tree will have, by (3.3c), an analogous cycle that contains node k as well. A directed cycle containing k, i.e. a cycle that starts and ends in node k, is impossible by (3.3d) since it does not allow the selection of an arc directed out of k in a $\mathbf{z}_k$-tree. Non-directed cycles are those that for a node $i \in V$, $i \neq k$, there exists two arcs with origin in i. This is also forbidden by (3.3e) since only one directed out arc is allowed for each node different of k. Thus, formulation (3.3) defines a MST of $\mathcal{G}$. The $\max - \min$ formulation that defines the interdiction of the MST, indicated with [MSTI] is the following:

$$[\text{MSTI}] \quad \max_{\mathbf{x}} \min_{\mathbf{y},\mathbf{z}} \sum_{e \in \mathcal{E}} (c_e + d_e x_e) y_e \tag{3.4a}$$

s.t.

$$\sum_{e \in \mathcal{E}} y_e = n - 1 \qquad : \{\beta\} \tag{3.4b}$$

$$z_{kij} + z_{kji} = y_e \qquad : \{\sigma_{ke}\} \quad \forall k \in \mathcal{V}, e = \{i,j\} \in \mathcal{E} \tag{3.4c}$$

$$\sum_{(m,j) \in \mathcal{A}_m} z_{kmj} \leq 1 \qquad : \{\tau_{km}\} \quad \forall k, m \in \mathcal{V}, k \neq m \tag{3.4d}$$

$$\sum_{(k,j) \in \mathcal{A}_k} z_{kkj} \leq 0 \qquad : \{\theta_k\} \quad \forall k \in \mathcal{V} \tag{3.4e}$$

$$\sum_{e \in \mathcal{E}} b_e x_e \leq B \tag{3.4f}$$

$$y_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{3.4g}$$

$$x_e \in \{0,1\} \qquad \forall e \in \mathcal{E} \tag{3.4h}$$

$$z_{kij} \geq 0 \qquad \forall k, i, j \in \mathcal{V}, i \neq j \tag{3.4i}$$

The objective function (3.4a) maximizes the weight of the edges chosen to be part of the MST. Constraints (3.4c)-(3.4e) are as in [MST]. The interdiction budget constraint is imposed by (3.4f). Constraints (3.4g)-(3.4i) define the variables.

Note that formulation [MSTI] is also a $\max - \min$ quadratic model. Two different approaches are followed to propose a single-level formulation. The first

one is the development of optimality constraints for the computation of the MST. This allows us to remove the minimization problem and thus to obtain a single-level problem. Subsequently, a classic linearization technique on the term $x_e y_e$ is used. This is presented in Section 3.3. The second approach is the one followed in Section 3.1, the inner part of the formulation is a linear program which can be replaced by its dual. This technique is applied to an equivalent formulation of [MSTI] that leads to an even more compact formulation. This is presented in Section 3.4.

3.3 Reformulation using Optimality Constraints

This section starts by stating a well-known optimal condition for the MST (Schrijver (2003)) which is later used to develop optimality constraints for formulation [MST].

Theorem 3.1. *A tree T of $\mathcal{G}$ is a MST if and only if $c_f \geq c_e, \forall e \in T, f \in \mathcal{E} \setminus T$ such that $T \setminus \{e\} \cup \{f\}$ defines a spanning tree.*

Let T^* be the MST of $\mathcal{G}$ and let $e = \{i, j\}, f = \{u, v\} \in \mathcal{E}$ be two edges that satisfy the conditions of Theorem 3.1. Due to constraints (3.3c), we have $z_{uij} + z_{uji} = 1$ and $z_{vij} + z_{vji} = 1$. Let us suppose that $z_{vij} = 1$. This means that in the $\mathbf{z}_v$-tree corresponding to T^* there exist a unique directed path from i to v that uses arc (i, j). Therefore, $z_{uij} = 0$ since there cannot be a path from i to u that uses arc (i, j) because $\{u, v\} \notin T^*$. Analogously, $z_{uji} = 0$ if we assume $z_{vji} = 1$. Thus, z_{vij} and z_{uij} (or z_{vji} and z_{uji}) cannot be simultaneously 1. It follows that $z_{uij} + z_{vji} = 2$ (or $z_{uji} + z_{vij} = 2$). Theorem 3.1 can be equivalently stated as follows.

Theorem 3.2. $(\mathbf{y}^*, \mathbf{z}^*)$ *is an optimal solution of [MST] if and only if $z^*_{uij} + z^*_{vji} = 2$ or $z^*_{uji} + z^*_{vij} = 2$ for all $e = \{i, j\}, f = \{u, v\}$ such that $c_f \geq c_e$.*

In order to integrate the optimality constraints in the formulations for the MSTI problem, Theorem 3.2 can be equivalently stated as:

Theorem 3.3. $(\mathbf{y}^*, \mathbf{z}^*)$ *is an optimal solution of [MST] if and only if $z^*_{uij} + z^*_{vji} \leq 1$ and $z^*_{uji} + z^*_{vij} \leq 1$ for all $e = \{i, j\}, f = \{u, v\}$ such that $c_f < c_e$.*

Conditions of Theorem 3.3 can be strengthened to obtain the following theorem.

Theorem 3.4. $(\mathbf{y}^*, \mathbf{z}^*)$ *is an optimal solution of [MST] if and only if* $z^*_{uij} + z^*_{vji} \leq y^*_e$ *and* $z^*_{uji} + z^*_{vij} \leq y^*_e$ *for all* $e = \{i, j\}, f = \{u, v\}$ *such that* $c_f < c_e$.

Proof. **Case 1**. When the edge e is selected the corresponding variable $y^*_e = 1$ and the conditions are as in Theorem 3.3.
Case 2. When the edge e is not selected the corresponding variable $y^*_e = 0$ and due to constraints (3.4c) $z^*_{kij} = z^*_{kji} = 0$, for all $k \in \mathcal{V}$. □

Note that, depending on the interdiction plan, the relation between the cost of the edges may change. As an example, it may indeed happen that $c_e < c_f$, but $c_f < c_e + d_e$. As a consequence, the optimality conditions based on Theorem 3.4 are further developed. To generate an appropriate condition for the different outcomes regarding the weights of the edges, several sets on the space $\mathcal{E} \times \mathcal{E}$ are introduced and listed below:

$$
\begin{aligned}
\mathcal{E}_1 &= \{(e, f) \mid c_f + d_f < c_e \ \forall e, f \in \mathcal{E}\} \\
\mathcal{E}_2 &= \{(e, f) \mid c_f < c_e,\ c_f + d_f < c_e + d_e,\ c_e \leq c_f + d_f \ \forall e, f \in \mathcal{E}\} \\
\mathcal{E}_3 &= \{(e, f) \mid c_f < c_e,\ c_e + d_e \leq c_f + d_f \ \forall e, f \in \mathcal{E}\} \\
\mathcal{E}_4 &= \{(e, f) \mid c_e \leq c_f,\ c_f + d_f < c_e + d_e \ \forall e, f \in \mathcal{E}\} \\
\mathcal{E}_5 &= \{(e, f) \mid c_e \leq c_f,\ c_e + d_e \leq c_f + d_f,\ c_f < c_e + d_e \ \forall e, f \in \mathcal{E}\}
\end{aligned}
$$

The sets $\mathcal{E}_i \subset \mathcal{E} \times \mathcal{E}$ represent all the different relations between edges given their weight and their augmented weight. For each set we can now derive optimality constraints.

Set $\mathcal{E}_1$: Note that on $\mathcal{E}_1$, regardless the interdiction plan, the weight of $e = \{i, j\}$ is always grater than the weight of $f = \{u, v\}$. Therefore, we can directly apply the conditions of Theorem 3.4 and define $z_{uji} + z_{vij} \leq y_e$ as optimality constraint for any pair (e, f) of $\mathcal{E}_1$ with $e = \{i, j\}$ and $f = \{u, v\}$.

Set $\mathcal{E}_3$: In this case the optimality constraints are modified as $z_{uji} + z_{vij} \leq y_e + x_f$. This imposes the optimality condition only when edge f is not interdicted. Indeed, when f is interdicted, the weight of edge e is lower than the weight of edge f and the corresponding constraint is deactivated by adding variable x_f.

Set $\mathcal{E}_4$: The conditions of Theorem 3.4 are met when edge e is interdicted. Thus, $z_{uji} + z_{vij} \leq y_e - x_e + 1$ is defined for any element of this set. Note that the latter expression is relaxed if edge e is not interdicted.

Set $\mathcal{E}_5$: The optimality conditions for this set are considered only in the specific case when edge e is interdicted and edge f is not. Once more, the optimality constraints are modified to $z_{uji} + z_{vij} \leq y_e - x_e + x_f + 1$. In fact, the right hand side of this expression becomes $y_e + 1$ in any other case.

Set $\mathcal{E}_2$: Since the weights of the edges of any element of $\mathcal{E}_2$ can meet three different cases, the constraints defined for $\mathcal{E}_3$ and $\mathcal{E}_4$ are used. This way, one of those two conditions imposes Theorem 3.4 while the other becomes relaxed.

For each pair of edges (e, f), $e, f \in \mathcal{E}$, it is considered that $e = \{i, j\}$ and $f = \{u, v\}$ unless stated otherwise. Introducing the optimality constraints allows to remove the minimization over $\mathbf{y}$ and $\mathbf{z}$ in formulation [MSTI] and to obtain the following single-level non-linear formulation for the MSTI problem.

$$[\text{MSTI}_\text{Q}] \quad \max_{\mathbf{x},\mathbf{y},\mathbf{z}} \sum_{e\in\mathcal{E}} (c_e + d_e x_e) y_e \tag{3.5a}$$

s.t.

$$\sum_{e\in\mathcal{E}} y_e = n - 1 \tag{3.5b}$$

$$z_{kij} + z_{kji} = y_e \qquad \forall k \in \mathcal{V}, e = \{i, j\} \in \mathcal{E} \tag{3.5c}$$

$$\sum_{(m,j)\in\mathcal{A}_m} z_{kmj} \leq 1 \qquad \forall k, m \in \mathcal{V}, k \neq m \tag{3.5d}$$

$$\sum_{(k,j)\in\mathcal{A}_k} z_{kkj} \leq 0 \qquad \forall k \in \mathcal{V} \tag{3.5e}$$

$$z_{uji} + z_{vij} \leq y_e \qquad \forall (e, f) \in \mathcal{E}_1 \tag{3.5f}$$

$$z_{uji} + z_{vij} \leq y_e + x_f \qquad \forall (e, f) \in \mathcal{E}_2, \forall (e, f) \in \mathcal{E}_3 \tag{3.5g}$$

$$z_{uji} + z_{vij} \leq y_e - x_e + 1 \qquad \forall (e, f) \in \mathcal{E}_2, \forall (e, f) \in \mathcal{E}_4 \tag{3.5h}$$

$$z_{uji} + z_{vij} \leq y_e - x_e + x_f + 1 \qquad \forall (e, f) \in \mathcal{E}_5 \tag{3.5i}$$

$$\sum_{e\in\mathcal{E}} b_e x_e \leq B \tag{3.5j}$$

$$x_e \in \{0, 1\} \qquad \forall e \in \mathcal{E} \tag{3.5k}$$

$$y_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{3.5l}$$

$$z_{kij} \geq 0 \qquad \forall k, i, j \in \mathcal{V}, i \neq j \tag{3.5m}$$

Constraints (3.5f)-(3.5i) are the optimality conditions discussed above. They ensure that the spanning tree determined by (3.5c)-(3.5e) is of minimum cost, thus a MST. The objective function (3.5a) is non-linear due to the product of the variables y_e and x_e. Therefore, a classic linearization technique and boundary inequalities are used and presented next.

First, we redefine y_e as a binary variable. Note that y_e variables are linear in [MST] due to Total Dual Integral properties. Let $t_e = x_e y_e$ for all $e \in \mathcal{E}$, then t_e is a binary variable that takes the value of 1 only if both x_e and y_e are equal to 1 and 0, otherwise. The linearization of formulation [MSTI_Q] reads as follows:

$$[\text{MSTI}_{\text{Opt}}] \quad \max_{\mathbf{t},\mathbf{x},\mathbf{y},\mathbf{z}} \sum_{e \in \mathcal{E}} (c_e y_e + d_e t_e) \tag{3.6a}$$

s.t.

$$\sum_{e \in \mathcal{E}} y_e = n - 1 \tag{3.6b}$$

$$z_{kij} + z_{kji} = y_e \qquad \forall k \in \mathcal{V}, e = \{i, j\} \in \mathcal{E} \tag{3.6c}$$

$$\sum_{(m,j) \in \mathcal{A}_m} z_{kmj} \leq 1 \qquad \forall k, m \in \mathcal{V}, k \neq m \tag{3.6d}$$

$$\sum_{(k,j) \in \mathcal{A}_k} z_{kkj} \leq 0 \qquad \forall k \in \mathcal{V} \tag{3.6e}$$

$$z_{uji} + z_{vij} \leq y_e \qquad \forall (e, f) \in \mathcal{E}_1 \tag{3.6f}$$

$$z_{uji} + z_{vij} \leq y_e + x_f \qquad \forall (e, f) \in \mathcal{E}_2, \forall (e, f) \in \mathcal{E}_3 \tag{3.6g}$$

$$z_{uji} + z_{vij} \leq y_e - x_e + 1 \qquad \forall (e, f) \in \mathcal{E}_2, \forall (e, f) \in \mathcal{E}_4 \tag{3.6h}$$

$$z_{uji} + z_{vij} \leq y_e - x_e + x_f + 1 \qquad \forall (e, f) \in \mathcal{E}_5 \tag{3.6i}$$

$$t_e \leq x_e \qquad \forall e \in \mathcal{E} \tag{3.6j}$$

$$t_e \leq y_e \qquad \forall e \in \mathcal{E} \tag{3.6k}$$

$$\sum_{e \in \mathcal{E}} b_e x_e \leq B \tag{3.6l}$$

$$t_e, x_e, y_e \in \{0, 1\} \qquad \forall e \in \mathcal{E} \tag{3.6m}$$

$$z_{kij} \geq 0 \qquad \forall k, i, j \in \mathcal{V}, i \neq j \tag{3.6n}$$

When variable x_e or y_e is equal to zero then t_e is equally zero due to the additional constraints (3.6j) and (3.6k). The latter allows to write the product $d_e t_e$ as in the objective function (3.6a). Finally, we obtain [MSTI_{Opt}] which is a single-level mixed-integer linear problem.

3.4 Dual Formulations

A second procedure to determine single-level linear formulations for the MSTI problem consists in replacing the inner part of [MSTI] by its dual formulation. The dual variables for the inner problem constraints are given in model (3.4). The [$\text{MSTI}_{\text{Dual}}$] formulation is presented below.

$$[\text{MSTI}_{\text{Dual}}] \quad \max_{\mathbf{x},\beta,\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\theta}} \beta(n-1) - \sum_{k\in\mathcal{V}} \sum_{\substack{m\in\mathcal{V} \\ m\neq k}} \tau_{km} \tag{3.7a}$$

s.t.

$$\beta + \sum_{k\in\mathcal{V}} \sigma_{ke} \leq c_e + d_e x_e \qquad \forall e \in \mathcal{E} \tag{3.7b}$$

$$\sigma_{ke} + \tau_{km} \geq 0 \qquad \forall k, m \in \mathcal{V}, e = \{m, j\} \in \mathcal{E} \tag{3.7c}$$

$$\sigma_{ke} + \theta_k \geq 0 \qquad \forall k \in \mathcal{V}, e = \{k, j\} \in \mathcal{E} \tag{3.7d}$$

$$\sum_{e\in\mathcal{E}} b_e x_e \leq B \tag{3.7e}$$

$$\tau_{km}, \theta_k \geq 0 \qquad \forall k, m \in \mathcal{V}, m \neq k \tag{3.7f}$$

$$x_e \in \{0, 1\} \qquad \forall e \in \mathcal{E} \tag{3.7g}$$

$$\beta, \sigma_{ke} = \textit{unrestricted} \qquad \forall k \in \mathcal{V}, \forall e \in \mathcal{E} \tag{3.7h}$$

It can be observed that constraints (3.4e) force variables z_{kkj} to be zero. This yields to have more variables and constraints for the dual formulation [$\text{MSTI}_{\text{Dual}}$]. The [MSTI] model can be reformulated so that the constraints (3.4e) are no longer necessary, thus making the dual of the inner problem more compact. As a result, it is possible to have a formulation with one less family of constraints and one less family of variables. Let us introduce $\mathcal{E}^k = \{\{k, i\} \in \mathcal{E} \mid i \in \mathcal{V} \setminus \{k\}\}$ as the set of all edges with k as one of the endpoints. The simplified version of [MSTI] is shown next:

$$[\text{MSTI}_\text{S}] \quad \max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y},\mathbf{z}} \sum_{e\in\mathcal{E}} (c_e + d_e x_e) y_e \tag{3.8a}$$

s.t.

$$\sum_{e\in\mathcal{E}} y_e = n - 1 \tag{3.8b}$$

$$z_{kij} + z_{kji} = y_e \qquad \forall k \in \mathcal{V}, e = \{i,j\} \in \mathcal{E}\backslash\mathcal{E}^k \tag{3.8c}$$

$$\sum_{(m,j)\in\mathcal{A}_m\backslash\{m,k\}} z_{kmj} + y_e \leq 1 \qquad \forall k,m \in \mathcal{V}, k \neq m, e = \{m,k\} \tag{3.8d}$$

$$y_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{3.8e}$$

$$z_{kij} \geq 0 \qquad \forall k,i,j \in \mathcal{V}, i \neq j \tag{3.8f}$$

where $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^{|\mathcal{E}|} \mid \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$. In (3.4), given an edge $e = \{k,j\}$, $y_e = z_{kjk}$ due to (3.4c) and (3.4e). The latter constraint is not defined in [MSTI_S] by not considering edges that are incident to k in (3.8c). In addition, the variable z_{kjk} that appeared in (3.4d) is substituted by y_e, $e = \{k,j\}$, in (3.8d). The procedure performed to obtain [MSTI_Dual] is now applied to [MSTI_S]. A simplified and compact version of [MSTI_Dual] reads as follows:

$$[\text{MSTI}_\text{SDual}] \quad \max_{\mathbf{x},\beta,\boldsymbol{\sigma},\boldsymbol{\tau}} \beta(n-1) - \sum_{k\in\mathcal{V}} \sum_{\substack{m\in\mathcal{V}\\ m\neq k}} \tau_{km} \tag{3.9a}$$

s.t.

$$\beta + \sum_{k\in\mathcal{V}\backslash\{m,h\}} \sigma_{ke} - \tau_{hm} - \tau_{mh} \leq c_e + d_e x_e \qquad \forall e = \{m,h\} \in \mathcal{E} \tag{3.9b}$$

$$\sigma_{ke} + \tau_{km} \geq 0 \qquad \forall k,m \in \mathcal{V}, e = \{m,j\} \in \mathcal{E}\backslash\mathcal{E}^k \tag{3.9c}$$

$$\sum_{e\in\mathcal{E}} b_e x_e \leq B \tag{3.9d}$$

$$\tau_{km} \geq 0 \qquad \forall k,m \in \mathcal{V}, m \neq k \tag{3.9e}$$

$$x_e \in \{0,1\} \qquad \forall e \in \mathcal{E} \tag{3.9f}$$

$$\beta, \sigma_{ke} = \mathit{unrestricted} \qquad \forall k \in \mathcal{V}, \forall e \in \mathcal{E} \tag{3.9g}$$

Constraints (3.4c) and (3.4d) are modified to avoid using indexes kkj. Conse-

quently, in constraints (3.4e) variables z_{kkj} are not needed anymore. We can see the results of these changes in [MSTI$_{\text{SDual}}$]. Constraints (3.9b), the constraints associated with the primal variables y_e, consider two variables τ_{hm} since variables y_e are now considered in (3.8d). Furthermore, two variables σ_{ke} where k is one of the endpoints of edge e are not considered because their corresponding constraints in (3.8c) are not defined. [MSTI$_{\text{SDual}}$] is a more compact formulation of the MSTI problem since it has one less set of variables and one less family of constraints with respect to [MSTI$_{\text{Dual}}$].

3.5 Total Interdiction Formulation

This section focuses on the total interdiction case, namely when $d_e = +\infty$ for all $e \in \mathcal{E}$. Intuitively, since total interdiction means removal of the edge, it is natural to claim that an edge may be interdicted or selected, but cannot be both simultaneously. Mathematically this is modeled by the constraints $x_e + y_e \leq 1$, for each $e \in \mathcal{E}$. Therefore, two new sets $\mathcal{E}_6 = \{(e, f) \mid c_f < c_e \ \forall e, f \in \mathcal{E}\}$ and $\mathcal{E}_7 = \{(e, f) \mid c_e \leq c_f \ \forall e, f \in \mathcal{E}\}$ are defined since the comparison between interdicted edges is no longer considered. The new optimality conditions for these sets look as follows. Let $e = \{i, j\}$ and $f = \{u, v\}$:

Set $\mathcal{E}_6$: The optimality condition is necessary only when edge f is not interdicted. Therefore, we define $z_{uji} + z_{vij} \leq y_e + x_f$. In any other case, the inequality is relaxed.

Set $\mathcal{E}_7$: An optimality condition is imposed for this set only when edge e is interdicted. However, $y_e = 0$ when $x_e = 1$ due to $x_e + y_e \leq 1$. Thus, $z_{uji} + z_{vij} \leq 0$. If edge e is not interdicted, then the optimality condition should be relaxed. Therefore, there is no need to define an optimality condition for this set.

The formulation for the total interdiction case then reads as:

$$[\text{MSTI}_{\text{Tot}}] \quad \max_{\mathbf{x},\mathbf{y},\mathbf{z}} \sum_{e \in \mathcal{E}} c_e y_e \tag{3.10a}$$

s.t.

$$y_e \leq 1 - x_e \qquad \forall e \in \mathcal{E} \tag{3.10b}$$

$$\sum_{e \in \mathcal{E}} y_e = n - 1 \tag{3.10c}$$

$$z_{kij} + z_{kji} = y_e \qquad \forall k \in \mathcal{V}, e = \{i,j\} \in \mathcal{E} \qquad (3.10d)$$

$$\sum_{(i,j)\in\mathcal{A}_m} z_{kmj} \leq 1 \qquad \forall k, m \in \mathcal{V}, k \neq m \qquad (3.10e)$$

$$\sum_{(i,j)\in\mathcal{A}_k} z_{kkj} \leq 0 \qquad \forall k \in \mathcal{V} \qquad (3.10f)$$

$$z_{uji} + z_{vij} \leq y_e + x_f \qquad \forall (e,f) \in \mathcal{E}_6 \qquad (3.10g)$$

$$\sum_{e\in\mathcal{E}} b_e x_e \leq B \qquad (3.10h)$$

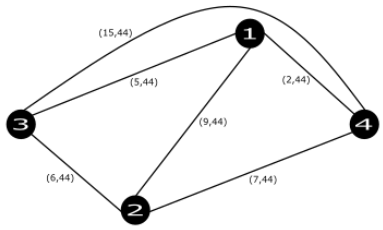
$$x_e \in \{0,1\} \qquad \forall e \in \mathcal{E} \qquad (3.10i)$$

$$y_e \geq 0 \qquad \forall e \in \mathcal{E} \qquad (3.10j)$$

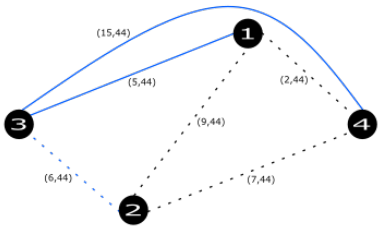
$$z_{kij} \geq 0 \qquad \forall k,i,j \in \mathcal{V}, i \neq j \qquad (3.10k)$$

The objective function (3.10a) does not need to maximize over variables x_e since constraints (3.10b) avoid the use of edges that are interdicted. Constraints (3.10g) are the optimality conditions introduced above. The remaining constraints are as in [MSTI$_{\text{Opt}}$].

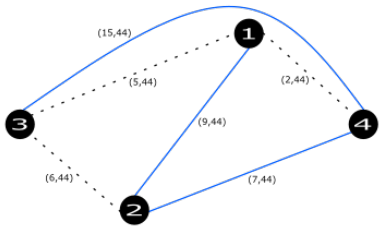
Formulation [MSTI$_{\text{Tot}}$] is not equivalent to [MSTI$_{\text{DBa}}$], [MSTI$_{\text{Opt}}$], [MSTI$_{\text{Dual}}$] nor [MSTI$_{\text{SDual}}$] for all possible settings of parameters B and d_e. To illustrate this point, consider the graph given in Figure 3.1a. Let $B = 4$ and let us assume a cardinality constraint, that is, $b_e = 1$ for all $e \in \mathcal{E}$. A pair (c_e, d_e) where $d_e = \sum_{f\in\mathcal{E}} c_f$ is reported on each edge $e \in \mathcal{E}$. Note that d_e is a large enough value for each $e \in \mathcal{E}$ to model edge removal. The optimal solution given by either formulation [MSTI$_{\text{Opt}}$], [MSTI$_{\text{DBa}}$], [MSTI$_{\text{Dual}}$] or [MSTI$_{\text{SDual}}$] is shown in Figure 3.1b. Dotted edges (either black or blue) represent the interdicted edges while the blue edges represent the MST after interdiction. Figure 3.1c shows the optimal solution of model [MSTI$_{\text{Tot}}$]. The solution illustrated in Figure 3.1b disconnects node 2 from the rest of the graph since the value of B is greater than the connectivity number equal to 3. Moreover, the MST provided by this same solution makes use of the interdicted edge $\{2,3\}$. Thus, we can conclude that the instance is not well-defined since the budget allows the interdictor to disconnect the graph. On the other hand, the [MSTI$_{\text{Tot}}$] solution provides a MST where no interdicted edges are considered. This is since constraints (3.10b) forbid the selection of an interdicted edge, not present in other models, whereas due to constraints (3.10c)-(3.10g) the solution must be a spanning tree. This explains



(a) Initial graph



(b) [MSTI_{Opt}] solution with $B = 4$



(c) [MSTI_{Tot}] solution with $B = 4$

Figure 3.1: Comparison of formulation solutions

the fact that the solution in Figure 3.1c interdicts only three edges when it is possible to interdict four. Therefore, under these parameter settings [MSTI_{Tot}] is not equivalent to [MSTI_{DBa}], [MSTI_{Opt}], [$\text{MSTI}_{\text{Dual}}$] nor [$\text{MSTI}_{\text{SDual}}$].

The assumption of a knapsack constraint also gives place for the formulations to not be equivalent. In this case, the connectivity number with respect to the augmentation costs is calculated by solving $\max\{\sum_{e\in\mathcal{E}} x_e \,:\, \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$. Then, setting a budget greater than or equal to the connectivity number will cause the behavior described above.

Next, it is demonstrated that the formulations [MSTI_{Opt}] and [MSTI_{Tot}], and subsequently the rest of the formulations, are equivalent under the assumption of total interdiction and a budget B less than the connectivity number of $\mathcal{G}$.

Proposition 3.5. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. Let us associate a weight c_e, an interdiction augmentation d_e and an interdiction cost b_e with all edges in $\mathcal{E}$. Let B be less than the connectivity number of $\mathcal{G}$ with respect to the augmentation costs and d_e at least $\max_{e\in\mathcal{E}} c_e + 1 - c_e$ for all $e \in \mathcal{E}$. i.e. d_e is large enough to define total interdiction. Then, [MSTI_{Opt}] and [MSTI_{Tot}] provide the same optimal solution.*

Proof. Let $(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{z}})$ be the optimal solution of [MSTI$_{\text{Tot}}$] under the parameter setting stated in the hypothesis. Note that $\mathcal{E}_1 = \emptyset$, $\mathcal{E}_6 = \mathcal{E}_2 \cup \mathcal{E}_3$ and $\mathcal{E}_7 = \mathcal{E}_4 \cup \mathcal{E}_5$. Constraint (3.10g) is met for all $(e, f) \in \mathcal{E}_6$, then constraint (3.6g) is verified for all (e, f) in $\mathcal{E}_2$ and in $\mathcal{E}_3$. For any $(e, f) \in \mathcal{E}_6 \cap \mathcal{E}_2$, if $\bar{x}_e = 0$ and $\bar{x}_f = 0$ then $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e$ from (3.10g); it follows that $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e - \bar{x}_e + 1$. If $\bar{x}_e = 0$ and $\bar{x}_f = 1$ then $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e + 1$; it follows that $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e - \bar{x}_e + 1$. If $\bar{x}_e = 0$ and $\bar{x}_f = 0$ or 1 then $\bar{z}_{uji} + \bar{z}_{vij} \leq 0$ due to (3.10b); it follows that $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e - \bar{x}_e + 1$. Therefore, (3.6h) is verified for all $(e, f) \in \mathcal{E}_2$.

Note that from (3.10d), $\bar{z}_{uji} \leq \bar{y}_e$ and $\bar{z}_{vij} \leq \bar{y}_e$. It follows that $\bar{z}_{uji} + \bar{z}_{vij} - \bar{y}_e \leq \bar{y}_e$ and from $x_e + y_e \leq 1$ it follows that $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e - \bar{x}_e + 1$. Then constraint (3.6h) is verified for all $(e, f) \in \mathcal{E}_4$. For any $(e, f) \in \mathcal{E}_7 \cap \mathcal{E}_5$, we can write $\bar{z}_{uji} + \bar{z}_{vij} \leq 2$ when $\bar{x}_e = 0$ since there is no need of an optimality condition in this case. It follows that $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e - \bar{x}_e + \bar{x}_f + 1$. When $\bar{x}_e = 1$, $\bar{z}_{uji} + \bar{z}_{vij} \leq 0$ due to (3.10b), it follows that $\bar{z}_{uji} + \bar{z}_{vij} \leq \bar{y}_e - \bar{x}_e + \bar{x}_f + 1$. Thus, (3.6i) is verified for all $(e, f) \in \mathcal{E}_5$. Therefore, the optimal solution of [MSTI$_{\text{Tot}}$] is feasible in [MSTI$_{\text{Opt}}$].

Let $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ be the optimal solution of [MSTI$_{\text{Opt}}$] under the parameter setting stated in the hypothesis. Constraint (3.6g) is met for all (e, f) in $\mathcal{E}_2$ and in $\mathcal{E}_3$, then constraint (3.10g) is verified for all $(e, f) \in \mathcal{E}_6$. We proceed to prove that (3.10b) is verified by $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ by absurd. Suppose there is at least one edge $e' \in \mathcal{E}$ such that $\hat{x}_{e'} + \hat{y}_{e'} = 2$. Let T be the edge set associated to the MST defined by $\hat{\mathbf{y}}$. If edge e' is removed from T then two connected components T_1 and T_2 are obtained. T_1 and T_2 are sub-trees of T. B is small enough to not allow a feasible interdiction plan that defines a cut of $\mathcal{G}$. Therefore, there must exist at least one non-interdicted edge $e'' \neq e'$ such that it connects T_1 and T_2 and such that $x_{e''} + y_{e''} \leq 1$. The interdicted weight of e' is $c_{e'} + d_{e'}$. Since $d_{e'}$ is at least $\max_{e \in \mathcal{E}} c_e + 1 - c_{e'}$, $c_{e''} < c_{e'} + d_{e'}$. Therefore, the weight of the tree defined by the edge set $T_1 \cup T_2 \cup \{e''\}$ is lower than the one of T which is a contradiction because the MST with respect to $\hat{\mathbf{x}}$ is T, i.e. the tree defined by $\hat{\mathbf{y}}$. Therefore, the optimal solution of [MSTI$_{\text{Opt}}$] is feasible in [MSTI$_{\text{Tot}}$]. Then, [MSTI$_{\text{Opt}}$] and [MSTI$_{\text{Tot}}$] provide the same optimal solution. □

3.6 Parameter Analyses and Illustrative Problem

We consider different setups for experimentation in Section 3.9. These refer to the use of total and partial interdiction as well as considering a cardinality or a knapsack constraint. Let ρ be the connectivity number of $\mathcal{G}$ with respect to interdiction costs b_e. Table 3.1 shows the possible settings of an MSTI problem instance.

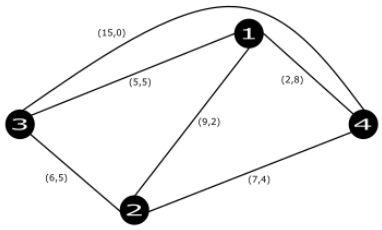
Constraint \ Interdiction	Partial $d_e < \infty, \forall e \in \mathcal{E}$		Total $d_e = \infty, \forall e \in \mathcal{E}$	
	$B < \rho$	$B \geq \rho$	$B < \rho$	$B \geq \rho$
Cardinality	✓	•	✓	✠
Knapsack	✓	•	✓	✠

Table 3.1: Instance cases

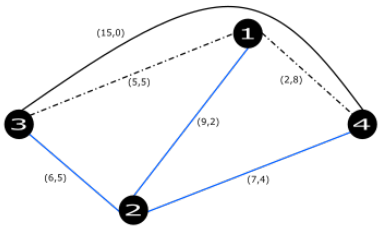
Cases marked with a check-mark (✓) are defined as non-trivial instances. On the other hand, cases marked with a cross (✠) are discarded since they represent the scenarios where it is possible for the interdictor to disconnect $\mathcal{G}$. For the cases marked with black dots (•), it has to be verified that the edges associated with interdiction augmentations equal to infinity do not define a cut in $\mathcal{G}$. The value of ρ is defined as the sum of the interdiction costs of the edges that define the minimum cut of $\mathcal{G}$, i.e. the set of edges when removed disconnects the network. However, a "real" cut might not exist for partial interdiction. In the case of partial interdiction, the augmentation of an edge may or may not be large enough to consider the interdiction as edge removal. Therefore, it is verified that if a budget larger than ρ is defined then the edges that do have a large enough augmentation do not define a cut in the graph.

An illustrative example is presented to show the main differences of the instance settings. We first consider partial interdiction and a cardinality interdiction constraint. Let $\mathcal{G}$ be a complete graph with $|\mathcal{V}| = 4$, $|\mathcal{E}| = 6$, $\rho = 3$ and weights and interdiction augmentations as shown in Figure 3.2a. Semi-dotted lines represent partially interdicted edges, dotted lines represent removed edges and blue lines represent the MST with respect to interdiction. The optimal solution for $B = 2$ is presented in Figure 3.2b. No interdicted edges were selected in the MST because the increase in the edge weight is sufficiently large to be considered as eliminated during the computation of the MST. This is not always

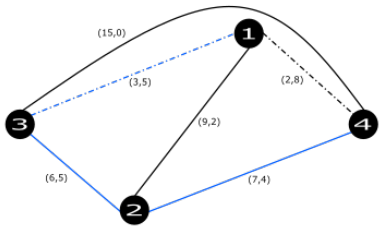
the case. Consider a slightly different network where $c_{13} = 3$ then solve for $B = 2$. The optimal solution results as illustrated in Figure 3.2c. In this example, the weight of one of the interdicted edges, that is the edge $\{1, 3\}$ with a weight of $c_{13} + d_{13} = 8$, is still small enough to be chosen as part of the MST instead of the non-interdicted edges $\{1, 2\}$ and $\{3, 4\}$. The question of whether interdicted edges are selected for the MST depends on the weights and augmentations of the network. Moreover, the larger the budget B, the more likely the network operator will pick interdicted edges.



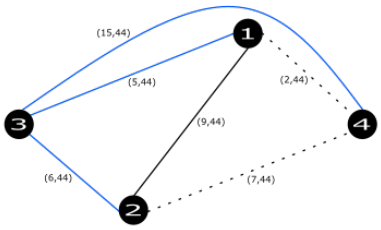
(a) Initial graph



(b) Partial interdiction with $B = 2$, Case 1



(c) Partial interdiction with $B = 2$, Case 2



(d) Total interdiction with $B = 2$

Figure 3.2: Illustrative example of $G = (4, 6)$

Using interdicted edges as part of the MST is not an issue when considering partial interdiction. However, this is not the case for total interdiction. First, if the interdictor has at her disposal a large budget that exceeds the value of ρ, she can immediately disconnect the graph which is a trivial case and is not considered throughout this manuscript. On the opposite case, we revise a small example when $B < \rho$ and the value of d_e define a large enough value to model edge removal. Consider the problem defined by Figure 3.2a but with d_e defining large values as in Section 3.5. The solution for $B = 2$ of this instance is presented in Figure 3.2d. As expected for total interdiction, interdicted edges are not part

of the MST. The interdictor prohibits all edges incident to node 4 except the one with the highest weight, thus the follower has no choice but to use edge $\{3, 4\}$ and thereby inducing a large increase in the weight of the tree. This was not the case in Figure 3.2b. These two examples show the drastic difference in the solutions of the MSTI problem defined on the same graph when partial and total interdiction are considered.

Since the cardinality constraint is a relaxation of the knapsack constraint, an instance that is "valid" for the cardinality constraint is thus "valid" for the knapsack constraint.

3.7 Valid Inequalities

This section presents valid inequalities to strengthen the linear relaxation of formulation [MSTI_{Opt}]. Two sets of inequalities are developed. The first makes use of the linearization technique used to linearize the objective function (3.5a) of [MSTI_{Opt}]. This is done in order to find tighter optimal conditions. The second set of inequalities modifies optimality constraints (3.6f)-(3.6i) to remove variables y_e, then we use the definition of t_e to further simplify the constraints.

3.7.0.1 First Set of Valid Inequalities

General conditions are stated next. Let the sum $z_{uji} + z_{vij}$ be equal to w_{ij}^{uv}. To ease notation we simply use w instead of w_{ij}^{uv}. First, note that if $y_e = 0$, $e = \{i, j\}$, then due to (3.6c) w is zero. Second, if $y_e = 1$, then w may be 1 or 2. We recall that $z_{uji} = 1$ means that in the optimal $\mathbf{z}_u$-tree there exists a path that uses the arc (j, i) and with anti-root in u. Equivalently z_{vij} means that in the optimal $\mathbf{z}_v$-tree a path that uses arc (i, j) and is ant-rooted in v exists. This is possible only when $y_f = 0$, otherwise edge $\{u, v\}$ would be selected and a cycle would be formed. Consequently when $y_f = 1$, then w cannot be 2.

We consider the following expression $w = y_e + (1 - y_f)(\Delta)$ where Δ represents a term which is set according to sets $\mathcal{E}_i$, $i = 2, 3, 4, 5$, introduced in Section 3.3 in order to cut off non-optimal integer solutions. Therefore, if $y_e = 1$ and for each set $\mathcal{E}_i$, the goal is to determine Δ terms as a expression of the interdiction variables x_e and x_f in order to make $w = 2$ for the cases when Theorem 3.4 is not

imposed. This is, an expression of the interdictions of e and f when optimality constraints (3.6f)-(3.6i) are relaxed.

Case $\mathcal{E}_2$. The expression $w = y_e + (1 - y_f)(1 - x_e)x_f$ is defined since $w = 2$ when $x_e = 0$ and $x_f = 1$. We develop de right-hand side and $w = y_e + x_f - x_e x_f - y_f x_f + y_f x_e x_f$. Note that $-x_e x_f + y_f x_e x_f = x_e x_f (y_f - 1) \leq 0$. Thus, $w \leq y_e + x_f - y_f x_f$. We replace $y_f x_f$ by t_f and $w \leq y_e + x_f - t_f$ is obtained.

Case $\mathcal{E}_3$. Expression w is equal to 2 only when edge f is interdicted. Therefore, $\Delta = x_f$ and $w = y_e + (1 - y_f)x_f$. A simple development of w and the use of expression $t_f = y_f x_f$ yields to $w = y_e + x_f - t_f$.

Case $\mathcal{E}_4$. In this case, we define $w = y_e + (1 - y_f)(1 - x_e)$. We have that $w = 2$ when edge e is not interdicted. Then w can be expressed as $y_e - x_e + 1 - y_f(1 - x_e)$. If we consider $x_e = 0$, $w \leq y_e - y_f + 1$ is obtained. The latter can be done since the interdiction of f does not affect the expression.

Case $\mathcal{E}_5$. We define $w = y_e + (1 - y_f)[(1 - x_e) + x_e x_f]$; all possible interdiction cases of e and f except $x_e = 1$ and $x_f = 0$ make $w = 2$. After computation, it is obtained $w = y_e - x_e + x_e x_f - y_f + x_e y_f - y_f x_e x_f + 1$. We observe that $x_e x_f \leq x_e + x_f$ and $x_e(y_f - t_f) \leq 0$. Therefore, the resulting expression is $w \leq y_e - y_f + x_f + 1$.

The valid inequalities are summarized below:

$$z_{uji} + z_{vij} \leq y_e + x_f - t_f \qquad \forall (e, f) \in \mathcal{E}_2, \mathcal{E}_3 \tag{3.11a}$$

$$z_{uji} + z_{vij} \leq y_e - y_f + 1 \qquad \forall (e, f) \in \mathcal{E}_4 \tag{3.11b}$$

$$z_{uji} + z_{vij} \leq y_e - y_f + x_f + 1 \qquad \forall (e, f) \in \mathcal{E}_5 \tag{3.11c}$$

The family of constraints (3.11a) can be added to [MSTI_{Opt}]. However, (3.11a) replaces (3.6g) due to the next Lemma.

Lemma 3.6. *Let $e = \{i, j\}$ and $f = \{u, v\}$ two edges either in $\mathcal{E}_2$ or $\mathcal{E}_3$. Then, constraints (3.6g) are dominated by (3.11a).*

Proof. Let e and f be in either $\mathcal{E}_2$ or $\mathcal{E}_3$. By (3.6g) we have $z_{uji} + z_{vij} \leq y_e + x_f$. Since y_f and x_f are binary, variables $t_f \geq 0$. Then $y_e + x_f - t_f \leq y_e + x_f$ and constraints (3.11a) dominates (3.6g). □

3.7.0.2 Second Set of Valid Inequalities

The second set of inequalities is obtained by adding up z_{uij} in both sides of constraints (3.6f). We can rearrange and group the terms in order to obtain

$$(z_{uij} + z_{uji}) + z_{vij} - z_{uij} \leq y_e \qquad \forall (e, f) \in \mathcal{E}_1$$

Then, by using constraint (3.3c) we obtain the following expression

$$y_e + z_{vij} - z_{uij} \leq y_e \qquad \forall (e, f) \in \mathcal{E}_1$$

Finally, we simplify and obtain $z_{vij} - z_{uij} \leq 0$. The same process is performed on constraints (3.6g)-(3.6i). The resulting constraints are shown next:

$$z_{vij} - z_{uij} \leq 0 \qquad \forall (e, f) \in \mathcal{E}_1 \tag{3.14a}$$
$$z_{vij} - z_{uij} \leq x_f \qquad \forall (e, f) \in \mathcal{E}_2, \forall (e, f) \in \mathcal{E}_3 \tag{3.14b}$$
$$z_{vij} - z_{uij} \leq 1 - x_e \qquad \forall (e, f) \in \mathcal{E}_2, \forall (e, f) \in \mathcal{E}_4 \tag{3.14c}$$
$$z_{vij} - z_{uij} \leq 1 - x_e + x_f \qquad \forall (e, f) \in \mathcal{E}_5 \tag{3.14d}$$

Note that we can multiply the right hand side of constraints (3.14a)-(3.14d) by y_e and the inequalities that we obtain are still valid. Indeed, when $y_e = 0$ variables z_{vij} and z_{uij} are zero due to constraint (3.6c). When $y_e = 1$ the equivalence of the expressions is trivial.

For this reason, it is possible to multiply the right hand side of constraints (3.14c) and (3.14d) by y_e in order to obtain

$$z_{vij} - z_{uij} \leq (1 - x_e)y_e \qquad \forall (e, f) \in \mathcal{E}_2, \forall (e, f) \in \mathcal{E}_4 \tag{3.15a}$$
$$z_{vij} - z_{uij} \leq (1 - x_e + x_f)y_e \qquad \forall (e, f) \in \mathcal{E}_5 \tag{3.15b}$$

Since $t_e = x_e y_e$ and $x_f \geq y_e x_f$, we obtain the next constraints:

$$z_{vij} - z_{uij} \leq y_e - t_e \qquad \forall (e,f) \in \mathcal{E}_2, \forall (e,f) \in \mathcal{E}_4 \tag{3.16a}$$

$$z_{vij} - z_{uij} \leq y_e + x_f - t_e \qquad \forall (e,f) \in \mathcal{E}_5 \tag{3.16b}$$

3.8 Extension of the $\bar{B}$-edge Connected Certificate

The reduction of $\mathcal{G}$ to $\mathcal{G}_{\bar{B}}$ in Section 2.1 is only proved for the case of total interdiction, i.e. interdicted edges are to be removed. In this section, the $\bar{B}$-edge connected certificate is extended to consider partial interdiction. Let $\mathcal{G}(\mathcal{S})$ be an augmented graph of $\mathcal{G}$ where the weight of the edges in $\mathcal{S}$ is $c_e + d_e$ and let $\bar{B} > 1$, $\bar{B} \in \mathbb{Z}^+$.

Lemma 3.7. *For any edge set $\mathcal{S}$, $\mathcal{S} \subseteq \mathcal{E}$, $|\mathcal{S}| < \bar{B}$ and $\sum_{e \in S} b_e \leq B$, the edges of minimum spanning tree of $\mathcal{G}(\mathcal{S})$ belong to $\mathcal{G}_{\bar{B}}$.*

Proof. Let $\mathcal{S} = \mathcal{S}_r \cup \mathcal{S}_a$. S_r is an edge set whose edges have an interdiction augmentation of infinity, i.e. these edges are considered as removed when interdicted. S_a is an edge set whose elements are to be augmented by a finite amount. Clearly, $\mathcal{S} \subseteq \mathcal{E}$; assume that $|\mathcal{S}| < \bar{B}$ and $\sum_{e \in S} b_e \leq B$.

If $\mathcal{S}_a = \emptyset$ then all edges of $\mathcal{S}$ are considered as removed when interdicted and the proof is reduced to that of Lemma 2.1.

Assume $\mathcal{S}_a \neq \emptyset$. The proof is developed by contradiction. Suppose there is an edge $e^* = (u,v) \in \mathcal{E} \setminus \cup_{i=0}^{\bar{B}} T_i$ in the MST T^* of $\mathcal{G}(\mathcal{S})$. If edge e^* is removed from T^*, then the connected components $\hat{T}_1$ and $\hat{T}_2$ are obtained. Assume that $\hat{T}_1$ and $\hat{T}_2$ contain node u and node v, respectively. In each T_i, $i = 0, ..., \bar{B}$, there exists a unique path between u and v. The weight of e^* is larger than the weight of any edge of any u-v path of T_i, $i = 0, ..., \bar{B}$, otherwise e^* would have been chosen for one of the trees. The budget B is not large enough to let $\mathcal{S}_r$ define a cut of $\mathcal{G}$. In addition, $|\mathcal{S}| < \bar{B}$. Therefore, there must exists a non-interdicted edge in one of the u-v paths that connects $\hat{T}_1$ and $\hat{T}_2$ that has a weight lower than the weight of e^*. Therefore, a tree of $\mathcal{G}(\mathcal{S})$ with a smaller weight than T^* is found. Hence, a contradiction. □

3.9 Computational Experiments

3.9.1 Instance Specifications

Instances of different sizes are generated to represent all cases shown in Table 3.1. The instance generation is partly based on that in Bazgan *et al.* (2012). Networks considered are undirected complete graphs of $|\mathcal{V}|$ nodes. The values of c_e, $e \in \mathcal{E}$, are uniformly generated in $[1, 100]$. An arbitrary total order is imposed for edges of the same weight without loss of generality. When total interdiction is considered, $d_e = \max_{f \in \mathcal{E}} c_f - c_e + 1$ for all $e \in \mathcal{E}$. In the partial interdiction case, d_e is randomly generated between 1 and $\max_{f \in \mathcal{E}} c_f - c_e + 1$. When the interdiction budget constraint is formulated as a cardinality constraint, $b_e = 1$ for all $e \in \mathcal{E}$. When it is formulated as a knapsack constraint, b_e is randomly generated between 1 and 100.

In Section 3.8, it is shown that the size of the network can be reduced by using the $\bar{B}$-edge connected certificate of $\mathcal{G}$. This preprocessing is implemented in our experimentation. The preprocessing of the network depends on knowing in advance the maximum number of edges to interdict. In the cardinality case, $\bar{B}$ is equal to the budget B. Clearly, it is possible to know the maximum number of edges that the interdictor can interdict when considering a knapsack constraint by setting $\bar{B} = \max\{\sum_{e \in \mathcal{E}} x_e : \sum_{e \in \mathcal{E}} b_e x_e \leq B\}$. However, preliminary experimentation proved that $\bar{B}$ turns out to be large enough such that the network reduction ends up providing the original network. Therefore, the preprocessing is only meaningful when considering a cardinality budget constraint.

The formulations are implemented in a 64-bit computer with an Intel Xeon(R) CPU E5-2630 v4 (2.20GHz) processor and 64 GB of RAM and solved using CPLEX 12.9. A time limit of 3600 seconds is imposed for each experiment.

3.9.2 Tuning Phase for Valid Inequalities

Preliminary tuning experiments are carried out to verify the impact of the valid inequalities developed in Section 3.7. Let $[V_0]$ be the $[MSTI_{Opt}]$ to which Lemma 3.6 is applied, i.e. constraints (3.6g) are replaced by (3.11a). $[V_1]$ represent model $[V_0]$ with constraints (3.11b), $[V_2]$ with constraints (3.11c) and $[V_3]$ with both of them. The $[MSTI_{Opt}]$ and the latter four formulations are used to solve the MSTI problem on complete graphs of different sizes and budgets. Total interdiction and

$\lvert\mathcal{V}\rvert$	B	[MSTI_{Opt}]	[V_0]	[V_1]	[V_2]	[V_3]
10	5	710.15	492.26	492.26	492.26	492.26
10	5	681.16	459.47	459.47	459.47	459.47
15	3	218.84	218.84	218.84	217.96	217.96
20	3	232.81	232.81	232.81	232.81	232.81

Table 3.2: Preliminary results of the first set of valid inequalities

cardinality constraint are assumed. The computational results are presented in Table 3.2.

The first two columns represent the number of nodes and the budget considered for each instance. Each row represents a different complete graph. Table 3.2 reports the LP relaxation of each of the five models. The values shown in column [MSTI_{Opt}] are upper bounds on the LP relaxations of formulations [V_i], $i = 0, 1, 2, 3$, since the latter models are [MSTI_{Opt}] with additional constraints. [V_0] and [V_1] provide tighter values for the first two instances and exactly the same values for the LP relaxation as [MSTI_{Opt}]. We see that both [V_2] and [V_3] slightly improves the LP relaxation of the third instance. The latter implies that constraint (3.11b) is inactive since the definition of these valid inequalities in [V_1] and [V_3] do not improve the bounds. On the other hand, even if it was for less than a unit, (3.11c) improved the LP value on the instance of 15 nodes. Therefore, only [V_2], considering constraints (3.11a) and (3.11c), is considered in the final experimentation. This formulation is renamed as [MSTI-V_2].

Analogously, experimentation of all possible combinations of the second set of valid inequalities is done. Table 3.3 presents the names representing the different implementations of each of the constraints.

Name	Constraints	Name	Constraints	Name	Constraints
[V_4]	(3.14a)	[V_9]	(3.14a), (3.16a)	[V_{14}]	(3.14a), (3.14b), (3.16a)
[V_5]	(3.14b)	[V_{10}]	(3.14a), (3.16b)	[V_{15}]	(3.14a), (3.14b), (3.16b)
[V_6]	(3.16a)	[V_{11}]	(3.14b), (3.16a)	[V_{16}]	(3.14a), (3.16a), (3.16b)
[V_7]	(3.16b)	[V_{12}]	(3.14b), (3.16b)	[V_{17}]	(3.14b), (3.16a), (3.16b)
[V_8]	(3.14a), (3.14b)	[V_{13}]	(3.16a), (3.16b)	[V_{18}]	(3.14a), (3.14b), (3.16a), (3.16b)

Table 3.3: Names of formulations with valid inequalities.

Table 3.4 shows the preliminary results of the second set of valid inequalities assuming total interdiction and cardinality constraint. Our first observation is that constraints (3.14a) and (3.16a) are inactive during optimization and do not

improve the LP relaxation of [MSTI_{Opt}]. This is confirmed in the results of columns [V_4], [V_6] and [V_9] thus we can discard all formulations including any of those constraints. For the first two instances, [V_7] and [V_{12}] provide the tightest values. The common family of valid inequalities in the last two models is (3.16b). For the instances of sizes 15 and 20, the best LP relaxation values are obtained by [V_5] and [V_{12}]. In this case, the common constraints is (3.14b). Therefore, [V_{12}] is the tighter formulation. For the final experimentation, only the best performing formulation, i.e. [V_{12}] considering (3.14b) and (3.16b), is analyzed and compared. [V_{12}] is renamed as [MSTI-V_{12}].

$\lvert\mathcal{V}\rvert$	B	[MSTI_{Opt}]	[V_4]	[V_5]	[V_6]	[V_7]	[V_8]	[V_9]	[V_{10}]	[V_{11}]	[V_{12}]	[V_{13}]	[V_{14}]	[V_{15}]	[V_{16}]	[V_{17}]	[V_{18}]
10	5	710.15	710.15	709.98	710.15	688.93	709.98	710.15	688.93	709.98	688.93	688.93	709.98	688.94	688.93	688.94	688.94
10	5	681.16	681.16	681.44	414.89	660.24	681.44	681.16	660.24	681.44	660.24	660.24	681.44	660.24	660.24	660.24	660.24
15	3	218.84	218.84	202.93	218.84	216.96	202.93	218.84	216.96	202.93	202.93	216.96	202.93	202.93	216.96	202.93	202.93
20	3	232.81	232.81	228.69	232.81	232.81	228.69	232.81	232.81	228.69	228.62	232.81	228.69	228.62	232.81	228.62	228.62

Table 3.4: Preliminary results of the second set of valid inequalities

3.9.3 Final Experimentation

Computational results on the performance of the different mathematical formulations presented in this manuscript are reported in this section. The formulations assessed in this chapter are: the formulation retrieved from the literature [MSTI_{DBa}]; our original model [MSTI_{Opt}] based on the MST formulation of Martin (1991); two versions of [MSTI_{Opt}], [MSTI-V_2] and [MSTI-V_{12}], strengthened by two sets of valid inequalities; the dual version [$\text{MSTI}_{\text{Dual}}$] of [$\text{MSTI}_{\text{Opt}}$] and a more compact version [$\text{MSTI}_{\text{SDual}}$] of [$\text{MSTI}_{\text{Dual}}$] by removing one family of constraints of the MST inner problem. Networks of size $|\mathcal{V}| = 10, 15, 20, 25, 30$ and 50 are considered. For a cardinality constraint, $B = 3, 5$ and 7. In the case of a knapsack constraint, the budget is defined as a proportion of the largest edge weight of the graph. Thus, $B = \eta \cdot \max_{e \in \mathcal{E}} c_e$ where $\eta = 0.2, 0.4, 0.6, 0.8, 1$ and 1.2.

For each instance in the following tables, the average computational time, the average number of Branch-and-Bound nodes that are explored, the average relative gap at root node, the average relative gap between the best known solution and the upper bound at the end of optimization of four different networks is reported. The last column of each set of data represents the number of solved instances out of the four test networks. A dash (-) is reported when the instance

could not be solved or no integer solution could be found to compute a optimality gap under the time limit. Assuming a cardinality constraint, Tables 3.5 and 3.6 detail the computational results when total and partial interdiction are considered, respectively.

When considering total interdiction, instances of 10 nodes are efficiently solved by both [MSTI_{DBa}] and [$\text{MSTI}_{\text{SDual}}$]. However, [$\text{MSTI}_{\text{DBa}}$] solves the instances where $B = 5$ at root node. On the other hand, [$\text{MSTI}_{\text{SDual}}$] is, in average, the best formulation in terms of computation time, size of the Branch-and-Bound tree and relative gap at root node when $B = 7$. Likewise, instances of 15 nodes are solved in a fairly short time by [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$]. [$\text{MSTI}_{\text{SDual}}$] obtains the shorter computation times in this case with the exception of instances of $B = 7$ where it is outperformed by [$\text{MSTI}_{\text{Dual}}$] only by an average 0.3 of seconds. Even though [MSTI_{Tot}] has the tighter relative gaps at root node and the less number of nodes explored in the Branch-and-Bound tree, it obtains much larger computation times. A similar situation occurs for [MSTI_{Opt}]. Moreover, [MSTI_{Opt}] and [MSTI_{Tot}] are unable to solve instances with the largest budget in graphs of size 20. However, [MSTI_{Tot}] still obtains better relative gaps at the root node when the budget is set to 5 and 7. [MSTI-V_2] and [MSTI-V_{12}] are no better than [MSTI_{Opt}], in several instances [MSTI-V_2] obtains a tighter relative gap at the root node but the computational time is practically doubled. In the case of [MSTI-V_{12}], greater relative gaps and computational times are obtained and even small size instances are unable to be solved. [$\text{MSTI}_{\text{SDual}}$] still provides the fastest computational times yet the tightest gaps belong to [MSTI_{Tot}] and [MSTI_{Opt}]. The limits of the formulations start on instances of 25 nodes where only one instance of budget $B = 7$ is solved by [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$]. A pattern similar to the other instance sizes is observed when $B = 3$ and 5. Instances of 30 nodes with a budget of 3 are solved by all models with [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] as the most efficient. Only [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] are able to solve three out of four instances when $B = 5$. The limitations of our models are reached at instances of 50 nodes and a budget of 7.

Models [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] provide the best results when considering partial interdiction as shown in Table 3.6. All graphs of size 10 are solved in less than a second by all formulations. [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] out-stand by

solving all them at root node. Then, a similar pattern is observed as in the experiments where total interdiction is imposed, i.e. [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] excel in computational time even though smaller relative gaps and fewer number of explored nodes are occasionally reported by the other formulations. Nevertheless, all formulations are unable to solve instances of 50 nodes.

The results obtained by considering a knapsack interdiction constraint are shown in Tables 3.7 and 3.8. The budget for each graph is reported as η, the fraction of the largest edge weight for simplification. We analyze first the statistics when total interdiction and a knapsack constraint are assumed. Unlike results of those of cardinality constraint, all test instances are solved by at least formulation [MSTI_{Tot}]. For instances of 10, 15 and 20 nodes and $\eta \leq 1$, the results are practically identical for any of the models except for [MSTI-V_{12}]. They differentiate in the computational time, although the largest difference in time between the models is no more than five seconds. For these same sizes, the computational time slightly increases when $\eta = 1.2$. Once more, similar results are obtained in instances of 25 and 30 nodes. However, instances where $\eta \geq 0.6$ are not solved at root node. For [MSTI_{Opt}], [MSTI-V_2] and [MSTI_{Tot}] the gap at root node is of less than 1.17 where $\eta \leq 0.8$. [MSTI_{Tot}] gives in average the tightest gaps but it is sometimes outperformed by [MSTI-V_{12}]. [MSTI-V_{12}] behaves differently than its counterparts, most of the instances are not solved at root node nevertheless [MSTI-V_{12}] occasionally reports better average computational times than [MSTI_{Opt}]; moreover [MSTI-V_{12}] solved all of the test instances. For the largest instance sizes, the statistics decay significantly for [MSTI_{DBa}], [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$]. In the case of [$\text{MSTI}_{\text{DBa}}$], no instance is solved when η is greater than or equal to 0.6 and greater than or equal to 0.8 for [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$].

In Table 3.8, [$\text{MSTI}_{\text{SDual}}$] is the only model capable of solving all of the instances at root node for networks of 10, 15 and 20 nodes when considering partial interdiction. When the size of the network increases to 50, [MSTI_{DBa}], [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] deteriorate. [$\text{MSTI}_{\text{DBa}}$] is unable to solve to optimality in the allocated time any of the largest instances. [$\text{MSTI}_{\text{SDual}}$] solves most of the instances with $\eta \leq 0.8$ but at a high computational time. [MSTI_{Opt}] and [MSTI-V_2] provide very similar results but [MSTI_{Opt}] out-stands in computational time. Even though [MSTI_{Opt}] have tightest relative gaps at root node than [MSTI-V_{12}], the latter reports shorter resolution times in instances of 25 nodes or more, however

the difference is not larger than several seconds. Furthermore, [MSTI-V_{12}] is able to solve a few of the largest instances.

Table 3.5: Computational results considering total interdiction and cardinality constraint.

Config.		Total Interdiction																			
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot
		[MSTI_{Opt}]					[MSTI-V_2]					[MSTI-V_{12}]					[MSTI_{DBa}]				
10	3	0.8	0.0	0.0	0.00	4/4	1.3	0.0	0.00	0.00	4/4	1.3	68.0	77.42	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	5	10.6	2517.0	161.5	0.00	4/4	15.4	1827.0	124.13	0.00	4/4	20.8	2336.5	318.92	0.00	4/4	0.4	0.0	0.00	0.00	4/4
	7	31.3	9722.0	190.6	0.00	4/4	57.4	11475.5	141.10	0.00	4/4	57.7	9103.8	325.27	0.00	4/4	1.0	1983.0	25.01	0.00	4/4
15	3	5.5	57.3	13.7	0.00	4/4	9.9	45.5	12.26	0.00	4/4	8.1	167.8	51.69	0.00	4/4	0.9	106.8	9.64	0.00	4/4
	5	94.6	2113.3	58.5	0.00	4/4	164.1	1521.0	54.55	0.00	4/4	131.2	2033.8	166.17	0.00	4/4	14.9	9315.8	68.28	0.00	4/4
	7	1903.6	11188.3	193.7	0.00	4/4	3404.7	12177.8	176.32	7.32	2/4	3600.0	8805.5	397.42	91.32	0/4	102.5	53304.8	99.68	0.00	4/4
20	3	24.7	97.0	12.7	0.00	4/4	49.8	111.8	14.46	0.00	4/4	46.6	484.8	51.12	0.00	4/4	4.3	655.5	26.69	0.00	4/4
	5	1379.2	4701.3	50.1	0.00	4/4	2468.6	4256.0	62.66	2.21	3/4	3600.0	5189.3	137.99	28.90	0/4	153.4	25790.8	71.17	0.00	4/4
	7	3600.0	1576.0	116.9	63.58	0/4	3600.0	294.5	98.37	77.92	0/4	3600.0	707.0	234.99	179.26	0/4	1468.3	186060.3	95.94	0.00	4/4
25	3	108.7	187.8	12.1	0.00	4/4	222.4	164.0	11.84	0.00	4/4	331.5	908.3	45.54	0.00	4/4	19.2	1445.5	26.51	0.00	4/4
	5	3032.0	1826.8	59.0	17.16	2/4	3506.0	529.5	43.80	26.19	1/4	3600.0	489.5	106.08	71.41	0/4	575.9	36222.8	60.36	0.00	4/4
	7	3600.0	37.3	113.3	104.09	0/4	3600.0	0.0	128.41	128.41	0/4	3600.0	72.5	185.92	170.42	0/4	3600.0	97829.0	84.33	20.20	0/4
30	3	896.4	911.3	18.50	0.00	4/4	2019.9	1016.5	24.20	0.00	4/4	3115.5	1871.5	54.12	8.37	2/4	48.0	2701.8	26.19	0.00	4/4
	5	3600.0	78.3	56.08	49.56	0/4	3600.0	0.0	79.00	79.00	0/4	3600.0	86.3	144.29	110.24	0/4	2502.1	54591.5	63.78	10.37	2/4
	7	3600.0	0.0	177.42	177.42	0/4	3600.0	0.0	171.56	171.56	0/4	3600.0	7.0	275.38	272.31	0/4	3600.0	33495.0	92.29	43.44	0/4
50	3	3600.0	0.0	26.62	26.62	0/4	3600.0	-	-	-	0/4	3600.0	-	-	-	0/4	2068.2	7661.8	20.95	5.32	2/4
		[$\text{MSTI}_{\text{Dual}}$]					[$\text{MSTI}_{\text{SDual}}$]					[MSTI_{Tot}]									
10	3	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4					
	5	0.3	0.0	0.00	0.00	4/4	0.4	279.0	11.57	0.00	4/4	2.7	55.8	18.03	0.00	4/4					
	7	1.0	1981.3	27.91	0.00	4/4	0.7	1716.3	17.41	0.00	4/4	3.2	37.5	14.34	0.00	4/4					
15	3	0.7	113.0	1.70	0.00	4/4	0.5	0.0	0.00	0.00	4/4	3.3	61.5	10.80	0.00	4/4					
	5	7.6	9031.0	57.46	0.00	4/4	6.9	8834.8	53.38	0.00	4/4	45.6	2371.5	47.74	0.00	4/4					
	7	56.8	54769.0	90.65	0.00	4/4	57.1	56743.8	95.52	0.00	4/4	603.0	9792.5	90.18	0.00	4/4					
20	3	1.5	0.0	0.00	0.00	4/4	1.4	0.0	0.00	0.00	4/4	13.6	72.3	10.98	0.00	4/4					
	5	71.9	23876.8	65.44	0.00	4/4	61.2	21990.3	62.42	0.00	4/4	507.9	3794.8	50.37	0.00	4/4					
	7	599.1	190184.5	90.99	0.00	4/4	579.0	185709.3	95.92	0.00	4/4	3600.0	6434.8	81.32	29.06	0/4					
25	3	10.0	1318.0	28.13	0.00	4/4	10.3	1490.8	29.07	0.00	4/4	55.7	212.3	10.32	0.00	4/4					
	5	328.7	33670.5	57.76	0.00	4/4	293.1	36135.8	61.67	0.00	4/4	1824.6	3311.3	45.24	2.74	3/4					
	7	3148.2	178417.0	84.04	15.88	1/4	3143.8	173132.0	88.22	16.70	1/4	3600.0	945.0	113.76	57.97	0/4					
30	3	42.8	2668.5	29.69	0.00	4/4	43.8	2571.0	28.93	0.00	4/4	406.9	876.8	14.70	0.00	4/4					
	5	2191.5	86170.3	64.67	2.70	3/4	2308.2	90281.5	63.83	2.69	3/4	3600.0	898.25	80.03	41.65	0/4					
	7	3600.0	58510.5	96.57	40.23	0/4	3600.0	57981.0	96.44	40.18	0/4	3600.0	29.3	138.52	131.72	0/4					
50	3	1951.5	8408.0	20.49	2.70	3/4	1995.6	9803.8	19.35	2.66	3/4	3600.0	44.3	22.30	21.43	0/4					

Table 3.6: Computational results considering partial interdiction and cardinality constraint.

Config.		Partial Interdiction														
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot
		[MSTI_{Opt}]					[MSTI-V_2]					[MSTI-V_{12}]				
10	3	0.2	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.4	44.0	30.56	0.00	4/4
	5	0.3	9.0	1.71	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.6	179.0	57.35	0.00	4/4
	7	0.3	0.0	0.00	0.00	4/4	0.4	0.0	0.00	0.00	4/4	0.6	254.0	54.44	0.00	4/4
15	3	3.1	79.5	9.60	0.00	4/4	4.1	64.3	7.81	0.00	4/4	3.6	169.8	39.06	0.00	4/4
	5	7.1	257.5	18.44	0.00	4/4	8.2	228.5	16.55	0.00	4/4	12.7	1167.0	79.70	0.00	4/4
	7	11.2	984.0	24.44	0.00	4/4	11.7	402.8	19.61	0.00	4/4	35.0	4718.3	99.37	0.00	4/4
20	3	6.5	35.8	6.97	0.00	4/4	8.6	31.3	4.50	0.00	4/4	10.9	282.0	28.02	0.00	4/4
	5	27.1	400.0	14.87	0.00	4/4	35.6	281.8	12.81	0.00	4/4	126.1	4407.3	63.23	0.00	4/4
	7	58.5	1416.5	25.83	0.00	4/4	71.6	909.8	25.56	0.00	4/4	423.8	8267.3	106.88	0.00	4/4
25	3	66.0	232.3	9.07	0.00	4/4	85.9	129.5	7.49	0.00	4/4	113.2	665.5	43.88	0.00	4/4
	5	1139.2	2262.0	21.05	1.95	3/4	1151.5	1087.3	20.62	1.79	3/4	2638.8	4751.3	98.70	9.33	3/4
	7	1871.8	4762.0	41.34	6.26	3/4	2072.7	3158.3	38.20	5.06	3/4	3600.0	4154.5	123.54	36.46	0/4
30	3	428.1	470.5	13.40	0.00	4/4	516.3	344.8	11.69	0.00	4/4	1583.1	2245.0	40.92	0.00	4/4
	5	3374.8	1344.5	49.30	14.90	1/4	3427.6	826.3	36.80	16.90	1/4	3600.0	390.0	104.66	56.87	0/4
	7	3600.0	537.8	61.12	36.87	0/4	3600.0	352.3	61.27	38.92	0/4	3600.0	279.5	141.56	83.76	0/4
50	3	3600.0	0.0	21.95	21.95	0/4	3600.0	-	-	-	0/4	3600.0	16.0	28.07	26.61	0/4
		[MSTI_{DBa}]					[$\text{MSTI}_{\text{Dual}}$]					[$\text{MSTI}_{\text{SDual}}$]				
10	3	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	5	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	7	0.3	39.3	2.11	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
15	3	0.7	0.0	0.00	0.00	4/4	0.8	25.8	3.43	0.00	4/4	0.5	0.0	0.00	0.00	4/4
	5	3.9	2102.3	34.96	0.00	4/4	1.9	1378.8	11.45	0.00	4/4	2.1	1666.0	13.06	0.00	4/4
	7	17.2	7612.8	45.03	0.00	4/4	9.4	6909.8	19.43	0.00	4/4	7.4	6390.5	16.34	0.00	4/4
20	3	4.0	578.8	23.21	0.00	4/4	1.5	0.0	0.00	0.00	4/4	1.6	191.3	4.02	0.00	4/4
	5	31.0	6137.3	38.88	0.00	4/4	11.7	4324.0	38.28	0.00	4/4	13.1	4093.5	35.74	0.00	4/4
	7	119.3	16313.3	40.83	0.00	4/4	70.1	16715.0	42.94	0.00	4/4	62.2	15839.8	39.72	0.00	0/4
25	3	19.6	1746.0	33.57	0.00	4/4	13.8	1617.3	37.44	0.00	4/4	16.5	1692.5	35.42	0.00	4/4
	5	429.4	27600.3	45.40	0.00	4/4	248.9	24697.3	44.76	0.00	4/4	244.8	24175.8	44.97	0.00	3/4
	7	2298.0	75185.8	48.33	3.09	3/4	1749.0	105014.3	53.17	1.03	1/4	1657.6	110006.8	52.54	0.00	0/4
30	3	58.5	3440.8	46.65	0.00	4/4	47.5	3754.8	47.44	0.00	4/4	44.6	3509.0	47.96	0.00	4/4
	5	2221.5	63990.8	62.09	0.00	4/4	1487.1	59806.0	62.49	0.00	3/4	1610.2	62609.5	61.52	0.00	0/4
	7	3600.0	37857.5	70.68	24.58	0/4	3600.0	49736.3	68.69	23.40	0/4	3600.0	57695.3	67.27	20.74	0/4
50	3	3600.0	8246.0	55.31	32.24	0/4	3600.0	10144.3	55.78	27.02	3/4	3600.0	10779.0	56.17	25.86	0/4

Table 3.7: Computational results considering total interdiction and knapsack constraint.

Config.		Total Interdiction																			
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot
		[MSTI_{Opt}]					[MSTI-V_2]					[MSTI-V_{12}]					[MSTI_{DBa}]				
10	0.2	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.3	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4
	0.4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4
	0.6	0.1	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.1	2.5	3.91	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	0.8	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.2	6.5	6.56	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	1.0	0.3	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4
	1.2	0.5	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4	0.6	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4
15	0.2	0.3	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.4	0.0	0.00	0.00	4/4
	0.4	0.4	0.0	0.00	0.00	4/4	0.8	0.0	0.00	0.00	4/4	0.4	2.8	1.42	0.00	4/4	0.7	0.0	0.00	0.00	4/4
	0.6	0.5	0.0	0.00	0.00	4/4	0.8	0.0	0.00	0.00	4/4	0.6	7.8	2.95	0.00	4/4	0.6	0.0	0.00	0.00	4/4
	0.8	0.9	0.0	0.00	0.00	4/4	1.5	0.0	0.00	0.00	4/4	1.0	20.0	7.01	0.00	4/4	1.0	0.0	0.00	0.00	4/4
	1.0	1.5	0.0	0.00	0.00	4/4	1.6	6.8	2.30	0.00	4/4	1.9	4.5	1.73	0.00	4/4	1.5	0.0	0.00	0.00	4/4
	1.2	9.3	69.5	29.02	0.00	4/4	10.1	77.5	37.34	0.00	4/4	4.8	17.0	4.77	0.00	4/4	1.9	31.5	15.13	0.00	4/4
20	0.2	0.9	0.0	0.00	0.00	4/4	1.7	0.0	0.00	0.00	4/4	1.0	0.5	0.14	0.00	4/4	1.5	0.0	0.00	0.00	4/4
	0.4	1.1	0.0	0.00	0.00	4/4	1.8	0.0	0.00	0.00	4/4	1.3	17.5	4.73	0.00	4/4	2.0	0.0	0.00	0.00	4/4
	0.6	1.5	0.0	0.00	0.00	4/4	2.6	0.0	0.00	0.00	4/4	1.7	12.5	4.32	0.00	4/4	5.7	0.0	0.00	0.00	4/4
	0.8	2.9	0.3	0.00	0.00	4/4	5.9	2.5	0.60	0.00	4/4	3.5	28.3	7.46	0.00	4/4	7.8	0.0	0.00	0.00	4/4
	1.0	11.6	48.5	4.74	0.00	4/4	13.2	25.0	4.75	0.00	4/4	7.8	5.5	2.22	0.00	4/4	16.1	136.3	31.42	0.00	4/4
	1.2	76.9	209.3	21.27	0.00	4/4	69.6	114.3	17.53	0.00	4/4	31.1	29.5	12.47	0.00	4/4	14.5	150.5	27.66	0.00	4/4
25	0.2	2.1	0.0	0.00	0.00	4/4	3.3	0.0	0.00	0.00	4/4	2.2	2.5	1.26	0.00	4/4	2.0	0.0	0.00	0.00	4/4
	0.4	2.4	0.0	0.00	0.00	4/4	4.0	0.0	0.00	0.00	4/4	2.6	11.3	2.37	0.00	4/4	5.0	0.0	0.00	0.00	4/4
	0.6	2.9	0.0	0.00	0.00	4/4	5.0	0.0	0.00	0.00	4/4	3.2	10.3	3.14	0.00	4/4	21.8	12.3	9.93	0.00	4/4
	0.8	6.4	6.8	1.17	0.00	4/4	11.6	3.3	0.47	0.00	4/4	5.9	18.5	3.77	0.00	4/4	48.6	89.0	24.89	0.00	4/4
	1.0	47.4	53.8	3.56	0.00	4/4	50.0	61.5	3.94	0.00	4/4	22.1	26.5	2.56	0.00	4/4	55.1	132.8	20.74	0.00	4/4
	1.2	318.5	314.3	79.22	0.00	4/4	323.3	195.8	40.69	0.00	4/4	104.3	80.0	5.88	0.00	4/4	89.4	521.0	54.71	0.00	4/4
30	0.2	4.2	0.0	0.00	0.00	4/4	6.7	0.0	0.00	0.00	4/4	4.5	3.5	1.96	0.00	4/4	6.0	1.3	0.79	0.00	4/4
	0.4	5.2	0.0	0.00	0.00	4/4	7.8	0.0	0.00	0.00	4/4	5.5	28.8	3.60	0.00	4/4	23.4	46.0	17.77	0.00	4/4
	0.6	7.9	7.3	0.69	0.00	4/4	12.4	3.5	0.75	0.00	4/4	7.6	29.5	5.74	0.00	4/4	100.1	115.5	37.11	0.00	4/4
	0.8	21.6	6.0	0.63	0.00	4/4	61.1	37.8	5.96	0.00	4/4	19.1	66.3	15.62	0.00	4/4	185.2	283.5	77.97	0.00	4/4
	1.0	599.9	332.5	15.15	0.00	4/4	649.4	377.5	12.89	0.00	4/4	121.4	77.0	7.06	0.00	4/4	247.4	598.3	80.25	0.00	4/4
	1.2	3146.5	379.3	183.86	71.58	1/4	3004.7	267.3	175.73	77.61	1/4	606.1	114.0	20.03	0.00	4/4	401.8	1247.8	91.62	0.00	4/4
50	0.2	34.0	0.0	0.00	0.00	4/4	35.5	3.5	0.68	0.00	4/4	33.5	2.0	0.46	0.00	4/4	180.0	16.5	7.09	0.00	4/4
	0.4	47.0	15.8	0.45	0.00	4/4	48.7	186.5	4.00	0.00	4/4	47.4	162.8	3.22	0.00	4/4	2027.1	98.5	36.98	12.07	3/4
	0.6	160.5	206.0	3.21	0.00	4/4	184.9	733.8	7.92	0.00	4/4	199.4	602.5	5.86	0.00	4/4	3600.0	13.0	61.13	59.70	0/4
	0.8	2508.4	277.5	3.37	0.85	2/4	3026.9	835.3	18.95	3.27	1/4	1340.3	988.0	15.41	0.00	4/4	3600.0	52.5	78.34	75.31	0/4
		[$\text{MSTI}_{\text{Dual}}$]					[$\text{MSTI}_{\text{SDual}}$]					[MSTI_{Tot}]									
10	0.2	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4					
	0.4	0.2	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4					
	0.6	0.2	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4					
	0.8	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4					
	1.0	0.2	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4					
	1.2	0.2	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4					

Table 3.7: Computational results considering total interdiction and knapsack constraint. (continued)

Config.		Total Interdiction														
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot
		[$\text{MSTI}_{\text{Dual}}$]					[$\text{MSTI}_{\text{SDual}}$]					[MSTI_{Tot}]				
15	0.2	0.3	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	0.4	0.5	0.0	0.00	0.00	4/4	0.6	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	0.6	0.6	0.0	0.00	0.00	4/4	0.6	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4
	0.8	0.9	0.0	0.00	0.00	4/4	0.8	0.0	0.00	0.00	4/4	0.6	0.0	0.00	0.00	4/4
	1.0	0.9	0.0	0.00	0.00	4/4	1.0	0.0	0.00	0.00	4/4	0.9	9.3	3.18	0.00	4/4
	1.2	1.3	32.3	4.38	0.00	4/4	1.1	0.0	0.00	0.00	4/4	3.1	37.0	8.68	0.00	4/4
20	0.2	0.6	0.0	0.00	0.00	4/4	0.6	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4
	0.4	1.0	0.0	0.00	0.00	4/4	1.1	0.0	0.00	0.00	4/4	0.6	0.0	0.00	0.00	4/4
	0.6	1.6	0.0	0.00	0.00	4/4	1.4	0.0	0.00	0.00	4/4	0.9	0.0	0.00	0.00	4/4
	0.8	2.3	0.0	0.00	0.00	4/4	2.8	0.0	0.00	0.00	4/4	1.5	0.0	0.00	0.00	4/4
	1.0	2.9	0.0	0.00	0.00	4/4	3.7	40.0	2.41	0.00	4/4	4.0	3.3	1.03	0.00	4/4
	1.2	6.3	311.5	23.84	0.00	4/4	4.3	54.5	11.16	0.00	4/4	16.0	59.3	5.85	0.00	4/4
25	0.2	1.5	0.0	0.00	0.00	4/4	1.4	0.0	0.00	0.00	4/4	1.0	0.0	0.00	0.00	4/4
	0.4	2.6	0.0	0.00	0.00	4/4	2.7	0.0	0.00	0.00	4/4	1.2	0.0	0.00	0.00	4/4
	0.6	7.0	0.0	0.00	0.00	4/4	9.1	11.3	2.13	0.00	4/4	1.4	0.0	0.00	0.00	4/4
	0.8	11.9	45.8	16.29	0.00	4/4	15.0	109.0	31.05	0.00	4/4	2.7	1.8	0.18	0.00	4/4
	1.0	23.2	136.8	20.07	0.00	4/4	17.4	104.3	9.97	0.00	4/4	15.5	17.8	2.17	0.00	4/4
	1.2	30.3	490.3	40.58	0.00	4/4	26.6	417.3	33.06	0.00	4/4	57.9	155.8	5.74	0.00	4/4
30	0.2	3.6	0.0	0.00	0.00	4/4	4.1	0.0	0.00	0.00	4/4	2.0	0.0	0.00	0.00	4/4
	0.4	8.7	0.0	0.00	0.00	4/4	8.7	0	0.00	0.00	4/4	2.4	0	0.00	0.00	4/4
	0.6	18.2	0.0	0.00	0.00	4/4	22.4	0	0.00	0.00	4/4	3.5	2.75	0.47	0.00	4/4
	0.8	63.7	208.8	21.40	0.00	4/4	62.5	165.5	24.04	0.00	4/4	7.7	6.0	1.33	0.00	4/4
	1.0	152.2	616.3	62.83	0.00	4/4	134.8	617.8	53.44	0.00	4/4	126.9	233.5	7.61	0.00	4/4
	1.2	209.3	1238.5	75.18	0.00	4/4	237.3	1168.0	69.92	0.00	4/4	945.5	1252.8	22.10	0.00	4/4
50	0.2	202.5	8.3	4.09	0.00	4/4	214.9	11.8	0.93	0.00	4/4	16.3	0.0	0.00	0.00	4/4
	0.4	1191.3	219.8	30.13	0.00	4/4	946.6	183.5	33.77	0.00	4/4	21.8	3.3	0.17	0.00	4/4
	0.6	2921.1	408.0	66.93	16.48	2/4	2855.5	477.3	64.29	9.51	3/4	38.6	82.8	1.33	0.00	4/4
	0.8	3600.0	459.0	80.34	53.35	0/4	3600.0	664.5	85.61	38.96	0/4	135.1	179.8	1.70	0.00	4/4

Table 3.8: Computational results considering partial interdiction and knapsack constraint.

Config.		Partial Interdiction														
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot
		[MSTI_{Opt}]					[MSTI-V_2]					[MSTI-V_{12}]				
10	0.2	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4
	0.4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4
	0.6	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.3	5.3	5.04	0.00	4/4
	0.8	0.2	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.2	1.8	1.78	0.00	4/4
	1.0	0.2	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	1.2	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4
15	0.2	0.3	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	0.4	0.4	0.0	0.00	0.00	4/4	0.4	0.0	0.00	0.00	4/4	0.4	2.5	1.14	0.00	4/4
	0.6	0.4	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4	0.5	8.5	2.93	0.00	4/4
	0.8	0.6	0.0	0.00	0.00	4/4	0.7	0.0	0.00	0.00	4/4	0.8	27.3	8.15	0.00	4/4
	1.0	0.8	0.0	0.00	0.00	4/4	0.8	0.0	0.00	0.00	4/4	0.9	0.0	0.00	0.00	4/4
	1.2	1.4	0.0	0.00	0.00	4/4	1.5	0.0	0.00	0.00	4/4	1.3	0.0	0.00	0.00	4/4
20	0.2	0.9	0.0	0.00	0.00	4/4	1.1	0.0	0.00	0.00	4/4	0.9	0.8	0.42	0.00	4/4
	0.4	1.1	0.0	0.00	0.00	4/4	1.3	0.0	0.00	0.00	4/4	1.2	16.5	3.83	0.00	4/4
	0.6	1.2	0.0	0.00	0.00	4/4	1.5	0.0	0.00	0.00	4/4	1.4	10.8	2.64	0.00	4/4
	0.8	1.8	0.0	0.00	0.00	4/4	2.6	2.3	0.70	0.00	4/4	2.2	46.8	6.41	0.00	4/4
	1.0	2.9	2.8	0.76	0.00	4/4	3.1	11.5	2.13	0.00	4/4	3.2	4.8	0.93	0.00	4/4
	1.2	4.5	20.8	3.29	0.00	4/4	5.2	9.0	1.16	0.00	4/4	4.7	20.8	3.44	0.00	4/4
25	0.2	2.2	0.0	0.00	0.00	4/4	2.8	0.0	0.00	0.00	4/4	1.5	3.8	0.75	0.00	4/4
	0.4	2.5	0.0	0.00	0.00	4/4	3.3	0.0	0.00	0.00	4/4	2.1	6.3	2.07	0.00	4/4
	0.6	3.0	0.0	0.00	0.00	4/4	3.9	0.0	0.00	0.00	4/4	2.4	11.8	3.95	0.00	4/4
	0.8	4.4	0.0	0.00	0.00	4/4	6.6	1.3	0.30	0.00	4/4	4.0	53.0	10.35	0.00	4/4
	1.0	16.1	28.0	2.46	0.00	4/4	15.4	21.8	2.10	0.00	4/4	8.8	2.8	0.50	0.00	4/4
	1.2	45.9	159.3	7.57	0.00	4/4	39.0	56.5	5.02	0.00	4/4	23.9	68.8	4.39	0.00	4/4
30	0.2	3.9	0.0	0.00	0.00	4/4	5.4	0.0	0.00	0.00	4/4	3.0	4.5	2.52	0.00	4/4
	0.4	5.2	0.0	0.00	0.00	4/4	6.1	0.0	0.00	0.00	4/4	3.8	34.3	3.76	0.00	4/4
	0.6	6.9	1.3	0.27	0.00	4/4	7.7	0.0	0.00	0.00	4/4	5.3	34.0	4.72	0.00	4/4
	0.8	14.9	7.3	1.41	0.00	4/4	19.0	15.0	1.77	0.00	4/4	10.4	71.3	6.53	0.00	4/4
	1.0	88.9	122.5	5.93	0.00	4/4	79.6	104.3	5.73	0.00	4/4	31.0	50.3	4.85	0.00	4/4
	1.2	282.6	393.5	10.60	0.00	4/4	255.8	277.5	8.81	0.00	4/4	77.8	88.8	6.91	0.00	4/4
50	0.2	29.7	0.0	0.00	0.00	4/4	30.4	4.3	0.68	0.00	4/4	21.0	11.3	1.20	0.00	4/4
	0.4	39.2	0.0	0.00	0.00	4/4	40.6	189.8	3.18	0.00	4/4	27.7	153.0	3.78	0.00	4/4
	0.6	91.2	123.8	3.40	0.00	4/4	103.3	716.0	5.75	0.00	4/4	97.0	879.5	11.34	0.00	4/4
	0.8	620.6	422.5	7.89	0.00	4/4	941.7	1845.0	16.53	0.00	4/4	626.8	1715.0	18.88	0.00	4/4
	1.0	3600.0	0.0	12.43	12.43	0/4	3600.0	0.0	17.78	17.78	0/4	2304.5	235.3	2.60	1.51	2/4
		[MSTI_{DBa}]					[$\text{MSTI}_{\text{Dual}}$]					[$\text{MSTI}_{\text{SDual}}$]				
10	0.2	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4
	0.4	0.1	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	0.6	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	0.8	0.2	2.8	0.45	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	1.0	0.2	0.0	0.00	0.00	4/4	0.1	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4
	1.2	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4	0.2	0.0	0.00	0.00	4/4

Table 3.8: Computational results considering partial interdiction and knapsack constraint. (continued)

Config.		Partial Interdiction														
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap_r(%)	Gap (%)	Opt/Tot
		[MSTI_{DBa}]					[$\text{MSTI}_{\text{Dual}}$]					[$\text{MSTI}_{\text{SDual}}$]				
15	0.2	0.4	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4	0.3	0.0	0.00	0.00	4/4
	0.4	0.5	1.3	2.31	0.00	4/4	0.5	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4
	0.6	0.6	2.5	1.35	0.00	4/4	0.8	0.0	0.00	0.00	4/4	0.5	0.0	0.00	0.00	4/4
	0.8	0.8	0.0	0.00	0.00	4/4	1.0	0.0	0.00	0.00	4/4	0.7	0.0	0.00	0.00	4/4
	1.0	0.9	0.0	0.00	0.00	4/4	0.7	0.0	0.00	0.00	4/4	0.6	0.0	0.00	0.00	4/4
	1.2	1.3	36.0	4.45	0.00	4/4	0.6	0.0	0.00	0.00	4/4	0.8	0.0	0.00	0.00	4/4
20	0.2	1.1	0.0	0.00	0.00	4/4	0.8	1.8	0.45	0.00	4/4	0.6	0.0	0.00	0.00	4/4
	0.4	1.7	0.0	0.00	0.00	4/4	0.8	0.0	0.00	0.00	4/4	0.8	0.0	0.00	0.00	4/4
	0.6	2.7	0.0	0.00	0.00	4/4	0.9	0.0	0.00	0.00	4/4	0.9	0.0	0.00	0.00	4/4
	0.8	3.7	0.0	0.00	0.00	4/4	1.0	0.0	0.00	0.00	4/4	1.1	0.0	0.00	0.00	4/4
	1.0	6.7	47.3	8.48	0.00	4/4	1.9	0.0	0.00	0.00	4/4	1.8	0.0	0.00	0.00	4/4
	1.2	9.4	205.5	11.33	0.00	4/4	1.9	0.0	0.00	0.00	4/4	1.9	0.0	0.00	0.00	4/4
25	0.2	2.0	0.0	0.00	0.00	4/4	1.3	0.0	0.00	0.00	4/4	1.8	3.0	3.60	0.00	4/4
	0.4	3.7	0.0	0.00	0.00	4/4	2.4	0.0	0.00	0.00	4/4	2.2	0.0	0.00	0.00	4/4
	0.6	15.9	11.3	2.30	0.00	4/4	3.3	0.0	0.00	0.00	4/4	3.0	0.0	0.00	0.00	4/4
	0.8	26.1	43.0	9.44	0.00	4/4	8.2	39.5	4.69	0.00	4/4	6.1	0.0	0.00	0.00	4/4
	1.0	33.7	162.5	18.48	0.00	4/4	11.1	86.0	6.19	0.00	4/4	10.7	119.8	11.88	0.00	4/4
	1.2	72.5	330.5	24.69	0.00	4/4	15.8	264.8	12.32	0.00	4/4	19.0	301.8	10.71	0.00	4/4
30	0.2	5.4	0.0	0.00	0.00	4/4	2.8	0.0	0.00	0.00	4/4	2.4	0.0	0.00	0.00	4/4
	0.4	29.4	0.0	0.00	0.00	4/4	7.1	0.0	0.00	0.00	4/4	7.1	0.00	0.00	0.00	4/4
	0.6	85.9	132.5	23.37	0.00	4/4	10.7	0.0	0.00	0.00	4/4	9.3	0.00	0.00	0.00	4/4
	0.8	136.9	283.0	35.39	0.00	4/4	23.5	98.5	7.42	0.00	4/4	34.1	68.0	2.26	0.00	4/4
	1.0	183.8	534.8	47.19	0.00	4/4	87.1	373.5	13.71	0.00	4/4	82.9	426.0	22.07	0.00	4/4
	1.2	239.9	991.3	51.81	0.00	4/4	127.4	898.3	26.46	0.00	4/4	153.8	996.8	27.79	0.00	4/4
50	0.2	219.0	15.8	6.25	0.00	4/4	219.7	13.8	4.10	0.00	4/4	212.2	7.3	0.54	0.00	4/4
	0.4	3153.1	34.3	20.21	12.25	2/4	924.1	116.0	14.79	0.00	4/4	1139.8	110.3	17.03	0.00	4/4
	0.6	3600.0	29.3	33.06	32.82	0/4	2777.1	501.3	34.97	1.85	3/4	2711.0	427.5	36.86	0.00	4/4
	0.8	3600.0	113.5	44.52	42.32	0/4	3600.0	845.0	51.81	21.61	0/4	3337.5	709.0	48.64	11.99	2/4
	1.0	3600.0	21.3	55.41	54.97	0/4	3600.0	571.5	231.90	36.50	0/4	3600.0	540.0	229.66	37.26	0/4

3.10 Conclusions

In this chapter we presented seven formulations in order to solve the MSTI problem. The first six formulations, i.e. [MSTI_{Opt}], [MSTI_{DBa}], [$\text{MSTI}_{\text{Dual}}$], [$\text{MSTI}_{\text{SDual}}$], [MSTI-$V_2$] and [MSTI-$V_{12}$], can handle total and partial interdiction as well as both types of budget constraints, cardinality and knapsack. The seventh formulation, [MSTI_{Tot}], is a special model that only considers total interdiction. [MSTI_{DBa}] is a model retrieved form the literature. [MSTI_{Opt}] is a formulation based on a linear model of the MST of Martin (1991). Two families of valid inequalities were developed to strengthen [MSTI_{Opt}] and consequently [MSTI-V_2] and [MSTI-V_{12}] are defined. [$\text{MSTI}_{\text{Dual}}$] is obtained by computing the dual of the follower's problem. Similarly, [$\text{MSTI}_{\text{SDual}}$] is as well obtained by computing the dual of the inner minimization problem but from a compact version of [MSTI_{Opt}]. [MSTI_{Tot}] is a special version of [MSTI_{Opt}] that only considers total interdiction and it is only equivalent to the other formulations under specific parameter settings.

Test instances with networks of different sizes were generated and used to evaluate the efficiency of the presented formulations. The experiments reported in Section 3.9 provide varied results. There is no formulation that consistently outperforms the other in all cases. The next conclusions follow:

- **Cardinality constraint - Total interdiction**. [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] proved to be the most efficient formulations by optimally solving most of the test instances in the shortest computational times. [MSTI_{DBa}] ranks second; although in several instances it obtained better relative gaps and a smaller number of explored Branch-and-Bound nodes, the computation times are significantly large. In addition, [MSTI_{DBa}] failed to solve some instances less than [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] when the budget increased. [MSTI_{Opt}], [MSTI-V_2], [MSTI-V_{12}] and [MSTI_{Tot}] become fairly inefficient starting on networks of 25 nodes. Furthermore, instances are unsolvable on graphs of size 50 or more with a budget greater than or equal to 5.

- **Cardinality constraint - Partial interdiction**. Similar conclusions as in the previous bullet point. [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] report the best results. However, [$\text{MSTI}_{\text{SDual}}$] obtained slightly better resolution times and

it solved one more instance than [$\text{MSTI}_{\text{Dual}}$]. [$\text{MSTI}_{\text{DBa}}$] occasionally continues to report slightly better relative gaps and smaller number of explored nodes but with a higher computational time. None of the formulations is capable of solving instances of 50 nodes, thus marking their limits in this case.

- **Knapsack constraint - Total interdiction**. [MSTI_{Tot}] is the most efficient formulation. Each instance is solved and most of them were solved at root node. [MSTI-V_{12}] takes the second over [MSTI_{Opt}] since the former obtain similar or better resolution times than the latter and it solved all instances. The roles of [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$] were reversed since in this case they did not report the best results. However, instances of 10 to 30 nodes are optimally solved in less than 240 seconds by these two formulations. Their results decay on the networks of largest size with largest budgets. [MSTI_{DBa}] becomes the most inefficient model in this case.

- **Knapsack constraint - Partial interdiction**. [MSTI_{Opt}] and [MSTI-V_{12}] becomes the most efficient models by solving every instance in similar times. [MSTI-V_{12}] is able to solve slightly larger instances than [MSTI_{Opt}]. [$\text{MSTI}_{\text{SDual}}$] ranks second and [$\text{MSTI}_{\text{Dual}}$] third. [$\text{MSTI}_{\text{DBa}}$] once again becomes the most inefficient model since it obtains the worst computational times and gaps.

Several formulations for the MSTI problem based on different MST models are proposed. These formulations are conceived to handle the various assumptions with respect to the types of interdiction and types of budget constraints. Computational experiments showed that at least one of our proposed formulations is efficient in networks of small and medium size for all four different cases. Experiments imply that larger networks are impossible to solve by means of the presented formulations. Therefore, different approaches other than pure mathematical models are proposed in the next chapters.

Chapter 4

A Branch-and-Price Algorithm for the MSTI Problem

The numerical experiments reported in the previous chapter indicate the need to design specific resolution methods. The computational time required by our formulations of the MSTI problem grows considerably as the size of the network increases.

One reason is the number of variables present in the models, for instance only variables x_e grow at a rate of $(|V|^2 - |V|)/2$ for complete graphs. Branch-and-Price (B&P) algorithm is a well-known method that can address models with a large number of variables. In this chapter, we develop a B&P algorithm based on the [$\text{MSTI}_{\text{SDual}}$] formulation, (3.9), in order to work with a much smaller number of variables and hopefully diminish the resolution time. The algorithm starts by reformulating [$\text{MSTI}_{\text{SDual}}$] via a Dantzig-Wolfe decomposition. Then the Master Problem (MP) is defined by relaxing the integrality of the latter reformulation. The MP has as many variables as the original model. Therefore, a more compact formulation that only considers a fraction of the variables (columns), named the Restricted Master Problem (RMP), is defined. This model is in turn solved by a column generation procedure embedded in a Branch-and-Bound scheme in order to recover integrality. In Section 4.1, we detail the decomposition of [$\text{MSTI}_{\text{SDual}}$] and the definition of the RMP. The pricing problem and the B&P algorithm for the MSTI problem are presented in Section 4.2. The chapter ends with computational experiments in Section 4.3, assuming all types of interdictions and budget constraints, and followed by conclusions in Section 4.4.

4.1 Dantzig-Wolfe Decomposition

In this section, the Dantzig-Wolfe decomposition is presented on formulation (3.9). Even though [MSTI$_{\text{SDual}}$] did not outperform all other models, it was chosen to be used in the B&P algorithm since it is the most compact formulation. Let $\mathcal{X}$ be the set of interdiction strategies defined as $\{\mathbf{x} \in \{0,1\}^{|\mathcal{E}|} \mid \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$. [MSTI$_{\text{SDual}}$] can be expressed in its *compact* form as:

$$[\text{MSTI}_{\text{SDual}}] \quad \max_{\mathbf{x},\beta,\boldsymbol{\sigma},\boldsymbol{\tau}} \beta(n-1) - \sum_{k\in\mathcal{V}} \sum_{\substack{m\in\mathcal{V} \\ m\neq k}} \tau_{km} \tag{4.1a}$$

s.t.

$$\beta + \sum_{k\in\mathcal{V}\setminus\{m,h\}} \sigma_{ke} - \tau_{hm} - \tau_{mh} \leq c_e + d_e x_e \qquad \forall e = \{m,h\} \in \mathcal{E} \tag{4.1b}$$

$$\sigma_{ke} + \tau_{km} \geq 0 \qquad \forall k,m \in \mathcal{V}, e = \{m,j\} \in \mathcal{E}\setminus\mathcal{E}^k \tag{4.1c}$$

$$\tau_{km} \geq 0 \qquad \forall k,m \in \mathcal{V}, m \neq k \tag{4.1d}$$

$$\beta, \sigma_{ke} = \textit{unrestricted} \qquad \forall k \in \mathcal{V}, \forall e \in \mathcal{E} \tag{4.1e}$$

$$x \in \mathcal{X} \tag{4.1f}$$

Let conv($\mathcal{X}$) represent the convex hull of $\mathcal{X}$. Due to the Weyl-Minkowski theorem (Schrijver (1998)), a point of $\mathcal{X}$ can be written as a convex combination of the extreme points in conv($\mathcal{X}$). Each point of conv($\mathcal{X}$) corresponds to an interdiction plan. The number of extreme points is finite since there is only a finite number of interdiction strategies. Therefore, any interdiction plan can be expressed as a convex combination of interdiction plans. Let $\mathcal{P}$ be the set of extreme points. The constraints that define the convex combination of interdiction strategies are the following:

$$\mathbf{x} = \sum_{p\in\mathcal{P}} \lambda_p \mathbf{x}_p \tag{4.2a}$$

$$\sum_{p\in\mathcal{P}} \lambda_p = 1 \tag{4.2b}$$

$$\lambda_p \geq 0 \qquad \forall p \in \mathcal{P} \tag{4.2c}$$

The non-negative weights of each interdiction plan $\mathbf{x}_p$ in the convexity constraint (4.2a) are represented by λ_p. The meaning of each λ_p is the proportion of strategy $\mathbf{x}_p$ to obtain $\mathbf{x}$. The component-wise version of (4.2a) is $x_e = \sum_{p\in\mathcal{P}} x_{pe}\lambda_p$ for all $p \in P$ where x_{pe} is equal to 1 if the edge e is interdicted in the interdiction strategy p, and 0 otherwise. Then, the next *extended* formulation is obtained by using expression (4.2a) to substitute x_e in model (4.1) and by adding constraints (4.2b)-(4.2c).

$$[\text{Ext-MSTI}_{\text{SDual}}] \quad \max_{\mathbf{x},\beta,\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\lambda}} \beta(n-1) - \sum_{k\in\mathcal{V}}\sum_{\substack{m\in\mathcal{V}\\ m\neq k}} \tau_{km} \tag{4.3a}$$

s.t.

$$\beta + \sum_{k\in\mathcal{V}\setminus\{m,h\}} \sigma_{ke} - \tau_{hm} - \tau_{mh} \leq c_e + \sum_{p\in\mathcal{P}} (d_e x_{pe})\lambda_p \quad : \{\pi^1_{mh}\} \quad \forall e = \{m,h\} \in \mathcal{E} \tag{4.3b}$$

$$\sigma_{ke} + \tau_{km} \geq 0 \qquad \forall k, m \in \mathcal{V}, e = \{m,j\} \in \mathcal{E}\setminus\mathcal{E}^k \tag{4.3c}$$

$$\sum_{p\in\mathcal{P}} \lambda_p = 1 \qquad : \{\pi^0\} \tag{4.3d}$$

$$x_e = \sum_{p\in\mathcal{P}} x_{pe}\lambda_p \qquad \forall e \in \mathcal{E} \tag{4.3e}$$

$$\lambda_p, \tau_{km} \geq 0, \qquad \forall p \in \mathcal{P}, \forall k, m \in \mathcal{V}, m \neq k \tag{4.3f}$$

$$\beta, \sigma_{ke} = unrestricted \qquad \forall k \in \mathcal{V}, \forall e \in \mathcal{E} \tag{4.3g}$$

$$x_e \in \{0,1\} \qquad \forall e \in \mathcal{E} \tag{4.3h}$$

[Ext-MSTI$_{\text{SDual}}$] contains a quadratic number of variables. Thus solving (4.3) with all the variables is computationally burdensome. Therefore, only a subset of variables is considered and the others are generated via a column generation procedure. We begin by considering the linear relaxation of (4.3) by removing (4.3h). The integrality of variables x_e and (4.3d) impose to have only one variable λ_p to be one. Since (4.3h) is removed, several variables λ_p may take positive values simultaneously. Additionally, parameters $x_{pe} \in \{0,1\}$ and $0 \leq \lambda_p \leq 1$, for all $p \in \mathcal{P}$, due to (4.3d) and (4.3f). Thus, $0 \leq \sum_{p\in\mathcal{P}} x_{pe}\lambda_p \leq 1$. Constraint (4.3e) is dropped since it is implicit in the formulation when the integrality of x_e is relaxed. The resulting formulation defined by (4.3a)-(4.3d), (4.3f) and (4.3g) is denoted as the master problem MP[$\mathcal{P}$].

Let $\hat{\mathcal{P}} \subseteq \mathcal{P}$ be a sub-set of interdiction strategies. We call Restricted Master Problem defined over $\hat{\mathcal{P}}$, indicated by [RMP($\hat{\mathcal{P}}$)], the problem defined by (4.3a)-(4.3d), (4.3f) and (4.3g). A column generation procedure is an iterative algorithm where the RMP and the pricing problem (or sub-problem) are solved sequentially and iteratively. The pricing problem identifies columns, which are in fact the extreme points of conv($\mathcal{X}$), with a positive reduced cost. These columns are added to [RMP($\hat{\mathcal{P}}$)] and then the problem is re-optimized. The pricing problem is presented in the following section.

We develop a Branch-and-Bound algorithm, where the column generation procedure is performed at each node of the search tree, to obtain an integer solution that can then be converted to a solution of an instance of the MSTI problem. This framework is known as a B&P algorithm since variables are prized in the course of the algorithm. The overall framework is presented in detail in the next section.

4.2 B&P Algorithm

In this section, we detail the B&P algorithm developed to solve the MSTI problem. The main idea is to design a classical Branch-and-Bound algorithm that solves at each node of the search tree the [RMP($\hat{\mathcal{P}}$)] by a column generation procedure. In the following sections, we describe the details of the different components of our algorithms.

4.2.1 Branching

The branching strategy chosen for the B&P algorithm is the traditional strategy. It consists in selecting the edge whose value is the closest to 0.5. Then, a binary branching is imposed on this edge. Since the [RMP($\hat{\mathcal{P}}$)] no longer has variables x_e to branch on, the branching is translated to use variables λ_p. Let $\hat{\mathcal{E}}_1, \hat{\mathcal{E}}_0 \subset \mathcal{E}$ be the sets of edges where a branching strategy is to be imposed at a given node, i.e. $x_e = 1$ for all $e \in \hat{\mathcal{E}}_1$ and $x_e = 0$ for all $e \in \hat{\mathcal{E}}_0$. The following constraints are thus defined:

$$\sum_{p\in\hat{\mathcal{P}}} x_{pe}\lambda_p \begin{cases} \geq 1, & \text{if } x_e = 1 \quad \forall e \in \hat{\mathcal{E}}_1 \\ \leq 0, & \text{if } x_e = 0 \quad \forall e \in \hat{\mathcal{E}}_0 \end{cases} \qquad : \{\pi_e^2\} \tag{4.4}$$

If an edge $e \in \hat{\mathcal{E}}_1$ is to be part of the final solution, only interdiction plans containing e must be selected. Therefore, the sum of variables λ_p that represent these interdiction strategies are set to be greater or equal to one. For the edges in $\hat{\mathcal{E}}_0$, they are set to zero.

The classical Branch-and-Bound stopping conditions for branching are considered in our algorithm. Those are: the solution at the node is integer, the solution at the node is infeasible or the upper bound of the node is lower than the lower bound. In addition, one more condition is defined to fathom nodes in an early stage. If $b_e > B - \sum_{e\in\hat{\mathcal{E}}_1} b_e$, for all $e \in \mathcal{E} \setminus \hat{\mathcal{E}}_1 \bigcup \hat{\mathcal{E}}_0$, then the node is fathomed. That is, the interdiction costs of the edges not yet considered for branching are greater than the remaining budget.

4.2.2 Pricing Problem

The pricing problem is defined by using the dual variables with respect to constraints involving λ_p, i.e. the convexity constraint, (4.3b) and the branching constraints (4.4). The dual variables corresponding to these constraints are given in (4.3) and (4.4). The pricing problem used to identify columns with a positive reduced cost turns out to be a knapsack problem with a modified objective function. The sub-problem is defined as follows:

$$\bar{c}(\hat{\mathbf{x}}) = \max \sum_{e=\{m,h\}\in\mathcal{E}} d_e\pi_{mh}^1\hat{x}_e - \sum_{e\in\hat{\mathcal{E}}_1} \pi_e^2\hat{x}_e - \pi^0 \tag{4.5a}$$

$$\sum_{e\in\mathcal{E}} b_e\hat{x}_e \leq B \tag{4.5b}$$

$$\hat{x}_e = 1 \qquad \forall e \in \hat{\mathcal{E}}_1 \tag{4.5c}$$

$$\sum_{e\in\hat{\mathcal{E}}_0} \hat{x}_e = 0 \tag{4.5d}$$

$$\hat{x}_e \in \{0,1\} \qquad \forall e \in \mathcal{E} \tag{4.5e}$$

The optimal solution of (4.5) is an interdiction plan that meets the budget constraint (4.5b) and the branching strategies of the node given by (4.5c) and (4.5d). A column $\hat{\mathbf{x}}$ is added to $\hat{\mathcal{P}}$ when $\bar{c}(\hat{\mathbf{x}}) > 0$. Afterwards, [RMP($\hat{\mathcal{P}}$)] is re-optimized considering the new column respect to $\hat{\mathbf{x}}$. This process is repeated until the optimal solution of the pricing problem does not generate an interdiction strategy with a positive reduced cost, that is $\bar{c}(\hat{\mathbf{x}}) \leq 0$. Thus, all relevant columns were found and the iterative procedure ends.

4.2.2.1 Strengthening of the Pricing Problem

The reduction of the network by means of the $\bar{B}$-edge connected certificate of $\mathcal{G}$, as shown in Section 3.8, gives rise to a family of constraints that can be used to strengthen the pricing problem (4.5). The graph $\mathcal{G}_{\bar{B}} = \cup_{i=0}^{i} T_i$ where T_0 is the MST of $\mathcal{G}$ and T_i the MST of $\mathcal{G}_i = \mathcal{G} \setminus \cup_{j=0}^{i-1} T_j$ represents the $\bar{B}$-edge connected certificate of $\mathcal{G}$. The equivalence of solving the MSTI problem in $\mathcal{G}$ and $\mathcal{G}_{\bar{B}}$ was proved in Chapter 3. In addition, the reduction of $\mathcal{G}$ is valid for both interdiction constraints where $\bar{B} = B$ in the cardinality case and $\bar{B} = \max\{\sum_{e\in\mathcal{E}} x_e : \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$ for the knapsack case. The following lemma is given.

Lemma 4.1. *Let $\mathcal{S}$ be an edge set whose elements are partially interdicted. Let $\mathcal{S}_a = \mathcal{S} \setminus \cup_{j=0}^{i-1} T_j$. If $e \in T_i$, then the replacement edge of e, i.e. an edge e' such that $T_i \setminus \{e\} \cup \{e'\}$ forms a tree and the increment in its weight is minimal, belongs to $T_{i+1} \cup \mathcal{S}_a \cup \{e\}$.*

Proof. Assuming total interdiction, the work of Bazgan *et al.* (2012) proves that the replacement edge of e belongs to T_{i+1}.

If partial interdiction is assumed, edge e is still part of the graph. It might be that $c_e < c_{e'}$ where e' is the edge with the smallest weight of T_{i+1} that connects the two connected components induced by removing e from T_i. Analogously, such edge e' can belong to $\mathcal{S}_a$. Thus, the replacement edge of e is found in the set $T_{i+1} \cup \mathcal{S}_a \cup \{e\}$. □

In Bazgan *et al.* (2012), a sequential procedure embedded in a construction of a search tree and based in Lemma 4.1 is proposed for the MSTI problem considering total interdiction and cardinality constraint. The sequential procedure consists in $\bar{B}$ iterations. At each iteration, one edge of T_0 is interdicted and replaced by

one edge of T_1, the edge removed from T_1 that was added to T_0 is now replaced for one edge of T_2, and so on. At the end of $\bar{B}$ iterations, $\bar{B}$ edges are interdicted. The authors proved that the optimal solution of the MSTI problem has the same structure given by the latter algorithm, i.e. the solution can be constructed by $\bar{B}$ iterative interdictions and replacements.

The particular structure of the procedure above gives rise to next family of valid inequalities of the MSTI problem.

$$\sum_{e \in T_i} x_e \leq \bar{B} - i \qquad \forall i = 0, 1, ..., \bar{B} \tag{4.6}$$

Constraints (4.6) give an upper bound on the number of edges that can be interdicted for each T_i, $0 \leq i \leq \bar{B}$. Note that at each step of the sequential algorithm, an edge of T_i is replaced by one in T_{i+1}. That is, edges of the "lower" trees are climbing up to T_0. A total of $\bar{B}$ edges in T_0 can be interdicted. For T_1, we can interdict at most $\bar{B} - 1$ since at least one edge in T_0 must be interdicted otherwise, the follower will chose T_0 as his optimal response to the leader's interdiction strategy. Analogously, in any optimal interdiction strategy at most $\bar{B} - 2$ edges in T_2 can be interdicted since at least one edge in T_0 and one in T_1 need to be interdicted by the leader. Thus, the pricing problem, denoted by [Pr], is defined by (4.5) and the family of constraints (4.6).

4.2.2.2 Heuristic for the Pricing Problem

The first idea to solve the pricing problem, i.e. (4.5) and (4.6) was through a commercial solver. However, preliminary experimentation showed that this approach was time consuming. Therefore, a simple heuristic is developed. The problem defined by (4.5a), (4.5b), (4.5e) and (4.6) is what is known in the literature as the $0-1$ multidimensional knapsack problem. Therefore, we base our heuristic on the one proposed in Akçay *et al.* (2007).

Let $\mathbf{Q} \in \mathbb{R}^{|\mathcal{E}|}$ be the vector of coefficients of x_e by developing the summations in objective function (4.5a). Let $\bar{B}_{\bar{B}-i} = \bar{B} - i$ for $i = 0, 1, ..., \bar{B}$. Parameters $\bar{B}_{\bar{B}-i}$ are called tree budgets and represent the right-hand sides of constraints (4.6). Let $\bar{c}(\hat{\mathbf{x}})$ be the objective value of [Pr] for the interdiction plan $\hat{\mathbf{x}}$. The heuristic consists in constructing an interdiction plan by adding the edge with

the largest Q_e to the solution one at a time. Each time an edge e is added to the interdiction strategy, the budget is reduced by the interdiction cost of the edge, i.e. $B - b_e$, and its corresponding tree budget is reduced by one. In addition, $\bar{c}(\hat{\mathbf{x}})$ is increased by Q_e.

The heuristic is divided into four main phases. First, the edges in $\mathcal{E}_1$ imposed by the branching strategy are added to the solution; their respective Q_e values are added to $\bar{c}(\hat{\mathbf{x}})$ as well as the budget and the tree budgets are correspondingly modified. The second phase forbids the interdiction of edges in $\mathcal{E}_0$. Thirdly, the edges that cannot be interdicted due to the violation of any constraints are pruned. The last phase selects one of the remaining edges with the largest Q_e coefficient, the edge is added to the interdiction plan and the reduced cost and the budgets are modified. The algorithm iterates between phases three and four until no more edges can be selected.

The pseudo-code of the heuristic for [Pr] is presented in Algorithm 1. Step 1 initialize $\bar{c}(\hat{\mathbf{x}})$ to constant π^0, the interdiction plan $\hat{\mathbf{x}}$ to the empty strategy and vector $\mathbf{Q}$ to the zero vector. Step 2 computes coefficient $d_e\pi^1_{mh}$ for each edge and saves it in Q_e. Steps 3 to 10 correspond to phase one described above; when an edge is added to the solution is then removed from the edge set $\mathcal{E}$ in Step 9. Phase two is carried out in Step 11 to 13 and the edges of $\mathcal{E}_0$ are removed form $\mathcal{E}$. The loop between phases three and four start in Step 14 where the third phase related to edge pruning goes from Step 15 to 20 and the fourth phase from Step 24 to 29. The termination condition in Step 21 ends the loop where all edges have been eliminated from the edge set. Finally, the algorithms returns a feasible interdiction plan $\hat{\mathbf{x}}$ with reduced cost $\bar{c}(\hat{\mathbf{x}})$.

The heuristic presented in Algorithm 1 might not always provide the optimal solution. To ensure the correctness of the B&P algorithm, [Pr] is solved by a commercial solver only when Algorithm 1 does not provide columns with positive reduced costs. If a column with a positive reduced cost is identified, it is added to $\hat{\mathcal{P}}$ and the MP is re-optimized. Otherwise, the node is considered to be completely solved and the B&P algorithm continues.

4.2.3 Lower Bound

The computation of the lower bound for the MSTI problem differs what is usually implemented in the literature. Each interdiction plan $\hat{\mathbf{x}}$ found by [Pr] defines

Algorithm 1 Heuristic for the Pricing Problem

Ensure: An interdiction plan $\hat{\mathbf{x}}$ with reduced cost $\bar{c}(\hat{\mathbf{x}})$

$\bar{c}(\hat{\mathbf{x}}) \leftarrow \pi^0$, $\hat{\mathbf{x}} \leftarrow \mathbf{0}$, $\boldsymbol{Q} \leftarrow \mathbf{0}$
$Q_e \leftarrow d_e \pi^1_{mh}$ for all $e \in \mathcal{E}$
for $e \in \hat{\mathcal{E}}_1$ **do**
    $Q_e \leftarrow Q_e - \pi^2_e$
    $\hat{x}_e \leftarrow 1$
    $B \leftarrow B - b_e$
    $B_{\bar{B}-i} \leftarrow B_{\bar{B}-i} - 1$ where $i \in \{0, 1, .., \bar{B}\}$ such that $e \in T_i$
    $\bar{c}(\hat{\mathbf{x}}) \leftarrow \bar{c}(\hat{\mathbf{x}}) + Q_e$
    $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e\}$
end for
for $e \in \hat{\mathcal{E}}_0$ **do**
    $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e\}$
end for
while $\mathcal{E} \neq \emptyset$ **do**
    for $e \in \mathcal{E}$ **do**
        Let $i \in \{0, 1, .., \bar{B}\}$ such that $e \in T_i$
        if $B < b_e \vee B_{\bar{B}-i} = 0$ **then**
            $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e\}$
        end if
    end for
    if $\mathcal{E} = \emptyset$ **then**
        Go to 31
    end if
    Let $e^* \in \mathcal{E}$ be the edge with the largest Q_e
    $\hat{x}_{e^*} \leftarrow 1$
    $B \leftarrow B - b_{e^*}$
    $B_{\bar{B}-i} \leftarrow B_{\bar{B}-i} - 1$ where $i \in \{0, 1, .., \bar{B}\}$ such that $e^* \in T_i$
    $\bar{c}(\hat{\mathbf{x}}) \leftarrow \bar{c}(\hat{\mathbf{x}}) + Q_{e^*}$
    $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e^*\}$
end while
Return $\hat{\mathbf{x}}$ and $\bar{c}(\hat{\mathbf{x}})$

a feasible interdiction plan for the MSTI problem since the only feasibility constraints of the interdictor's problem are the budget and integrality constrains. Therefore, the MST with respect to each $\hat{\mathbf{x}}$ can be computed on graph $G_{\hat{\mathbf{x}}}$ where the weight of each edge is $c_e + d_e\hat{x}_e$. Each MST computed in this manner provides a valid lower bound for the MSTI problem.

In a classical Branch-and-Bound algorithm, when an integer solution is found at any given node, the objective value of this solution is compared to the current best lower bound. If it has a greater value, the current lower bound is replaced by this solution. An integer solution of [MP($\mathcal{T}$)] has only one variable λ_{p^*} positive and the remaining ones are zero. In other words, the interdiction plan represented by p^* is selected among all others in $\hat{\mathcal{P}}$ as optimal plan. This means that the interdiction plan that gives the best lower bound was generated at some iteration before the algorithm selects this strategy as an optimal (global or local) integer solution. What we proposed is to compute the MST with respect to $\hat{\mathbf{x}}$, as stated above, for all columns generated by the pricing problem during the resolution of a given node. The latter leads to identify the best lower bound even when no integer solution has been found.

4.2.4 Overall presentation of the algorithm

Let $\mathcal{ST}$ and n_i, $i \in \mathbb{N}$, represent the set of nodes and a node of the search tree, respectively. For any set of edges $\mathcal{E}'$, let $w(\mathcal{E}') = \sum_{e\in\mathcal{E}'} c_e$. Given an interdiction plan $\hat{\mathbf{x}}$, the reaction of the follower to the interdiction, i.e. the MST of the augmented graph $\mathcal{G}_{\hat{\mathbf{x}}}$ where the weight of each edge is $c_e + d_e\hat{x}_e$, is denoted by $T(\hat{\mathbf{x}})$. Algorithm 2 describes the B&P algorithm developed to solve the MSTI problem.

Algorithm 2 defines a column generation procedure embedded in a Branch-and-Bound algorithm with all components presented in previous sections. The B&P algorithm ends when all nodes have been explored or the time limit is reached, ensured in Step 3. Node selection in Step 4 can be done in different ways. In the computation experiments, we explore the search tree using the best first strategy. In the best first strategy, the most promising node is selected as the first to be explored, that is the node with the best upper bound. Step 5 fathoms nodes that have an upper bound smaller than the lower bound.

Algorithm 2 Branch-and-Price Algorithm (Part 1)

Require: An interdiction plan $\mathbf{x}^*$ with its corresponding MST $\mathbf{y}^*$

1: Define n_0 as the root node
2: $\mathcal{ST} \leftarrow \{n_0\}$, $\hat{\mathcal{P}} \leftarrow \{\mathbf{0}\}$, $\underline{u} \leftarrow -\infty$, $\bar{c}(\hat{\mathbf{x}}) \leftarrow \infty$, $k \leftarrow 0$ $j \leftarrow 0$
3: **while** $\mathcal{ST} \neq \emptyset \vee$ Time limit is not reached **do**
4: Select $n_i \in \mathcal{ST}$
5: **if** $\underline{u} < \bar{u}_i$ **then**
6: (Re)Define branching constraints (4.4) of n_i
7: **while** $\bar{c}(\hat{\mathbf{x}}) > 0$ **do**
8: Solve [RMP($\hat{\mathcal{P}}$)] with objective v
9: $\bar{u}_i \leftarrow v$
10: Retrieve π^0, π^1_{mh} and π^2_e and define [Pr]
11: Solve [Pr] with Algorithm 1 and obtain solution $\hat{\mathbf{x}}$ with reduced cost $\bar{c}(\hat{\mathbf{x}})$
12: **if** $(\bar{c}(\hat{\mathbf{x}}) < 0) \vee (\lfloor \bar{u}_i \rfloor = \lfloor \bar{u}_i + \bar{c}(\hat{\mathbf{x}}) \rfloor) \vee (\bar{u}_i + \bar{c}(\hat{\mathbf{x}}) \leq \underline{u})$ **then**
13: Solve [Pr] with exact method and obtain solution $\hat{\mathbf{x}}$ with reduced cost $\bar{c}(\hat{\mathbf{x}})$
14: **if** $(\bar{c}(\hat{\mathbf{x}}) < 0) \vee (\lfloor \bar{u}_i \rfloor = \lfloor \bar{u}_i + \bar{c}(\hat{\mathbf{x}}) \rfloor) \vee (\bar{u}_i + \bar{c}(\hat{\mathbf{x}}) \leq \underline{u})$ **then**
15: Go to 21
16: **end if**
17: **end if**
18: Add variable λ_p corresponding to $\hat{\mathbf{x}}$ to [RMP($\hat{\mathcal{P}}$)]
19: $\hat{\mathcal{P}} \leftarrow \hat{\mathcal{P}} \cup \{\hat{\mathbf{x}}\}$
20: **end while**
21: **for** $\hat{\mathbf{x}}_{p_j} \in \hat{\mathcal{P}}$ **do**
22: Calculate $T(\hat{\mathbf{x}}_{p_j})$
23: **if** $w(T(\hat{\mathbf{x}}_{p_j})) > \underline{u}$ **then**
24: $\underline{u} \leftarrow w(T(\hat{\mathbf{x}}_{p_j}))$
25: $\mathbf{x}^* \leftarrow \hat{x}_{p_j}$
26: $\mathbf{y}^* \leftarrow T(\hat{\mathbf{x}}_{p_j})$
27: **end if**
28: $j \leftarrow j + 1$
29: **end for**

Algorithm 2 Branch-and-Price Algorithm (Part 2)

30: **if** $(\boldsymbol{\lambda} \notin \{0,1\}^{|\hat{\mathcal{P}}|}) \wedge (\underline{u} < \bar{u}_i) \wedge (\sum_{e \in \hat{\mathcal{E}}_1} b_e < B) \wedge (\exists e \in \mathcal{E} \setminus \hat{\mathcal{E}}_1 \bigcup \hat{\mathcal{E}}_0 : b_e \leq B - \sum_{e \in \hat{\mathcal{E}}_1} b_e)$ **then**
31: Select an edge e' to branch
32: Define n_{k+1} with $x_{e'} = 1$ and n_{k+2} with $x_{e'} = 0$
33: $\mathcal{ST} \leftarrow \mathcal{ST} \cup \{n_{k+1}\} \cup \{n_{k+2}\}$
34: $k \leftarrow k + 2$
35: **end if**
36: $\mathcal{ST} \leftarrow \mathcal{ST} \setminus \{n_i\}$
37: **else**
38: $\mathcal{ST} \leftarrow \mathcal{ST} \setminus \{n_i\}$
39: **end if**
40: **end while**
41: Return $\mathbf{x}^*$ and $\mathbf{y}^*$

Steps 6 to 36 represent the resolution and branching on a node. First, the RMP is adapted to meet the branching conditions of the given node in Step 6. The column generation procedure starts in Step 7. Basically, [RMP($\hat{\mathcal{P}}$)] is solved considering the restricted set of interdiction strategies $\hat{\mathcal{P}}$. The value of the dual variables with respect to the constraints involving λ_p are retrieved and used to define the pricing problem in Step 11. The pricing problem is solved by the heuristic described in Algorithm 1. If a column with a positive reduced cost is found, it is added to [RMP($\hat{\mathcal{P}}$)] and set $\hat{\mathcal{P}}$ is updated. Otherwise, a column with a positive reduced cost is tried to be obtained by a commercial solver, if it fails the column generation ends. The column generation procedure has other termination conditions apart from not finding solutions of the pricing problem with positive reduced costs. These termination conditions are listed below:

1. A negative reduced cost is obtained by the pricing problem.

2. The sum of the upper bound of the node and the positive reduced cost of the solution given by the [Pr] is lower than or equal to the global lower bound.

3. The reduced cost of the column is not large enough to allow the upper bound of the node to increase to the next unit. This is, the sum of the upper bound and the reduced cost is not larger than the floor of the upper

bound. This is only applied if the coefficients and right-hand sides of (4.5b) and (4.6) are integer.

Algorithm 2 updates the lower bound by considering the elements of $\hat{\mathcal{P}}$. Only the new columns generated once a node is solved by column generation are considered to improve the value of the lower bound. The procedure that computes a MST for each element of $\hat{\mathcal{P}}$ is performed from step 21 to 28. The weight of the MST with the largest weight becomes the best lower bound.

The only case where the integrality of the solution is verified is in step 30 in order to decide whether there is a need of branching. Steps 30 to 36 are classical steps of branching node. If the solution of the node is not integer and it has a smaller value than the lower bound, no children are generated and the node is deleted by dominance. Otherwise, an edge is selected and two children are generated. The most traditional branching strategy is used for edge selection for the branching. The edge with the value closest to 0.5 is chosen. The first edge found by the algorithm during the search for such an edge is selected in case of a tie. Once every node of the search tree is explored, the best lower bound becomes the optimal solution of the problem and the algorithm ends by providing the solution.

4.3 Computational Experiments

To evaluate the efficiency of the B&P algorithm is tested in this section, four configurations of type of interdiction and budget constraints are considered. The testbed includes the instances generated in Section 3.9. The $\bar{B}$-edge connected certificate is considered for each of the networks. When a cardinality constraint is imposed, we set $\bar{B} = B$. In the case of a knapsack constraint, we have $\bar{B} = \max\{\sum_{e\in\mathcal{E}} x_e : \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$ as in the experimentation of Section 3.9. The reduction on the size of $\mathcal{G}$ and the efficiency of constraints (4.6) only have an impact when we consider a cardinality budget constraint. This is due to the large value for $\bar{B}$ obtained by solving the problem $\max\{\sum_{e\in\mathcal{E}} x_e : \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$.

The pricing problem, defined by (4.5) and (4.6), is solved by the heuristic described in Algorithm 1 to find a column with a positive reduced cost. The Kruskal algorithm is considered for the computation of the MSTs. Every procedure is implemented in C++. The time limit is one hour. The experimentation

Config.		Cardinality Budget Constraint							
		Total Interdiction				Partial Interdiction			
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
10	3	4.5	148.5	0.00	4/4	3.5	131.5	0.00	4/4
	5	143.6	922.5	0.00	4/4	45.4	589.5	0.00	4/4
	7	1870.9	2487.5	0.85	3/4	129.0	659.5	0.00	4/4
15	3	36.3	581.5	0.00	4/4	81.6	1136.5	0.00	4/4
	5	3600.0	4187.0	13.96	0/4	3110.5	4006.5	5.81	2/4
20	3	206.2	858.0	0.00	4/4	506.9	1581.5	0.00	4/4
	5	3600.0	2605.5	22.83	0/4	3600.0	2991.5	16.41	0/4
25	3	1238.4	1954.5	0.00	4/4	2831.0	3830.5	2.32	3/4
	5	3600.0	1697.0	26.64	0/4	3600.0	1562.5	32.12	0/4
30	3	2692.3	3530.5	0.00	4/4	3600.0	3749.5	12.23	0/4
	5	3600.0	1322.0	37.35	0/4	3600.0	1258.0	49.80	0/4

Table 4.1: Computational results for Cardinality Budget Constraint

is performed in a 64-bit desk computer with a 2.20GHz Intel(R) Xeon(R) CPU E5-2630 v4 processor and 64 GB of RAM memory.

Computational results for both types of interdiction when we impose a cardinality constraint for the budget are reported in Table 4.1. For each configuration, the average time, the average number of nodes of the Branch-and-Bound tree and the average gap of four test instances are given. The last column of each result set presents the number of optimally solved instances out of the four tested.

Most of the networks with a budget of $B = 3$ are solved. However, the resolution time becomes large as the size of the network increases, being larger than 200 seconds for graphs of 20 nodes or more. Except for networks with $|\mathcal{V}| = 10$, the B&P algorithm is practically unable to solve graphs with an interdiction budget of 5 or more. For total interdiction, the average relative gaps between the best known solution and the upper bound retrieved at time limit of networks with 15 nodes more increase gradually from 13.96% to 37.50%. In the case of partial interdiction, from 5.81% to 49.80%. Moreover, for the latter interdiction, no network of 30 nodes in size was optimally solved.

Table 4.2 contains the computational experimentation with respect to the knapsack budget constraint. For simplification, the fraction η of the largest edge weight is reported as the budget. Networks of 10 nodes are optimally solved in a reasonable time of less than 24 seconds. The B&P algorithm starts to struggle on graphs of 15 nodes, $\eta = 0.8$ and total interdiction. Two out of four instances are not solved. For partial interdiction, all 15 nodes graphs are solved. However,

Config.		Knapsack Budget Constraint							
		Total Interdiction				Partial Interdiction			
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
10	0.2	0.1	5.5	0.00	4/4	0.1	3.5	0.00	4/4
	0.4	0.4	15.5	0.00	4/4	0.2	5.5	0.00	4/4
	0.6	3.0	94.0	0.00	4/4	1.5	44.5	0.00	4/4
	0.8	24.0	594.5	0.00	4/4	4.4	99.0	0.00	4/4
15	0.2	0.5	4.0	0.00	4/4	0.6	5.5	0.00	4/4
	0.4	10.4	85.5	0.00	4/4	4.5	48.5	0.00	4/4
	0.6	236.3	1218.5	0.00	4/4	41.7	348.0	0.00	4/4
	0.8	2107.0	3908.5	15.61	2/4	1092.2	2226.0	0.00	4/4
20	0.2	9.5	28.5	0.00	4/4	9.7	33.0	0.00	4/4
	0.4	159.5	414.5	0.00	4/4	213.3	509.0	0.00	4/4
	0.6	2781.0	3680.0	13.37	1/4	1555.6	2483.0	1.55	3/4
	0.8	3600.0	3206.0	27.04	0/4	1690.4	2163.0	3.03	3/4
25	0.2	37.6	91.5	0.00	4/4	31.2	66.5	0.00	4/4
	0.4	1412.4	2893.0	0.32	3/4	1197.0	2213.0	2.56	3/4
	0.6	3295.3	2672.5	35.17	1/4	1965.2	2022.0	10.73	2/4
30	0.2	104.1	142.5	0.00	4/4	225.4	296.5	0.00	4/4
	0.4	2753.0	2291.5	11.61	2/4	3065.2	2628.0	7.58	2/4
	0.6	3600.0	1110.0	51.78	0/4	3600.0	1844.5	21.88	0/4

Table 4.2: Computational results for Knapsack Budget Constraint

there is a drastic increment in resolution time of more than 1000 seconds when $\eta = 0.8$. No instance was solved for a budget of 0.8 and 20 nodes for total interdiction. A clear tendency can be seen, where the bigger the budget the less test instances are solved. Unlike the results of cardinality constraint, more test instances are solved when considering partial interdiction in the knapsack case.

4.4 Conclusions

The B&P algorithm is very limited, specially in the cardinality case. When considering a cardinality constraint, [$\text{MSTI}_{\text{SDual}}$] solves instances of up to 50 nodes in Chapter 3 whereas in the current section, the B&P algorithm reached the time limit even in small instances of 15 nodes and with a budget of 5. Two reasons might cause this behavior. First, the solver employed to solve the mathematical formulations handles more efficiently the branching of the variables. In our B&P algorithm, the most traditional branching strategy is used to select the variable on which we branch. According to research in Achterberg (2007), this strategy is generally not significantly better than choosing a variable at random. Second, imposing the branching rules and re-optimizing the [RMP($\hat{\mathcal{P}}$)] is more time con-

suming than solving the original model. Therefore, a B&P algorithm where a decomposition on the variables defining an interdiction plan is not as efficient than solving a MIP of the MSTI problem.

Chapter 5

Benders Decomposition for the MSTI Problem

The numerical experiments of Chapters 3 and 4 indicate the need to move to a different approach to design a resolution method. The MSTI problem becomes more challenging to solve as the size of the instance increases. This is highlighted by the results of previous chapters where instances of 30 nodes or more cannot be solved and the relative gaps between the best known solutions and the upper bounds of even larger instances of our proposed formulations are greater than 50%.

The number of edges of a graph, and thus the number of variables associated with each edge, grows in a quadratic rate as the number of nodes increases. In Chapter 4, we proposed an B&P algorithm that is initialized with a restricted set of variables. As the algorithm solves an MSTI instance, more variables are generated. In this manner, it is hoped to work with a less amount of variables than the MIPs defined in Chapter 3. However, the computational experiments of Chapter 4 proved that the B&P algorithm is inefficient.

All formulations presented in Chapter 3 also grow quadratically in constraints since some of the families of constraints are defined over the set of edges. A Benders Decomposition (BD) algorithm is a technique that mitigates this issue; it divides a complicated problem into two problems with significantly less constraints than the original problem.

In this chapter, we propose a revised BD algorithm for the MSTI problem. Section 5.1 introduces the BD and gives the Master Problem (MP) and sub-

problem. A family of constraints of the MP is lifted in Section 5.2. A second family of constraints is defined in Section 5.3 to strengthen the MP. Section 5.4 presents two basic procedures to initialize the BD algorithm. Upper and lower bounds are discussed in Section 5.5. The revised BD decomposition is finally presented in 5.6. The separation procedure of optimality cuts as well as the particular implementation in a commercial solver of our algorithm is detailed. The chapter ends with computational experiments in Section 5.7 and conclusions in Section 5.8.

5.1 A Benders Decomposition

J.F. Benders in his seminal paper (Benders (1962)) proposed a decomposition technique that would later become known as Benders Decomposition. This decomposition technique aims to address hard problems that are broken down into two smaller, less complicated problems, namely the Master Problem (MP) and the sub-problem. In the original approach, the latter is necessarily a linear problem. An iterative procedure, consisting mainly of repeatedly solving both problems, is designed in order to converge to the optimal solution of the original problem. The MP is meant to work with a much smaller set of constraints than the original formulation would have, which is expected to decrease the burdensome resolution of the original problem. The reader is referred to Rahmaniani *et al.* (2017) for recent results on the BD algorithm.

The general formulation of the MSTI problem reads as follows:

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} \sum_{e\in\mathcal{E}} (c_e + d_e x_e) y_e \tag{5.1}$$

The interdictor controls variables $\mathbf{x}$ and $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^{|\mathcal{E}|} \mid \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$. The network operator controls variables $\mathbf{y}$ and $\mathcal{Y}$ represent the set of constraints that define the spanning trees of $\mathcal{G}$. Therefore, the follower computes a MST on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with augmented weights $c_e + d_e x_e$. Let $\mathcal{T}$ be the set of points of $\mathcal{Y}$. Our goal is to define a MP that only uses a subset of $\mathcal{T}$ to provide the optimal solution of the MSTI problem. Formulation (5.1) can be restated as follows:

$$\max_{\hat{\mathbf{x}} \in \mathcal{X}} \left\{ \min_{\mathbf{y}} \left\{ \sum_{e \in \mathcal{E}} (c_e + d_e \hat{x}_e) y_e : \mathbf{y} \in \mathcal{Y} \right\} \right\} \tag{5.2}$$

The inner minimization problem of (5.2) is parameterized with $\hat{\mathbf{x}}$ thus the objective function becomes linear. If $\mathcal{G}$ is connected, the search of the MST is always feasible. The optimal solutions of the inner minimization problem corresponds to points of $\mathcal{T}$. Therefore, (5.2) is reformulated as

$$\max_{\hat{\mathbf{x}} \in \mathcal{X}} \min_{T \in \mathcal{T}} \sum_{e \in T} (c_e + d_e \hat{x}_e) \tag{5.3}$$

where T is a spanning tree, i.e. a point of $\mathcal{Y}$. To finally obtain the Benders MP, a new variable z is defined in order to linearize the objective function of (5.3). The MP reads as follows:

$$[\text{MP}(\mathcal{T})] \quad \max_{\mathbf{x}, z} z \tag{5.4a}$$

s.t.

$$z \leq \sum_{e \in T} (c_e + d_e x_e) \qquad \forall T \in \mathcal{T} \tag{5.4b}$$

$$\sum_{e \in \mathcal{E}} b_e x_e \leq B \tag{5.4c}$$

$$x_e \in \{0, 1\} \qquad \forall e \in \mathcal{E} \tag{5.4d}$$

[MP($\mathcal{T}$)] maximizes variable z. Constraints (5.4b) bound z by above with the weights of the spanning trees in $\mathcal{T}$. Therefore, the goal of (5.4) is to find an interdiction plan, which respects the budget constraint (5.4c), such that it increases the weights of the spanning trees of $\mathcal{T}$, thus allowing z to increment its value. Model (5.4) requires the enumeration of all the spanning trees of $\mathcal{G}$. Therefore, we proceed to solve the MP by using a restricted number of spanning trees. Let $\hat{\mathcal{T}}$ be a subset of $\mathcal{T}$ and let us define [MP($\hat{\mathcal{T}}$)] where the spanning trees considered in (5.4b) are only those of $\hat{\mathcal{T}}$.

For each solution $\hat{\mathbf{x}}$ of [MP($\hat{\mathcal{T}}$)], a new tree T can be found by solving the

sub-problem which is defined by the inner problem of (5.2). We denote as [MST-Sub($\hat{\mathbf{x}}$)] the minimization problem of (5.2).

5.2 Lifted Optimality Cuts

The family of constraints (5.4b) are known as optimality cuts or constraints since they do not play a role in the feasibility of the solutions and also cut off non-optimal solutions from the feasible space. In this section, we present a lifted version of these cuts. The concept of feasibility of a spanning tree with respect to an interdiction plan is introduced first.

Definition 5.1. *A spanning tree T of G is feasible with respect to an interdiction strategy $\hat{\mathbf{x}} \in \mathcal{X}$ if interdicted edges are not considered in T if total interdiction is assumed or the edges of T have a weight of $c_e + d_e\hat{x}_e$ if partial interdiction is assumed.*

Note that every spanning tree of G is feasible with respect to at least one interdiction strategy. Such a strategy being the trivial interdiction plan in which no edge is interdicted. Let $\mathcal{G}_{\hat{\mathbf{x}}}$ be a graph where the weight of each edge is $c_e+d_e\hat{x}_e$.

Proposition 5.2. *Let $\hat{\mathbf{x}} \in \mathcal{X}$ be an interdiction strategy and T an edge set that defines a feasible spanning tree with respect to $\hat{\mathbf{x}}$. Let $L = \{f \in \mathcal{E} \mid \hat{x}_f = 1\}$. In addition, L is arbitrarily ordered so that $L = \{f_1, f_2, ..., f_{|L|}\}$. Let $\hat{\mathbf{x}}_i$ be a modified interdiction plan of $\hat{\mathbf{x}}$ where edges x_{f_1}, x_{f_2},..., $x_{f_{i-1}}$ and x_{f_i} are no longer interdicted. Let q_0 be the weight of T and q_i represents the weight of the MST of $\mathcal{G}_{\hat{\mathbf{x}}_i}$ for $i = 1, 2, .., |L|$. Then, the lifted optimality constraints (5.5a) are tighter constraints than optimality constraints (5.4b).*

$$z \leq \sum_{e \in T}(c_e + d_e x_e) + \sum_{f_i \in L}(q_i - q_{i-1})(1 - x_{f_i}) \tag{5.5a}$$

Proof. The second summation of the right-hand side of the lifted optimality cut is a valid upper bound of z since it is the recalculation of the weight of T if the interdiction of some of the edges in $\hat{\mathbf{x}}$ is removed. Since coefficient $q_i - q_{i-1}$ is negative by definition on the interdiction plan $\hat{\mathbf{x}}$, the next inequality follows.

$$z \leq \sum_{e \in T}(c_e + d_e x_e) + \sum_{f_i \in L}(q_i - q_{i-1})(1 - x_{f_i}) \leq \sum_{e \in T}(c_e + d_e x_e) \tag{5.6a}$$

Clearly, the lifted optimality cuts (5.5a) are tighter inequalities than (5.4b) and the proposition is proved. □

The interpretation of the second summation in (5.5a) is the following: if the interdiction of edge f_i were to be released then a decrement of $q_i - q_{i-1}$ units in the current weight of T will occur.

5.3 Set Covering Constraints

A second family of valid inequalities for the MSTI problem is defined by the set covering constraints that are presented in this section.

Let $\mathcal{SC}(v^*) := \{T \in \mathcal{T} \mid \sum_{e \in T} c_e < v^*\}$ be the set that contains the trees of $\mathcal{T}$ whose weight is less than v^*, the optimal value. The set covering constraints are stated as follows:

$$\sum_{e \in T} x_e \geq 1 \qquad \forall T \in \mathcal{SC}(v^*) \tag{5.7a}$$

Constraints (5.7a) ensure the interdiction of at least one edge of each tree that has a weight less than v^*. If there exists a tree $\bar{T}$ with weight $\bar{v}$ such that $\bar{v} < v^*$ and none of its edges are interdicted, then the network operator will select this tree as its optimal solution which contradicts the fact that v^* is the optimal objective value. Clearly, v^* is not known a priori, however constraints (5.7a) are valid for any value $v' \leq v^*$. Therefore, the value of a known solution is used to define $\mathcal{SC}(v')$. If a solution with a better objective value v'' is found, the set $\mathcal{SC}(v')$ is updated to $\mathcal{SC}(v'')$ and more constraints can be defined. This same type of valid inequalities is proposed in Israeli & Wood (2002) for the SPI problem. In Wei *et al.* (2021), the authors give a mathematical formulation for the MSTI problem defined only by set covering inequalities.

5.4 Initialization of $\hat{\mathcal{T}}$ and the initial solution

In a BD algorithm, the MP is initialized with a restricted number of constraints. Then, as the algorithm iterates more constraints are added to the MP. In this section, we present how set $\hat{\mathcal{T}}$ is initialized in the developed BD algorithm.

Two trees are used to initialize $\hat{\mathcal{T}}$. The first one is the MST of $\mathcal{G}$ since this is a spanning tree that must be interdicted to worsen the weight of the MST computed by the follower.

The second spanning tree is obtained by a simple procedure developed to obtain a good feasible solution for the MSTI problem. The latter is built to define the initial solution of our algorithm in order to have a good initial lower bound for the problem and to define the set $\mathcal{SC}(v')$ where v' is the value of this solution. Therefore, two basic procedures, depending on the type of budget constraint, are presented to generate the second spanning tree.

An enumeration algorithm is developed for the cardinality case. Given B edges to be interdicted, all interdiction plans of B edges that contain only edges of T_0 are computed, where T_0 is the MST of $\mathcal{G}$. They are considered one by one to interdict the network and compute the MST in the augmented graph. The best solution found is then added to $\hat{\mathcal{T}}$.

For the knapsack case, a more basic approach is taken. Since in the enumeration algorithm proposed for the cardinality case it is needed to know in advance the number of edges to interdict, the same approach cannot be followed here. Therefore, we provide the solution of the most vital edge as initial solution. The algorithm is straightforward. The $|\mathcal{V}| - 1$ single-edge interdiction plans of T_0 are generated. The one that causes the greatest increase of the weight of the MST is the most vital edge and the associated tree is added to $\hat{\mathcal{T}}$.

5.5 Upper and Lower Bounds of [MP($\hat{\mathcal{T}}$)]

The classical BD algorithm consists of iteratively solving the MP and the sub-problem. [MP($\hat{\mathcal{T}}$)] is initialized with a limited number of trees in $\hat{\mathcal{T}}$, for instance the ones proposed in Section 5.4. The master problem is solved and an interdiction plan $\hat{\mathbf{x}}$ is found. [MST-Sub($\hat{\mathbf{x}}$)] is parameterized with this interdiction plan. A new tree T is then computed by solving the sub-problem. T is subsequently

added to [MP($\hat{\mathcal{T}}$)], i.e. a new optimality cut (5.4b) is added to $\hat{\mathcal{T}}$. The MP is re-optimized and the process is repeated. In this section, we explain that the classical BD algorithm for the MSTI problem is valid and finite.

Since $\hat{\mathcal{T}} \subseteq \mathcal{T}$, [MP($\hat{\mathcal{T}}$)] considers a smaller number of spanning trees than (5.4). Therefore, the former formulation is a relaxation of the latter. For this reason, solutions of [MP($\hat{\mathcal{T}}$)] provide valid upper bounds for the MSTI problem. The solution given by [MST-Sub($\hat{\mathbf{x}}$)] is a valid lower bound for the MSTI problem since it is the optimal response to the interdiction strategy $\hat{\mathbf{x}}$.

At each iteration of the BD algorithm, the best bounds are retrieved. The algorithm ends when both bounds are equal. Given a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the number of spanning trees in $\mathcal{G}$ is $|\mathcal{V}|^{|\mathcal{V}|-2}$ (Bedrosian (1964)). Since there is a finite number of spanning trees in $\mathcal{T}$, the procedure terminates in a finite number of iterations. It is expected that only a subset of $\mathcal{T}$ is generated by the iterative algorithm thus leading to solve a significantly smaller problem than [MP($\mathcal{T}$)].

Based on the procedure above, a revised BD algorithm is presented in the following section where [MP($\hat{\mathcal{T}}$)] is solved only once and lifted optimality cuts as well as set covering constraints are defined dynamically during optimization.

5.6 Revised Benders Decomposition Algorithm

In the previous sections, the main components of the revised BD algorithm proposed in this manuscript were introduced. In this section, it is explained how these components interact. The separation procedures for the optimality cuts and set covering constraints and the implementation of our algorithm are presented.

5.6.1 Enhanced Master Problem

The lifted optimality constraints (5.5a) and the set covering constraints (5.7a) and the tree constraints (4.6) were demonstrated to be valid for the [MP($\hat{\mathcal{T}}$)]. Therefore, an enhanced MP can be defined. Let $\mathcal{L}$ be a set of triplets $(T, L, \mathbf{q})$ where each of its components is defined as in Proposition 5.2. Let $\mathcal{SC}(v') := \{T \in \hat{\mathcal{T}} \mid \sum_{e \in T} c_e < v'\}$ be a collection that contains the edge sets that define spanning trees of $\mathcal{G}$ and whose weights have a value less than v'. In addition, v' is set to be greater than or equal to the weight of T_0. The enhanced MP reads as follows:

$$[\text{MP}(\mathcal{L}, \mathcal{SC}(v'))] \quad \max_{\mathbf{x},z} \; z, \tag{5.8a}$$

s.t.

$$z \leq \sum_{e\in T}(c_e + d_e x_e) + \sum_{f_i \in L}(q_i - q_{i-1})(1 - x_{f_i}), \qquad \forall (T, L, \mathbf{q}) \in \mathcal{L} \tag{5.8b}$$

$$\sum_{e\in T} x_e \geq 1 \qquad \forall T \in \mathcal{SC}(v') \tag{5.8c}$$

$$\sum_{e\in\mathcal{E}} b_e x_e \leq B \tag{5.8d}$$

$$\sum_{e\in T_i} x_e \leq \bar{B} - i \qquad \forall i = 0, 1, ..., \bar{B} \tag{5.8e}$$

$$x_e \in \{0, 1\} \qquad \forall e \in \mathcal{E} \tag{5.8f}$$

In $[\text{MP}(\mathcal{L}, \mathcal{SC}(v'))]$, constraints (5.4b) are replaced by their lifted version (5.8b). Set covering constraints (5.8c) are added. Finally, the tree constraints (5.8e), defined in Section 4.2.2.1 for the pricing problem, are as well considered for the enhanced MP. The latter family of constraints are also valid for $[\text{MP}(\mathcal{L}, \mathcal{SC}(v'))]$ since they cut-off non-optimal interdiction strategies.

5.6.2 Separation Procedure for the Lifted Optimality Cuts

The lifted optimality cuts defined by Proposition 5.2 require a specific setting of set L and vector $\mathbf{q}$ in order to be valid. In the following, a procedure that determines both parameters is presented. For any edge set $\mathcal{E}'$, let $w(\mathcal{E}') = \sum_{e\in\mathcal{E}'} c_e$ and T_0 the MST of $\mathcal{G}$. Let T be the MST given by [MST-Sub($\hat{\mathbf{x}}$)] for a given interdiction plan $\hat{\mathbf{x}}$. Let q_0 denote the weight of T. Let $\mathcal{I}_{\hat{\mathbf{x}}}$ be the set of edges that have a positive component in vector $\hat{\mathbf{x}}$. Assume that each time an element is added to L, it is arbitrarily ordered so that $L = \{f_1, f_2, ..., f_{|L|}\}$ where $f_{|L|}$ is the last element added. Algorithm 3 presents a procedure which lifts constraints (5.4b).

Algorithm 3 has as input a pair $(\hat{\mathbf{x}}, T)$, that is an interdiction strategy with its respective spanning tree that is the optimal reaction of the follower. In Step 2, a graph $\mathcal{G}'$ with respect to $\hat{\mathbf{x}}$ is defined by removing all interdicted edges, i.e. all edges associated with a non zero x_e value in $\hat{\mathbf{x}}$ are removed from $\mathcal{G}$. A loop on

Algorithm 3 Separation Procedure for the Lifted Optimality Cuts

Require: A pair $(\hat{\mathbf{x}}, T)$
Ensure: A triplet $(T, L, \mathbf{q})$ where each component is as in Proposition 5.2
1: $i \leftarrow 1$; $L \leftarrow \emptyset$; $\mathbf{q} \leftarrow \mathbf{0}$
2: $\mathcal{G}' \leftarrow \mathcal{G} \setminus \mathcal{I}_{\hat{\mathbf{x}}}$
3: $L' \leftarrow T_0 \setminus T$
4: **while** $L' \neq \emptyset$ **do**
5: Select $f \in L'$
6: $\mathcal{G}' \leftarrow \mathcal{G}' \cup \{f\}$
7: Calculate MST T_i of $\mathcal{G}'$
8: $q_i \leftarrow w(T_i)$
9: $L \leftarrow L \cup \{f\}$
10: $L' \leftarrow L' \setminus \{f\}$
11: $i \leftarrow i + 1$
12: **end while**
13: Return $(T, L, \mathbf{q})$

set L', which characterizes the edges of T_0 that do not appear in T, is performed from Steps 4 to 12. Set L' is composed of edges of T_0 that were not selected in T since those edges will cause a decrement on the weights of the trees of $\mathcal{G}'$ when they are added to $\mathcal{G}'$. Since $|L'| \leq |V| - 1$, Algorithm 3 ends in at most $|V| - 1$ iterations. The loop consists in selecting an edge $f \in L'$. Then the graph is augmented by adding this edge to $\mathcal{G}'$ and the MST is recalculated and its weight is retrieved. Finally, we include f in L.

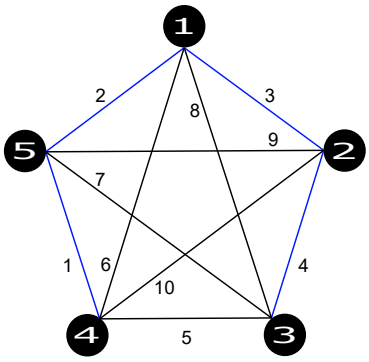
The output of Algorithm 3 is a triplet $(T, L, \mathbf{q})$. Due to the assumption stated above, set L is a rearrangement of the edges of L' to match the weights of the trees in $\mathbf{q}$. Note that $q_i \leq q_{i-1}$ for $i = 1, ..., |L|$ since, at each iteration of the loop of the Algorithm 3, the edges which belong to T_0 are added to $\mathcal{G}'$; the weight of the new tree is thus necessarily equal or smaller. For each pair $(\hat{\mathbf{x}}, T)$, a triplet $(T, L, \mathbf{q})$ is calculated by Algorithm 3 which is subsequently added to a collection $\mathcal{L}$.

Figure 5.1 presents a illustrative example of the lifting procedure. The initial graph and T_0 is shown in Figure 5.1a where the blue edges represent T_0. The weight of this tree is 10. By assuming total interdiction, a cardinality constraint and a budget $B = 2$, the graph in Figure 5.1b is obtained. $\mathcal{G}'$ is defined by the black and blue edges, dotted lines represent $\hat{\mathbf{x}}$ and the blue edges represent T which is the MST with respect to $\hat{\mathbf{x}}$. Scalar q_0 is 19, i.e. the weight of T. Thus,

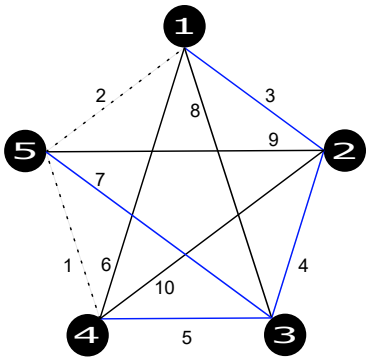
$L' = \{\{1,5\},\{4,5\}\}$. We select an edge of L', in this case $\{4,5\}$. Edge $\{4,5\}$ is added to $\mathcal{G}'$ and the MST is computed. The resulting graph is shown in 5.1c. The weight of the new tree is 13, thus $q_1 = 13$. The same procedure is repeated for edge $\{1,5\}$. The initial graph is obtained, i.e. the one shown in Figure 5.1a. Therefore, $q_2 = 10$. Now, it is possible to define the optimality constraint (5.5a) with respect to this example as:

$$\begin{aligned} z &\leq \sum_{e \in T}(c_e + d_e x_e) + (13-19)(1-x_{45}) + (10-13)(1-x_{12}) \\ &= \sum_{e \in T}(c_e + d_e x_e) - 6(1-x_{45}) - 3(1-x_{12}) \end{aligned}$$

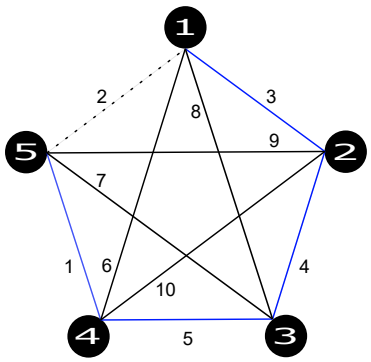
Therefore, a decrement of 6 units in the weight of T is obtained if x_{45} is not interdicted. Analogously, the weight of T diminishes by three units if x_{12} is not interdicted.



(a) Initial graph



(b) A MSTI problem solution where $B = 2$



(c) Tree when interdiction of x_{45} is released

Figure 5.1: Example of lifting procedure

5.6.3 Implementation of the Revised BD Algorithm

In a classical BD algorithm, the MP and the sub-problem are iteratively solved to optimality, one after the other, to converge to the optimal solution of the original problem. In our revised algorithm, $[\text{MP}(\mathcal{L}, \mathcal{SC}(v'))]$ is solved only once via a Branch-and-Cut algorithm managed by a commercial optimizer where constraints (5.8b) and (5.8c) are dynamically added via user cut and lazy constraint callbacks. Both user cut and lazy constraint callbacks allow to add user-defined cuts at each node of the search tree.

A lazy constraint callback is a procedure that adds constraints that define the feasible space of a problem. Those constraints are too many to include all at once at the beginning of the optimization process. These are exactly constraints (5.8b) originally defined with respect to $\mathcal{T}$. The lazy constraint callback is called when an integer interdiction plan $\hat{\mathbf{x}}$ is found. The procedure generates a constraint (5.8b) and checks whether this cut is violated by the integer solution. If yes, the constraint is added to the full model. In addition, a constraint (5.8c) is defined, and added to the model when it is violated. If constraint (5.8b) is not violated, the candidate solution becomes the new incumbent of the problem.

When solving a node in a Branch-and-Cut algorithm, problem-specific cuts are added through user cut callbacks that tighten the LP relaxation of the node. User cut callbacks mainly separate fractional solutions, i.e. they try to find a constraint that violates this solution to strengthen the LP relaxation. In this case, constraints (5.8b) are defined to improve the relaxation. If the lifted optimality cut is violated, a set covering constraint is generated too. If no violated cuts are found, the node is solved and a new node is treated or the algorithm stops.

The classical BD algorithm is implicit in the optimization process explained above. The lazy constraint callback plays the role of the sub-problem. When an integer solution candidate is found, it means that this solution is optimal for the current $[\text{MP}(\mathcal{L}, \mathcal{SC}(v'))]$ and a new cut that violates the solution is added, just as in a classical Benders Decomposition. Algorithm 4 presents a pseudo-code of our revised BD algorithm.

$[\text{MP}(\mathcal{L}, \mathcal{SC}(v'))]$ is implemented in a commercial solver with an initial number of elements in $\mathcal{L}$ and $\mathcal{SC}(v')$. In Step 1 of Algorithm 4, $\mathcal{L}$ is initialized with the triplet corresponding to the MST. $[\text{MP}(\mathcal{L}, \mathcal{SC}(v'))]$ has to be initialized with at least the lifted optimality constraint with respect to T_0. Since T_0 is the MST

Algorithm 4 Revised Benders Decomposition for the MSTI problem

Require: An instances of a MSTI problem
Ensure: Optimal interdiction plan $\mathbf{x}$ with its associated MST $\mathbf{y}$

1: $\mathcal{L} \leftarrow \{(T_0, \emptyset, \mathbf{q})\}$ where $\mathbf{q} \in \mathbf{0}^{|L|+1}$
2: Generate feasible solution $(\hat{\mathbf{x}}', T')$ as in Section 5.4
3: Compute $(T', L, \mathbf{q})$ using Algorithm 3
4: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(T', L, \mathbf{q})\}$
5: $\mathcal{SC}(v') \leftarrow \emptyset$ where $v' = w(T')$
6: Provide $\hat{\mathbf{x}}'$ as initial solution to the solver
7: **if** Time limit is not met **then**
8: (Re)Optimize [MP$(\mathcal{L}, \mathcal{SC}(v'))$] for $\hat{\mathbf{x}}$
9: **if** Solver calls a callback **then**
10: In a Control Callback:
11: Obtain T by solving [MST-Sub$(\hat{\mathbf{x}})$]
12: Compute $(T, L, \mathbf{q})$ using Algorithm 3
13: **if** (5.5a) is violated **then**
14: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(T, L, \mathbf{q})\}$
15: **if** $w(\mathcal{T}) < v'$ **then**
16: $\mathcal{SC}(v') \leftarrow \mathcal{SC}(v') \cup \{T\}$
17: **end if**
18: **end if**
19: Go to Step 8
20: **end if**
21: **end if**
22: Compute $\mathbf{y}$ with respect to $\mathbf{x}$
23: Return $(\mathbf{x}, \mathbf{y})$

of $\mathcal{G}$ or in other words T_0 is the optimal reaction of the follower to the trivial interdiction strategy, no non-empty edge set L induces a decrement of the weight of T_0. Thus, $L = \emptyset$ and $\mathbf{q} \in \mathbf{0}^{|L|+1}$. In Steps from 2 to 6, one of the procedures for either cardinality or knapsack constraint presented in Section 5.4 is invoked to generate a feasible initial solution T' in order to define a second optimality cut. Then, we update $\mathcal{SC}(v')$ with $v' = w(T')$ and provide $\hat{\mathbf{x}}'$ as an incumbent solution to the solver. A lower bound for the MSTI problem is automatically computed by the solver form the incumbent solution. Adding a second triplet $(T', L, \mathbf{q})$ to $\mathcal{L}$ is optional. However, the latter is done in the hope of fathoming nodes of the search tree if a strong lower bound is provided.

The optimization via the Branch-and-Cut algorithm is launched in Step 8. During optimization, the solver can call a control callback, either user cut or lazy

constraint callback, to tighten the LP relaxation of a given node or to check if a candidate integer solution can become the new incumbent of the problem. The latter is declared in Step 9. From step 10 to 19, additional constraints (5.8b) and (5.8c) are generated and added by both callbacks. The user cut and lazy constraint callback procedures are implemented in the same manner. Both use a non-optimal solution $\hat{\mathbf{x}}$ (either integer or fractional) to parameterize the [MST-Sub($\hat{\mathbf{x}}$)]. The sub-problem is solved to obtain a spanning tree T. Then, the pair $(\hat{\mathbf{x}}, T)$ is given as input to Algorithm 3 in order to define a lifted optimality constraint. If the lifted optimality constraint with respect to T is violated, it is added to $\mathcal{L}$. A tolerance of 10% on the violation of (5.5a) with respect to its right-hand side is considered for constraints generated by fractional solutions. This is, a violated cut is only added to the collection $\mathcal{L}$ if z is greater than or equal to 110% percent of the right-hand side value of constraint (5.5a). The latter is done in the hope of speeding up the resolution of the problem by avoiding the inclusion of violated cuts to the model that will not greatly improve the upper bound of a Branch-and-Bound node.

The value of the incumbent integer solution of the initial solution or from another solution found during the optimization process is denoted by v'. In Step 15, v' is compared to the weight of T. If this spanning tree has a smaller weight than the incumbent value, it is added to the collection $\mathcal{SC}(v')$. The value v' is automatically updated by the solver when a new incumbent solution is found. Regardless that constraints where added or not by a control callback, the procedure re-optimizes the model as stated in Step 19. Algorithm 4 jumps to Step 22 when the solver has found the optimal solution or the time limit has been met according to the conditions of Step 7. Vector $\mathbf{y}$ is computed in this step since [MP($\mathcal{L}, \mathcal{SC}(v')$)] only provides the optimal plan $\mathbf{x}$. The computation of $\mathbf{y}$ is easily done by solving [MST-Sub($\mathbf{x}$)]. Finally, the algorithm ends by returning the optimal solution $(\mathbf{x}, \mathbf{y})$ for the MSTI problem.

The main advantage of the Algorithm 4 is that the solver handles the branching of variables with respect to the criteria it finds the most advantageous and which are usually complicated for the user to identify. Note that the termination condition is also imposed by the solver, the optimization process will not stop until the branching is completed and all the nodes of the search tree are examined or the time limit is met.

5.7 Computational Experiments

This section is divided into to parts. First, the generation of the test instances is explained. Four different types of instances are considered where the main differences are in the magnitude of the weights of the edges or the sparsity of the graph. This is explained in Section 5.7.1. The second part consists in the numerical results of our revised BD algorithm and are reported in Section 5.7.2. The results are discussed and conclusions are then given.

5.7.1 Instance Generation

A total of 408 graphs of different sizes and densities for our numerical experiments are proposed. We test, on each graph, both types of interdiction and both types of budget constraints. In addition, we vary the budget size for each setup. In total, 6744 instances, which include those of Chapters 3 and 4, are considered to determine the efficiency of the BD algorithm. The details of the instance data are as follow.

We classify the graphs into four types, namely Types 1 to 4. Type 1 instances are complete graphs, i.e. $|\mathcal{E}| = |\mathcal{V}|(|\mathcal{V}| - 1)/2$. The weights c_e are randomly assigned from 1 to $|\mathcal{E}|$ to have edges with different weights. The latter can be done without loss of generality. Interdiction costs b_e for each $e \in \mathcal{E}$ are randomly generated and uniformly distributed in $[1, |\mathcal{E}|]$. Two different interdiction augmentations are defined, one for total interdiction and the other one for partial interdiction. For each edge $e \in \mathcal{E}$, we define $d_e = |\mathcal{E}| - c_e + 1$ for total interdiction. For partial interdiction, d_e is randomly generated and according to a uniform distribution in $[1, |\mathcal{E}| - c_e + 1]$.

The instances considered in the sections of computational experiments of Chapters 3 and 4 are what we refer to as Type 2 instances. The weights are randomly generated and uniformly distributed in $[1, 100]$. Without loss of generality, an arbitrary total order is proposed for edges of the same weight. Similarly as in Type 1 instances, the interdiction augmentation for total interdiction is defined as $d_e = \max_{f \in \mathcal{E}} c_f - c_e + 1$. For partial interdiction, a random value between $[1, \max_{f \in \mathcal{E}} c_f - c_e + 1]$ is generated. Last, the interdiction costs are randomly generated and uniformly distributed in $[1, 100]$.

Instances of Type 1 and 2 are complete graphs. We extend our experimentation to sparse graphs and retrieve Type 3 instances proposed in Wei *et al.* (2021). The density of these networks ranges from 3% to 10%. Their instances contain fractional weights and interdiction costs randomly generated in $[1, 100]$. We round up these values and also generate the remaining parameters, i.e. the interdiction augmentations for both total and partial interdiction, as in instances of Type 2.

The parameters of the last set of instances, Type 4, are generated exactly as those of Type 2 instances. The difference between both of them lies in the number of edges of the network. The generated graphs of Type 4 have a density between 50% and 70%.

A total of 44 graphs were created for each of the instance Types 1, 2, and 4. The size of the graphs are $|\mathcal{V}| = 20, 25, 30, 50, 75, 100, 125, 150$ and 200. There are 300 instances of Type 3 that were modified as previously stated. They include 40, 80, 120, 160, and 200 nodes.

The budget considered depends on the size of the network and the type of budget constraint. For a cardinality constraint, budgets of 3, 5, 7, 8 and 9 are set. The budgets in the knapsack case are determined using the largest edge weight of the network. A percentage of the largest c_e is computed and used as B. These fractions are $\eta = 0.2$, 0.4, 0.6, 0.8, 1 and 1.2. Since budgets are impartially set for all instances of Type 1, 2 and 4, it may happen that the allocated budget is large enough to disconnect the network. If it occurs, the instance is disregarded.

Budgets for the cardinality case are different for Type 3 instances since those networks have a much smaller density than the networks of other type of instances. Therefore, budgets of 2, 3 and 4 are considered for the cardinality case. For the knapsack case, we consider the same budgets as in the other instance types. However, as in Type 4 instances, the allocated budget can be large enough to disconnect the graph. Again, these instances are discarded in this case.

5.7.2 Computational Results

The efficiency of Algorithm 4 for the MSTI problem is assessed in this section. All procedures and algorithms are implemented in C++. CPLEX 12.9 is the commercial solver used to solve the MP and to implement the callback procedures. Experiments are conducted on a 64-bit workstation with a 2.20GHz Intel(R) Xeon(R) CPU E5-2630 v4 processor and 64 GB of RAM memory. A time limit

Config.		Total Interdiction - Cardinality Budget Constraint											
		Type 1				Type 2				Type 4			
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
20	3	0.2	128.8	0.00	4/4	0.1	151.5	0.00	4/4	0.2	195.3	0.00	4/4
	5	6.1	1973.8	0.00	4/4	5.4	2066.8	0.00	4/4	5.6	1948.3	0.00	3/3
	7	399.6	19552.8	0.00	4/4	321.2	16679.3	0.00	4/4	323.8	19605.5	0.00	2/2
	9	3171.4	35437.0	11.97	1/4	2550.2	38385.8	9.59	2/4	3600.0	52899.0	7.44	0/1
25	3	0.2	197.0	0.00	4/4	0.3	301.0	0.00	4/4	0.3	284.5	0.00	4/4
	5	10.5	3120.3	0.00	4/4	7.2	1512.0	0.00	4/4	24.1	3778.8	0.00	4/4
	7	843.2	20619.8	0.00	4/4	179.8	10131.5	0.00	4/4	702.6	22477.3	0.00	4/4
	8	2892.3	27151.0	6.80	2/4	1581.6	27425.5	0.81	3/4	2495.4	39200.7	2.63	2/3
30	3	0.6	499.0	0.00	4/4	0.7	826.0	0.00	4/4	0.5	455.3	0.00	4/4
	5	43.6	9001.3	0.00	4/4	31.0	5288.5	0.00	4/4	13.0	3705.5	0.00	4/4
	7	3258.1	39565.5	3.81	1/4	2372.3	32272.3	3.03	2/4	1688.7	32093.3	1.57	3/4
50	3	3.5	3009.8	0.00	4/4	2.1	1409.8	0.00	4/4	4.0	2623.0	0.00	4/4
	5	386.9	24758.8	0.00	4/4	229.7	16076.0	0.00	4/4	311.7	18363.0	0.00	4/4
	7	2922.1	16936.5	9.71	1/4	3600.0	23528.3	11.27	0/4	3406.1	23281.5	8.33	1/4
75	3	7.0	1748.3	0.00	4/4	8.4	5692.0	0.00	4/4	12.6	6857.3	0.00	4/4
	5	329.9	16042.8	0.00	4/4	437.3	89611.5	0.00	4/4	1471.8	92709.3	0.00	4/4
	7	3600.0	16526.3	7.85	0/4	3600.0	47200.0	5.28	0/4	3600.0	17688.0	8.30	0/4
100	3	55.2	17399.0	0.00	4/4	759.4	66121.8	0.00	4/4	102.1	32823.8	0.00	4/4
	5	1904.6	98721.3	0.53	3/4	3600.0	357069.8	2.37	1/4	3056.3	261785.5	1.53	2/4
	7	3600.0	16776.0	7.88	0/4	3600.0	42201.3	7.35	0/4	3600.0	24573.3	8.08	0/4

Table 5.1: Computational results for Type 1, 2 and 4 instances under total interdiction and cardinality budget constraint.

of one hour (3600 seconds) is imposed for each experiment. Wherever a MST needs to be computed, Kruskal's algorithm is used. For each instance, the MSTI problem is defined over the network $\mathcal{G}_{\bar{B}}$, i.e. the $\bar{B}$-edge connected certificate. The certificate, which depends on the budget, is a reduction of the network that prunes edges that are not necessary to solve the problem. When considering a cardinality budget constraint, we set $\bar{B} = B$. In the case of the knapsack constraint, $\bar{B} = \max\{\sum_{e\in\mathcal{E}} x_e \mid \sum_{e\in\mathcal{E}} b_e x_e \leq B\}$. The resulting certificate when considering a knapsack constraint is equivalent to the original network $\mathcal{G}$ since $\bar{B}$ results in a large number.

The tables with the computational results contain the following data. The left column *Config.* is the configuration of each test instance, i.e. the number of nodes $|\mathcal{V}|$ and the budget B. Each reported value is an average on the number of instances considered. The reported data are detailed in the following. *Time*, in seconds, is the resolution time of the tested instances. *Nodes* is the number of nodes in the search tree of the Branch-and-Cut algorithm. *Gap* is the percentage of the relative gap between the upper bound and the best known solution at the end of the algorithm when an instance has been solved optimally or has reached the time limit. Finally, *Opt/Tot* represents the number of optimally solved instances over the total number of instances tested.

Numerical results considering total interdiction and cardinality budget constraint are reported in Table 5.1. Note that for some instances of the Type 4

column, there are less than four instances tested. For some of these instances, the allocated budget is large enough to disconnect the networks due to their sparsity. Therefore, these instances are omitted. For each instance size, our BD algorithm is able to solve all instances with the smallest budget $B = 3$. The CPU time is in less than one minute with the exceptions of Type 2 and Type 4 instances in 100 size instances. As the budget increases, the instances becomes more challenging to solve which is supported by the large search tree sizes for the larger values of B. For instances of size 20, only one instance of Type 1, two of Type 2 and none of Type 4 with a budget of 9 are solved. The average gap over the three types is of 9.66%. The same remark is valid for the other sizes, none of the instances of size 75 and 100 with $B = 7$ are optimally solved. On the other hand, our algorithm is able to solve instances of 75 nodes or less when $B \leq 5$ in less than 600 seconds; with the exception of 75 node instances of Type 4, where the time increased to nearly thirty minutes. For these instances, the time ranges from a few seconds for the smallest instances to 437.3 seconds for instances of size 75 except for the instances of Type 4 where the average time increased to 1471.8 seconds. Type 2 instances are the testbed of the computational experimentation of Chapter 3. The results obtained by the BD algorithm are fairly better than those obtained by the mathematical formulations for this setup. In this section, instances of 30 nodes and a budget of 5 are solved in 31 seconds. [$\text{MSTI}_{\text{Dual}}$] solves the same instances in more than 2000 seconds. Similar comparisons can be done for instances including 20, 25 and 50 nodes.

Table 5.2 shows experimentation with the same configuration as Table 5.1 but considering partial interdiction. The results are slightly different from total interdiction. The BD algorithm appears to efficiently solve instances up to 30 nodes in size with all allocated budgets in less than 147.2 seconds. Instances of 50 nodes with 3 and 5 of budget are solved optimally, but with a time increase of at least five times over smaller sizes. From this point on, the efficiency of the proposed algorithm starts to decrease. For $B = 7$, only 5 out of 12 instances are solved. For instances of 75 nodes and $B = 3$, we can still obtain the optimal solution in less than four minutes. Thereafter, only a few instances are solved. Note that in the Type 4 column, more instances are solved compared to the other columns, which is explained by the sparsity of those networks. Type 2 instances solved by the BD algorithm provides again better results for these assumptions

Config.		Partial Interdiction - Cardinality Budget Constraint											
		Type 1				Type 2				Type 4			
$\|V\|$	B	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
20	3	0.1	62.3	0.00	4/4	0.2	144.0	0.00	4/4	0.1	34.3	0.00	4/4
	5	0.5	534.5	0.00	4/4	0.8	793.8	0.00	4/4	0.2	369.8	0.00	4/4
	7	1.9	1793.8	0.00	4/4	4.1	2639.3	0.00	4/4	0.4	827.3	0.00	4/4
	9	11.6	6419.5	0.00	4/4	20.3	7315.8	0.00	4/4	1.1	1880.3	0.00	4/4
25	3	0.4	270.8	0.00	4/4	0.5	484.3	0.00	4/4	0.2	182.0	0.00	4/4
	5	2.3	1818.8	0.00	4/4	2.5	1606.3	0.00	4/4	0.8	987.8	0.00	4/4
	7	16.1	5335.5	0.00	4/4	9.7	4393.8	0.00	4/4	3.0	2472.5	0.00	4/4
	8	33.4	8913.8	0.00	4/4	28.9	7788.0	0.00	4/4	22.2	9988.3	0.00	4/4
30	3	0.8	451.8	0.00	4/4	1.5	940.3	0.00	4/4	0.3	292.8	0.00	4/4
	5	7.2	3734.5	0.00	4/4	12.7	2897.5	0.00	4/4	1.4	1447.8	0.00	4/4
	7	45.3	9304.3	0.00	4/4	147.2	11212.0	0.00	4/4	10.6	6086.0	0.00	4/4
50	3	20.2	3518.5	0.00	4/4	25.6	2923.8	0.00	4/4	9.2	3138.0	0.00	4/4
	5	723.3	16142.3	0.00	4/4	659.4	24700.3	0.00	4/4	479.6	32378.3	0.00	4/4
	7	2842.8	29175.0	5.56	2/4	3143.5	30536.0	8.65	1/4	1886.4	38771.0	2.53	2/4
75	3	224.8	8392.0	0.00	4/4	225.9	15195.5	0.00	4/4	66.1	13212.0	0.00	4/4
	5	3209.0	17356.5	4.02	1/4	3424.8	37458.8	4.24	1/4	1700.3	115594.25	0.00	4/4
	7	3600.0	6782.5	14.86	0/4	3600.0	10600.0	11.77	0/4	3600.0	42362.3	6.65	0/4
100	3	2663.0	33777.0	2.26	2/4	3455.4	103914.0	1.23	1/4	1371.9	59859.0	1.15	3/4
	5	3600.0	3387.3	15.10	0/4	3600.0	7752.0	10.07	0/4	3165.1	54111.0	4.67	1/4
	7	3600.0	1456.3	21.71	0/4	3600.0	1697.5	17.12	0/4	3600.0	14173.8	10.89	0/4

Table 5.2: Computational results for Type 1, 2 and 4 instances under partial interdiction and cardinality budget constraint.

of type of interdiction and budget constraint with respect to the models. The difference in computational time is in some cases greater than 1000 seconds.

In Table 5.3, we report the results obtained when we consider the total interdiction case and a knapsack budget constraint. The testbed is extended to consider networks including 125, 150 and 200 nodes. Most instances are solved optimally. The first unsolved instances appear for networks with 100 nodes and with the highest budget. Similarly, larger size instances are not solved with budgets of 0.8 or greater which marks the limits of our algorithm. For every size and $\eta = 0.2$, the time is of less than 31.1 seconds. Graphs of 50 nodes or less have a resolution time of less than 175 seconds, the only exception is a 75 nodes Type 1 instance and $\eta = 1.2$ with a time of 669.8. For 75 nodes instances, the average resolution time is less than 132.9 with $\eta \leq 0.8$. Starting with networks of 75 nodes and the largest budgets, the resolution times increase to more than a thousand seconds. The best results are obtained for Type 4 instances where only two instances of size 125 and six of size 200 are not solved. This again can be explained by the sparsity of these networks that reduces the combinatorics of the problem. The computational results obtained by solving instances of Type 2 with the mathematical models are not better than those reported in this section. However, the differences in computational time are not very large. The exceptions are instances of 50 nodes that could not be solved for the larger budgets.

Table 5.4 presents the results for partial interdiction and knapsack constraint.

		Total Interdiction - Knapsack Budget Constraint											
Config.		Type 1				Type 2				Type 4			
$\|\mathcal{V}\|$	η	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
20	0.2	0.0	0.3	0.00	4/4	0.0	0.0	0.0	4/4	0.0	0.0	0.00	4/4
	0.4	0.1	6.3	0.00	4/4	0.1	1.5	0.0	4/4	0.1	2.0	0.00	4/4
	0.6	0.2	19.0	0.00	4/4	0.1	15.8	0.0	4/4	0.1	6.5	0.00	4/4
	0.8	0.3	75.5	0.00	4/4	0.2	64.5	0.0	4/4	0.1	45.0	0.00	4/4
	1.0	0.5	195.0	0.00	4/4	0.3	193.8	0.0	4/4	0.2	137.3	0.00	4/4
	1.2	1.0	359.5	0.00	4/4	0.5	385.5	0.0	4/4	0.2	237.5	0.00	4/4
25	0.2	0.0	0.0	0.00	4/4	0.0	0.3	0.0	4/4	0.0	0.0	0.00	4/4
	0.4	0.1	1.5	0.00	4/4	0.1	2.0	0.0	4/4	0.1	4.0	0.00	4/4
	0.6	0.4	31.0	0.00	4/4	0.2	7.5	0.0	4/4	0.2	32.0	0.00	4/4
	0.8	0.7	104.5	0.00	4/4	0.5	64.3	0.0	4/4	0.3	80.5	0.00	4/4
	1.0	1.2	324.8	0.00	4/4	0.7	224.8	0.0	4/4	0.5	395.8	0.00	4/4
	1.2	3.0	692.5	0.00	4/4	1.7	545.5	0.0	4/4	0.7	507.3	0.00	4/4
30	0.2	0.0	0.0	0.00	4/4	0.0	0.0	0.0	4/4	0.0	0.0	0.00	4/4
	0.4	0.1	8.8	0.00	4/4	0.1	9.0	0.0	4/4	0.1	4.3	0.00	4/4
	0.6	0.7	76.3	0.00	4/4	0.3	69.8	0.0	4/4	0.4	35.8	0.00	4/4
	0.8	1.4	197.8	0.00	4/4	0.9	260.8	0.0	4/4	0.7	178.8	0.00	4/4
	1.0	4.1	673.8	0.00	4/4	2.2	681.3	0.0	4/4	1.0	513.8	0.00	4/4
	1.2	9.0	1350.8	0.00	4/4	7.3	1561.3	0.0	4/4	2.7	1181.0	0.00	4/4
50	0.2	0.3	0.0	0.00	4/4	0.1	0.0	0.0	4/4	0.2	2.0	0.00	4/4
	0.4	1.2	35.0	0.00	4/4	0.6	14.0	0.0	4/4	0.7	33.0	0.00	4/4
	0.6	3.2	227.0	0.00	4/4	3.5	220.0	0.0	4/4	1.7	218.8	0.00	4/4
	0.8	12.1	908.0	0.00	4/4	11.6	1188.0	0.0	4/4	5.9	1269.8	0.00	4/4
	1.0	42.4	3482.0	0.00	4/4	37.3	1789.0	0.0	4/4	14.0	2062.5	0.00	4/4
	1.2	669.8	10052.3	0.00	4/4	174.2	5395.5	0.0	4/4	80.6	5339.5	0.00	4/4
75	0.2	0.7	2.8	0.00	4/4	0.3	0.8	0.0	4/4	0.3	11.8	0.00	4/4
	0.4	4.7	76.3	0.00	4/4	1.8	53.0	0.0	4/4	1.7	25.8	0.00	4/4
	0.6	19.8	567.5	0.00	4/4	16.9	582.3	0.0	4/4	8.8	561.0	0.00	4/4
	0.8	119.7	2909.0	0.00	4/4	132.9	4189.0	0.0	4/4	47.6	4690.5	0.00	4/4
	1.0	768.6	9638.8	0.00	4/4	675.0	13464.0	0.0	4/4	128.7	8728.0	0.00	4/4
	1.2	2405.7	14672.8	4.67	3/4	1896.4	18948.3	3.6	3/4	429.4	22061.3	0.00	4/4
100	0.2	2.9	12.3	0.00	4/4	2.4	23.3	0.0	4/4	1.8	11.5	0.00	4/4
	0.4	19.8	172.0	0.00	4/4	20.7	393.5	0.0	4/4	7.8	201.8	0.00	4/4
	0.6	141.4	3136.8	0.00	4/4	153.2	5856.0	0.0	4/4	42.2	2654.8	0.00	4/4
	0.8	953.6	15976.3	0.00	4/4	1268.2	42886.5	0.0	4/4	201.5	11760.8	0.00	4/4
	1.0	3502.9	12813.3	14.89	1/4	1960.7	28347.0	0.7	3/4	1387.1	43205.8	0.00	4/4
125	0.2	5.6	6.5	0.00	4/4	5.6	18.8	0.0	4/4	3.1	10.3	0.00	4/4
	0.4	59.4	429.3	0.00	4/4	56.9	670.3	0.0	4/4	15.0	159.0	0.00	4/4
	0.6	495.2	10465.8	0.00	4/4	298.8	5746.8	0.0	4/4	138.6	3601.3	0.00	4/4
	0.8	3208.1	17081.3	2.12	1/4	2427.0	26676.0	1.8	3/4	859.3	23457.8	0.00	4/4
	1.0	3600.0	4.3	28.23	0/4	3600.0	1456.3	18.6	0/4	2336.7	20495.5	4.54	2/4
150	0.2	17.6	55.0	0.00	4/4	7.9	14.8	0.0	4/4	5.9	126.0	0.00	4/4
	0.4	142.8	698.5	0.00	4/4	106.3	417.5	0.0	4/4	53.2	832.8	0.00	4/4
	0.6	2013.9	57984.8	0.01	4/4	744.5	13593.3	0.0	4/4	344.4	11272.3	0.00	4/4
	0.8	3600.0	499.3	19.33	0/4	3600.0	30113.5	5.7	1/4	1957.7	55260.5	0.00	4/4
200	0.2	31.1	24.8	0.00	4/4	29.6	74.5	0.0	4/4	19.3	187.3	0.00	4/4
	0.4	698.4	1056.5	0.00	4/4	394.7	1423.0	0.0	4/4	290.0	3101.8	0.00	4/4
	0.6	3600.0	650.0	13.25	0/4	3271.7	16141.3	4.7	1/4	3065.5	42554.0	1.85	2/4
	0.8	3600.0	0.0	32.58	0/4	3600.0	0.0	16.7	0/4	3600.0	802.3	12.24	0/4

Table 5.3: Computational results for Type 1, 2 and 4 instances under total interdiction and knapsack budget constraint.

Partial Interdiction - Knapsack Budget Constraint													
Config.		Type 1				Type 2				Type 4			
$\|\mathcal{V}\|$	η	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
20	0.2	0.0	0.0	0.00	4/4	0.0	0.0	0.00	4/4	0.0	0.0	0.00	4/4
	0.4	0.0	4.8	0.00	4/4	0.0	8.8	0.00	4/4	0.0	7.8	0.00	4/4
	0.6	0.1	28.3	0.00	4/4	0.1	21.8	0.00	4/4	0.0	9.5	0.00	4/4
	0.8	0.1	51.8	0.00	4/4	0.1	47.3	0.00	4/4	0.1	53.3	0.00	4/4
	1.0	0.1	87.5	0.00	4/4	0.1	154.0	0.00	4/4	0.1	98.8	0.00	4/4
	1.2	0.2	193.0	0.00	4/4	0.2	254.5	0.00	4/4	0.1	187.0	0.00	4/4
25	0.2	0.0	0.0	0.00	4/4	0.0	1.8	0.00	4/4	0.0	0.0	0.00	4/4
	0.4	0.1	5.5	0.00	4/4	0.1	7.0	0.00	4/4	0.0	5.0	0.00	4/4
	0.6	0.1	30.3	0.00	4/4	0.1	10.8	0.00	4/4	0.1	26.8	0.00	4/4
	0.8	0.2	80.8	0.00	4/4	0.2	49.5	0.00	4/4	0.1	62.5	0.00	4/4
	1.0	0.4	377.0	0.00	4/4	0.3	173.3	0.00	4/4	0.1	134.3	0.00	4/4
	1.2	0.5	593.3	0.00	4/4	0.3	324.3	0.00	4/4	0.2	328.8	0.00	4/4
30	0.2	0.0	0.3	0.00	4/4	0.0	0.0	0.00	4/4	0.0	0.3	0.00	4/4
	0.4	0.1	5.0	0.00	4/4	0.1	7.3	0.00	4/4	0.1	6.5	0.00	4/4
	0.6	0.3	60.8	0.00	4/4	0.2	48.0	0.00	4/4	0.1	51.0	0.00	4/4
	0.8	0.5	137.0	0.00	4/4	0.4	150.0	0.00	4/4	0.2	156.0	0.00	4/4
	1.0	0.7	489.8	0.00	4/4	0.7	655.8	0.00	4/4	0.2	231.8	0.00	4/4
	1.2	1.0	671.3	0.00	4/4	1.3	1121.8	0.00	4/4	0.4	565.5	0.00	4/4
50	0.2	0.2	4.8	0.00	4/4	0.1	3.0	0.00	4/4	0.1	5.0	0.00	4/4
	0.4	0.6	50.0	0.00	4/4	0.5	31.3	0.00	4/4	0.4	45.0	0.00	4/4
	0.6	1.1	224.8	0.00	4/4	1.4	197.3	0.00	4/4	0.8	301.3	0.00	4/4
	0.8	2.7	1080.8	0.00	4/4	2.8	747.8	0.00	4/4	1.2	521.0	0.00	4/4
	1.0	5.6	2233.3	0.00	4/4	5.6	1887.3	0.00	4/4	2.4	1434.5	0.00	4/4
	1.2	13.8	4757.8	0.00	4/4	14.1	4135.0	0.00	4/4	5.1	2785.3	0.00	4/4
75	0.2	0.6	8.3	0.00	4/4	0.3	3.8	0.00	4/4	0.2	9.0	0.00	4/4
	0.4	2.2	190.0	0.00	4/4	1.8	98.0	0.00	4/4	0.8	124.3	0.00	4/4
	0.6	4.9	357.0	0.00	4/4	5.8	762.3	0.00	4/4	2.3	583.3	0.00	4/4
	0.8	17.2	2544.3	0.00	4/4	23.1	4416.0	0.00	4/4	7.2	2641.3	0.00	4/4
	1.0	46.8	6122.8	0.00	4/4	64.3	12093.3	0.00	4/4	16.6	6838.0	0.00	4/4
	1.2	134.8	13725.8	0.00	4/4	184.0	26834.3	0.00	4/4	39.6	14244.8	0.00	4/4
100	0.2	2.0	26.5	0.00	4/4	2.1	71.3	0.00	4/4	1.3	116.8	0.00	4/4
	0.4	9.4	417.0	0.00	4/4	12.6	1318.3	0.00	4/4	6.8	1261.5	0.00	4/4
	0.6	46.1	4022.5	0.00	4/4	66.8	8064.3	0.00	4/4	25.7	4852.3	0.00	4/4
	0.8	300.1	22957.5	0.00	4/4	397.7	40418.0	0.00	4/4	112.3	18144.0	0.00	4/4
	1.0	739.1	53330.0	0.00	4/4	1246.0	92237.3	0.00	4/4	207.3	35710.0	0.00	4/4
	1.2	2147.0	64481.0	2.02	3/4	3092.2	140224.8	1.09	3/4	853.4	97739.0	0.00	4/4
125	0.2	3.5	34.8	0.00	4/4	4.5	56.5	0.00	4/4	2.6	169.3	0.00	4/4
	0.4	22.8	1080.8	0.00	4/4	21.3	1025.0	0.00	4/4	8.8	922.3	0.00	4/4
	0.6	228.5	16414.5	0.00	4/4	199.6	15685.5	0.00	4/4	114.9	12250.0	0.00	4/4
	0.8	1101.3	56660.3	0.01	4/4	1576.4	73335.0	0.51	3/4	608.2	55224.3	0.00	4/4
	1.0	2629.9	80940.0	1.13	2/4	3600.0	130033.5	2.46	0/4	2480.9	109520.5	0.61	3/4
150	0.2	12.1	346.5	0.00	4/4	5.9	92.5	0.00	4/4	14.6	1806.5	0.00	4/4
	0.4	114.2	7598.3	0.00	4/4	65.2	3340.3	0.00	4/4	70.9	7707.0	0.00	4/4
	0.6	1688.2	69318.3	0.41	3/4	872.9	34955.0	0.00	4/4	1089.3	45267.5	0.15	4/4
	0.8	3369.8	86787.0	1.63	1/4	3238.0	98469.0	2.10	1/4	2939.6	152951.8	0.65	3/4
200	0.2	19.0	105.8	0.00	4/4	50.9	1520.8	0.00	4/4	36.9	1890.8	0.00	4/4
	0.4	284.5	6089.8	0.01	4/4	1663.7	22516.8	0.41	3/4	2242.7	56680.8	1.10	2/4
	0.6	3138.6	46454.0	1.95	2/4	3600.0	53259.5	2.90	0/4	3600.0	83116.8	2.98	1/4
	0.8	3600.0	7867.3	10.49	0/4	3600.0	41230.5	5.98	0/4	3600.0	73966.3	5.04	0/4

Table 5.4: Computational results for Type 1, 2 and 4 instances under partial interdiction and knapsack budget constraint.

Config.		Total Interdiction				Partial Interdiction			
$\|\mathcal{V}\|$	B	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
40	2	0.3	164.9	0.00	21/21	0.0	25.4	0.00	60/60
	3	6.3	2220.7	0.00	7/7	0.1	113.3	0.00	60/60
	4					0.1	253.1	0.00	60/60
80	2	2.0	562.1	0.00	19/19	0.1	68.6	0.00	60/60
	3	15.4	5499.8	0.00	4/4	0.2	276.4	0.00	60/60
	4					0.7	940.4	0.00	60/60
120	2	3.9	1803.6	0.00	21/21	0.5	210.9	0.00	60/60
	3	92.9	20994.0	0.00	5/5	4.0	1394.2	0.00	60/60
	4					40.0	6994.1	0.00	60/60
160	2	18.4	2857.3	0.00	20/20	0.7	282.9	0.00	60/60
	3	165.6	26422.4	0.00	5/5	6.3	2430.3	0.00	60/60
	4					100.6	12509.9	0.00	60/60
200	2	183.4	5380.6	0.00	18/18	4.8	373.0	0.00	60/60
	3	490.8	58323.0	0.00	5/5	68.2	5544.7	0.01	59/60
	4					157.1	20855.2	0.06	59/60

Table 5.5: Computational results for Type 3 instances considering a cardinality budget constraint.

The results are very similar to total interdiction. For each instance type, instances of up to 100 nodes are solved for every budget with two exceptions. The computation times for instances of 100 nodes already exceed 1000 seconds and keep increasing for larger instances. However, networks of 150 and 200 nodes are still solved in reasonable times for $\eta = 0.2$ and 0.4. In this case, the BD algorithm is able to solve Type 2 instances beyond the limits of the mathematical models. These limits are instances including 50 nodes.

Type 3 instances were not mentioned in our previous discussion since they are of different size. The results for Type 3 networks are summarized in Table 5.5 and Table 5.6. Table 5.5 presents the experimentation for both total and partial interdiction with a cardinality budget constraint. The results for both types of interdiction with a knapsack constraint are reported in Table 5.6. As stated above, there are 60 different graphs for each size $|\mathcal{V}|$. Note that for total interdiction, the instances tested are less than 60. This occurs for two reasons, for several instances the budget defined either is too small to interdict any edge in the MST or is sufficiently large to disconnect the network. This comes from the fact that instances of Type 3 are very sparse Last, all Type 3 graphs have a connectivity number of 4 or less. All instances are trivial when $B = 4$ in the cardinality case or B sufficiently large in the knapsack case. No results are reported if the latter occurs.

Config.		Total Interdiction				Partial Interdiction			
$\|\mathcal{V}\|$	η	Time (s)	Nodes	Gap (%)	Opt/Tot	Time (s)	Nodes	Gap (%)	Opt/Tot
40	0.2	0.0	5.2	0.00	53/53	0.0	2.3	0.00	60/60
	0.4	0.1	70.4	0.00	42/42	0.0	14.2	0.00	60/60
	0.6	0.1	147.3	0.00	31/31	0.1	55.8	0.00	60/60
	0.8	0.3	482.1	0.00	26/26	0.1	118.7	0.00	60/60
	1.0	0.8	1275.8	0.00	14/14	0.1	215.5	0.00	60/60
	1.2	1.4	1789.6	0.00	8/8	0.2	398.0	0.00	60/60
80	0.2	0.1	48.7	0.00	49/49	0.0	13.0	0.00	60/60
	0.4	0.4	464.0	0.00	33/33	0.1	106.5	0.00	60/60
	0.6	3.1	2442.9	0.00	24/24	0.2	337.2	0.00	60/60
	0.8	14.4	7131.9	0.00	16/16	0.7	1050.4	0.00	60/60
	1.0	219.0	28947.0	0.00	8/8	1.5	1911.0	0.00	60/60
	1.2	36.7	20205.5	0.00	2/2	4.0	3776.1	0.00	60/60
120	0.2	0.3	253.3	0.00	44/44	0.1	59.6	0.00	60/60
	0.4	9.9	3007.6	0.00	30/30	0.4	408.6	0.00	60/60
	0.6	168.2	14366.8	0.00	18/18	1.4	1484.5	0.00	60/60
	0.8	850.1	65915.2	0.14	7/9	4.1	3310.2	0.00	60/60
	1.0	1077.4	83273.8	0.37	3/4	36.7	8287.1	0.00	60/60
	1.2	154.3	81319.0	0.00	1/1	105.0	16493.9	0.01	59/60
160	0.2	0.9	639.0	0.00	39/39	0.2	137.3	0.00	60/60
	0.4	79.6	13213.8	0.00	26/26	2.6	1539.8	0.00	60/60
	0.6	869.0	71853.7	0.05	13/15	39.3	8011.2	0.00	60/60
	0.8	1045.8	80504.1	0.50	5/7	159.0	22732.8	0.02	59/60
	1.0	2153.4	184462.5	1.08	1/2	364.5	33741.8	0.11	55/60
	1.2					405.8	44518.6	0.19	55/60
200	0.2	5.4	2097.7	0.00	35/35	0.4	279.5	0.00	60/60
	0.4	664.1	52399.5	0.06	23/25	17.4	4169.5	0.00	60/60
	0.6	1917.4	155321.7	0.47	10/16	172.3	19129.2	0.07	58/60
	0.8	2362.4	194469.4	1.20	3/7	382.9	37788.0	0.18	55/60
	1.0	3600.2	198632.0	2.59	0/1	504.7	40435.5	0.37	53/60
	1.2					576.7	46845.8	0.50	52/60

Table 5.6: Computational results for Type 3 instances considering a knapsack budget constraint.

Under a cardinality constraint, the instances considered are solved optimally. Times range from 0.3 in the smallest case to 490.8 seconds. For partial interdiction, only two instances with 200 nodes are not solved. However their gaps are of less than 1%. The remaining instances are solved in less than three minutes. The increased efficiency of the BD algorithm for Type 3 instances is due to the sparsity of the graph. For a $|\mathcal{V}| = 100$ nodes network, a Type 1 or Type 2 instance includes $|\mathcal{V}|(|\mathcal{V}| - 1)/2$ edges whereas a Type 3 instance has a number of edges equal to around 6% of that value.

Table 5.6 presents good results for both total and partial interdiction. Some instances with the largest sizes are not solved, such as 120, 160 and 200 nodes instances with $\eta \geq 0.8$, nevertheless their gaps are of less than 2.59%. Computa-

tional times are reasonable for networks including less than 120 nodes, climbing up to 1077.4 seconds. Resolution time and the search tree size start to increase for graphs with 160 nodes. However, our BD algorithm still solves 84 out of 89 instances of size 160 and 71 out of 84 for networks with 200 nodes in the case of total interdiction. For partial interdiction, the resolution time is at most 576.7 seconds for the largest configuration and only 34 out of 1800 instances are not solved.

A total of 5047 instances out of 6744 originally proposed were tested out. Those 1697 instances were discarded since, as previously indicated, they were trivially solved. We tested 1498 and 3549 instances assuming a cardinality and a knapsack budget constraint, respectively. The revised BD algorithm succeeded to solve 92.39% of the instances for both types of interdiction assuming a cardinality constraint. Whereas for the knapsack case, 95.9% of the instances were optimally solved. Those small percentages of unsolved problems represent the limits of the algorithm, i.e. networks of 100 nodes with $\eta \geq 5$ for the cardinality case and graphs including 125 nodes or more with $B \geq 0.8$ for the knapsack case. The revised BD algorithm is able to efficiently solve all instances defined on sparse graphs when the number of nodes is less than 200. Only a few instances remain unsolved when the graph includes 200 nodes.

5.8 Conclusions

Based on the experimentation, the proposed BD algorithm can efficiently solve instances including network of medium to relatively large sizes considering a variety of budgets. The main contribution of our algorithm is its versatility. Most of the research on the MSTI problem presented in Chapter 2 focuses on the $\bar{B}$-most vital edges which is a complicated version of the MSTI problem due to the significantly larger feasible space that a cardinality constraint defines. However, the other versions are also important, since it is not always desired to completely remove a link of the network or the cost of doing so is not necessarily equal for each link. For example, in a war situation, if the enemy wishes to block the use of a segment of road he can destroy it by bombing it with planes, tanks or some other explosive device, or simply send some units to patrol the area. Either of these options clearly does not have the same cost in terms of manpower or resources.

The algorithm proposed can handle any of the four version of the MSTI problem. In addition, it can also solve the partial interdiction version when the augmentation of an interdicted edge can be fractional, i.e. $c_e + d_e x_e$ where x_e is continuous. The latter is obtained by solving the root node of the Branch-and-Cut algorithm only.

Chapter 6

Conclusion and Recommendations

6.1 Thesis Summary

This thesis addresses the MSTI problem, providing modeling techniques and solution methods. Chapter 3 introduces four original formulations. The first one is a max − min non-linear model. Optimality constraints are developed in order to define a single-level formulation. In turn, the model is linearized using classical linearization techniques and [MSTI_{Opt}] is obtained. The second model is derived by considering the dual of the inner problem of the max − min formulation and thus obtaining a single-level linear problem. i.e. [$\text{MSTI}_{\text{Dual}}$]. A more compact dual formulation of [MSTI_{Opt}] is derived by redefining several of its constraints and by following the same dualization procedure. The resulting model is [$\text{MSTI}_{\text{SDual}}$]. The last formulation [MSTI_{Tot}] is only equivalent to the other formulations when total interdiction is considered and only under certain parameter values. Finally, two sets of valid inequalities are developed to strengthen [MSTI_{Opt}]. These four models, two versions of [MSTI_{Opt}] with the set of valid inequalities and a formulation retrieved form the literature are tested and compared. The results do not allow us to determine a unique efficient formulation. However, the best results are obtained with [MSTI_{Opt}], [$\text{MSTI}_{\text{Dual}}$] and [$\text{MSTI}_{\text{SDual}}$]. Furthermore, the computational experimentation show that all models start to struggle to solve networks including 20 or more nodes. Therefore, modeling techniques are not sufficient if instances defined on larger networks need to be solved.

Chapter 4 presents a Branch-and-Price algorithm. First, a Dantzig-Wolfe decomposition is applied on [$\text{MSTI}_{\text{SDual}}$]. The master problem is obtained by

relaxing the integrality constraint. The Restricted Master Problem is defined on the collection $\hat{\mathcal{P}}$ of interdiction plans defined by the feasible set $\mathcal{X}$. The Restricted Master Problem is solved by column generation, i.e. a column with a positive reduced cost is found by the pricing problem which is then added to $\hat{\mathcal{P}}$ and [RMP($\hat{\mathcal{P}}$)] is reoptimized. The pricing problem turns to be a knapsack problem with a modified objective function. Furthermore, the sub-problem is strengthen with a new original family of constraints. The iterative procedure of solving [RMP($\hat{\mathcal{P}}$)] and the pricing problem is repeated until no column with positive reduced cost is found. A Branch-and-Price algorithm is a Branch-and-Bound framework where at each node of the search tree the Restricted Master Problem is solved by column generation. An integer solution can be calculated for each generated column during the column generation procedure. The latter allows us not to check if integer solutions are found during the branching and only update the incumbent with the solutions provided by the columns. The computational experimentation shows that this approach is inefficient with respect to the mathematical formulations since much better computation times where obtained even for the small instances.

The revised Benders Decomposition presented in Chapter 5 is the main contribution of this thesis. The MSTI problem is decomposed in a master problem and a sub-problem. The master problem is defined over a restricted collection of trees. The objective of the master problem is to find the optimal interdiction plan that maximizes the weight of these spanning trees simultaneously. The sub-problem is parameterized with the solution of the master problem which, when solved, generates a new tree that is added to the collection of trees. The classical BD algorithm solves iteratively the master problem and sub-problem. In this thesis, we proposed a revised BD algorithm embedded in a CPLEX framework. The master problem is implemented in CPLEX, then the sub-problem is solved via control callbacks to dynamically add new trees. Moreover, the constraints of the master problem are lifted and a second family of valid inequalities is also defined to strengthen the master problem.

Different test instances were generated to test the BD algorithm. The differences between the instances are in the density or the edge weights. The results obtained with the revised BD algorithm are much better of those of Chapters 3 and 4. In the total interdiction case, instances of up to 75 nodes can be solved

fairly efficiently for both the cardinality and knapsack budget assumptions. Under the same assumptions, 100 nodes instances with a small budget are solved as well but with a significant resolution time. The BD algorithm behaves even more efficiently in the knapsack case. Instances of up to 125 nodes are solved. Instances of 150 and 200 are also solved but the efficiency start to decrease when the budget increases. Finally, a final instance set retrieved from the literature was as well efficiently solved by the algorithm proposed.

The three solution approaches presented in this manuscript, i.e. mathematical formulations, the B&P algorithm and the revised Benders Decomposition are conceived to handle the four configurations of type of interdiction and budget constraints. Whereas in the literature, the case of partial interdiction has been neglected. Therefore, the research of this thesis addresses a generalization of the MSTI problem.

6.2 Future Research

Two families of constraints are considered in our BD algorithm: the optimality constraints which are mandatory in order to have a well-defined problem and the set covering constraints which are meant to strengthen the MP. More families of constraints could be added to the MP with this same goal. The so-called Critical Edge Set constraints (Wei *et al.* (2021)) are a stronger variation of the set covering constraints. Each critical edge set constraint represents a subset of edges whose elements are part of several spanning trees. Thus, interdicting one edge in the critical edge set means the interdiction of one edge of all those spanning trees. The latter implies to have less constraints with less variables than those of the set covering constraints. The next step of our research is to include this new family of constraints in our algorithm. However, separating these inequalities from a fractional solution is $\mathcal{NP}$-hard. Wei *et al.* (2021) propose a separation algorithm from integer solutions nevertheless it remains fairly more complex than separating the set covering constraints. A first step is to incorporate this algorithm in the lazy callbacks to test its efficiency. If positive results are obtained, the second step is to design an algorithm to separate a fraction of critical edge set constraints from fractional solutions.

A direct extension of our work is to consider the MSTI problem with multiple interdiction resources. Two assumptions can be made: an edge needs different resources to be interdicted or an edge can be interdicted by one of the different resources. The modifications in our formulations and in the BD algorithm concern only the budget constraint which have to be modified to handle multiple resources.

The research of this manuscript concerns the MSTI problem where the interdictor has the capacity of interdicting any edge of the network. After interdiction, the network operator reacts optimally to this interdiction. This problem is modeled as a particular case of a bilevel problem, i.e. a $\max - \min$ problem with the same objective function. The edge reinforcement variant of the MFI and SPI problems were studied in different occasions as stated in Chapter 2. To the best of our knowledge, the MSTI problem with edge reinforcement has not been studied in the literature. This variant is normally modeled as a two-player game where the network operator plays twice. Constrained by a budget, he moves first by selecting a subset of edges that will be reinforced, i.e. edges that the interdictor cannot interdict. Then, the interdictor proceeds to interdict the network. Lastly, the network operator computes the MST after the network has been interdicted. The resulting formulation is a $\min - \max - \min$ problem with the same objective function. Our formulations can be easily adapted to this variant.

The bi-objective MSTI problem is once more a variant that, to the best of our knowledge, has not been studied in the literature. The bi-objective variant optimizes simultaneously the maximization of the weight MST and the minimization of the cost of the interdiction strategy. The optimal solution of the MSTI problem is only one of the solutions of the Pareto front of the bi-objective MSTI problem. Thus, this variant is a generalization of the MSTI problem.

Sensibility analyses of real networks is a direct application of our research. Gas pipelines, electric grids or transportation lines are examples that up to some extent can be modeled as spanning trees. They connect cities, neighborhoods or regions in order to provide services or deliver products. These networks can be subject to interdiction by malicious agents which can alter their proper functioning. A path to follow for the near future is to test out the mathematical formulations and the revised BD decomposition on these type of networks in order to locate the critical links. The latter will be used by the network operator to

prepare countermeasures in case of failures or to reinforce these links to prevent them from being tampered with.

References

Achterberg, T. (2007). *Constraint integer programming*. Ph.D. thesis, Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften. 71

Afshari Rad, M. & Kakhki, H.T. (2017). Two extended formulations for cardinality maximum flow network interdiction problem. *Networks*, **69**, 367–377. 13

Akçay, Y., Li, H. & Xu, S.H. (2007). Greedy algorithm for the general multidimensional knapsack problem. *Annals of Operations Research*, **150**, 17–29. 63

Altner, D.S., Ergun, O. & Uhan, N.A. (2010). The maximum flow network interdiction problem: Valid inequalities, integrality gaps, and approximability. *Operations Research Letters*, **38**, 33–38. 13

Baïou, M. & Barahona, F. (2008). A linear programming approach to increasing the weight of all minimum spanning trees. *Networks*, **52**, 227–234. 3, 16

Ball, M.O., Golden, B.L. & Vohra, R.V. (1989). Finding the most vital arcs in a network. *Operations Research Letters*, **8**, 73–76. 13, 20

Baycik, N.O., Sharkey, T.C. & Rainwater, C.E. (2018). Interdicting layered physical and information flow networks. *IISE Transactions*, **50**, 316–331. 19

Bayrak, H. & Bailey, M.D. (2008). Shortest path network interdiction with asymmetric information. *Networks*, **52**, 133–140. 20

Bazgan, C., Toubaline, S. & Vanderpooten, D. (2012). Efficient determination of the k most vital edges for the minimum spanning tree problem. *Computers & Operations Research*, **39**, 2888–2898. iii, 3, 16, 20, 21, 22, 23, 42, 62

Bedrosian, S. (1964). Generating formulas for the number of trees in a graph. *Journal of the Franklin Institute*, **277**, 313–326. 79

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, **4**, 238–252. 74

Bertsimas, D., Nasrabadi, E. & Orlin, J.B. (2016). On the power of randomization in network interdiction. *Operations Research Letters*, **44**, 114–120. 20

Brown, G., Carlyle, M., SalmerÃşn, J. & Wood, K. (2006). Defending critical infrastructure. *Interfaces*, **36**, 530–544. 18

Brown, G.G., Carlyle, W.M., SalmerÃşn, J. & Wood, K. (2005). *Analyzing the Vulnerability of Critical Infrastructure to Attack and Planning Defenses*, 102–123. INFORMS. 20

Burch, C., Carr, R., Krumke, S., Marathe, M., Phillips, C. & Sundberg, E. (2003). *A Decomposition-Based Pseudoapproximation Algorithm for Network Flow Inhibition*, vol. 22, 51–68. Kluwer Academic Publishers. 20

Calude, C., Marcus, S. & Tevy, I. (1979). The first example of a recursive function which is not primitive recursive. *Historia Mathematica*, **6**, 380–384. 15

Church, R.L., Scaparra, M.P. & Middleton, R.S. (2004). Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers*, **94**, 491–502. 20

Cook, W.J., Cunningham, W.H., Pulleyblank, W.R. & Schrijver, A. (1997). *Combinatorial Optimization*. John Wiley & Sons, Ltd. 24

Corley, H. & Sha, D.Y. (1982). Most vital links and nodes in weighted networks. *Operations Research Letters*, **1**, 157–160. 13, 17

Corley, H.W. & Chang, H. (1974). Finding the n most vital nodes in a flow network. *Management Science*, **21**, 362–364. 17, 20

Cormican, K.J., Morton, D.P. & Wood, R.K. (1998). Stochastic network interdiction. *Operations Research*, **46**, 184–197. 18, 20

Delgadillo, A., Arroyo, J.M. & Alguacil, N. (2010). Analysis of electric grid interdiction with line switching. *IEEE Transactions on Power Systems*, **25**, 633–641. 18

Dempe, S. (2002). *Foundations of bilevel programming*. Springer Science & Business Media. 7

Dempe, S., Kalashnikov, V., PÃŁrez-ValdÃŁs, G.A. & Kalashnykova, N. (2015). *Bilevel Programming Problems*. Energy Systems, Springer Berlin Heidelberg. 8

Dinitz, M. & Gupta, A. (2013). *Packing Interdiction and Partial Covering Problems*, vol. 7801, 157–168. Springer Berlin Heidelberg. 20

Etesami, S.R. & Başar, T. (2019). Dynamic games in cyber-physical security: An overview. *Dynamic Games and Applications*, **9**, 884–913. 19

Frederickson, G.N. & Solis-Oba, R. (1999). Increasing the weight of minimum spanning trees. *Journal of Algorithms*, **33**, 244–266. 3, 16, 20

Fulkerson, D.R. & Harding, G.C. (1977). Maximizing the minimum source-sink path subject to a budget constraint. *Mathematical Programming*, **13**, 116–118. 13, 20

Furini, F., Ljubić, I., Martin, S. & San Segundo, P. (2019). The maximum clique interdiction problem. *European Journal of Operational Research*, **277**, 112–127. 20

Golden, B. (1978). A problem in network interdiction. *Naval Research Logistics Quarterly*, **25**, 711–713. 14

Held, H., Hemmecke, R. & Woodruff, D.L. (2005a). A decomposition algorithm applied to planning the interdiction of stochastic networks. *Naval Research Logistics*, **52**, 321–328. 18

Held, H., Hemmecke, R. & Woodruff, D.L. (2005b). A decomposition algorithm applied to planning the interdiction of stochastic networks. *Naval Research Logistics*, **52**, 321–328. 20

Hemmecke, R., Schultz, R. & Woodruff, D.L. (2003). *Interdicting Stochastic Networks with Binary Interdiction Effort*, vol. 22, 69–84. Kluwer Academic Publishers. 18, 20

Hsu, L.H., Jan, R.H., Lee, Y.C. & Hung, C.N. (1991). Finding the most vital edge with respect to minimum spanning tree in weighted graphs. *Information Processing Letters*, **39**, 5. 3, 15, 20

Israeli, E. & Wood, R.K. (2002). Shortest-path network interdiction. *Networks: An International Journal*, **40**, 97–111. 2, 14, 17, 20, 77

Iwano, K. & Katoh, N. (1993). Efficient algorithms for finding the most vital edge of a minimum spanning tree. *Information Processing Letters*, **48**, 21–213. 15

Jabarzare, Z., Zolfagharinia, H. & Najafi, M. (2019). Dynamic interdiction networks with applications in illicit supply chains. *Omega*. 18

Jan, R.H., Hsu, L.H. & Lee, Y.Y. (1992). The most vital edges with respect to the number of spanning trees in two-terminal series-parallel graphs. *BIT*, **32**, 403–412. 3

Janjarassuk, U. & Linderoth, J. (2008). Reformulation and sampling to solve a stochastic network interdiction problem. *Networks*, **52**, 120–132. 18

Khachiyan, L., Gurvich, V. & Zhao, J. (2006). *Extending DijkstraâĂŹs Algorithm to Maximize the Shortest Path by Node-Wise Limited Arc Interdiction*, vol. 3967, 221–234. Springer Berlin Heidelberg, Berlin, Heidelberg. 17

Khachiyan, L., Boros, E., Borys, K., Elbassioni, K., Gurvich, V., Rudolf, G. & Zhao, J. (2008). On short paths interdiction problems: Total and node-wise limited interdiction. *Theory of Computing Systems*, **43**, 204–233. 17, 20

Kleinert, T., Labbé, M., Ljubić, I. & Schmidt, M. (2021). A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, **9**. 8

Konrad, R.A., Trapp, A.C., Palmbach, T.M. & Blom, J.S. (2017). Overcoming human trafficking via operations research and analytics: Opportunities for methods, models, and applications. *European Journal of Operational Research*, **259**, 733–745. 19

Lachhwani, K. & Dwivedi, A. (2018). Bi-level and multi-level programming problems: Taxonomy of literature review and research issues. *Archives of Computational Methods in Engineering*, **25**, 847–877. 8

Liang, W. & Shen, X. (1997). Finding the k most vital edges in the minimum spanning tree problem. *Parallel Computing*, **23**, 1889–1907. 15, 20

Lim, C. & Smith, J.C. (2007). Algorithms for discrete and continuous multicommodity flow network interdiction problems. *IIE Transactions*, **39**, 15–26. 20

Lin, K.C. & Chern, M.S. (1993). The most vital edges in the minimum spanning tree problem. *Information Processing Letters*, **45**, 25–31. 15, 20

Lubore, S.H., Ratliff, H. & Sicilia, G. (1971). Determining the most vital link in a flow network. *Naval Research Logistics Quarterly*, **18**, 497–502. 12

Magnanti, T.L. & Wolsey, L.A. (1995). Optimal trees. *Handbooks in operations research and management science*, **7**, 503–615. 22

Malaviya, A., Rainwater, C. & Sharkey, T. (2012). Multi-period network interdiction problems with applications to city-level drug enforcement. *IIE Transactions*, **44**, 368–380. 18

Malik, K., Mittal, A.K. & Gupta, S.K. (1989). The most vital k arcs in the shortest path probelm. *Operation Research Letters*, **8**, 5. 13

Martin, R.K. (1991). Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, **10**, 119–128. iii, 4, 21, 24, 25, 44, 54

McMasters, A.W. & Mustin, T.M. (1970). Optimal interdiction of a supply network. *Naval Research Logistics Quarterly*, **17**, 261–268. 12

Morton, D.P. (2010). Stochastic network interdiction. *Wiley encyclopedia of operations research and management science*. 18

Morton, D.P., Pan, F. & Saeger, K.J. (2007). Models for nuclear smuggling interdiction. *IIE Transactions*, **39**, 3–14. 2, 20

Pan, F. (2005). *Stochastic network interdiction: models and methods*. The University of Texas at Austin. 18

Pan, F., Charlton, W.S. & Morton, D.P. (2003). *A Stochastic Program for Interdicting Smuggled Nuclear Material*, vol. 22, 1–19. Kluwer Academic Publishers. 20

Pay, B.S., Merrick, J.R.W. & Song, Y. (2019). Stochastic network interdiction with incomplete preference. *Networks*, **73**, 3–22. 20

Phillips, C.A. (1993). The network inhibition problem. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, 776–785. 12

Powell, R. (2007). Defending against terrorist attacks with limited resources. *American Political Science Review*, **101**, 527–541. 19

Rahmaniani, R., Crainic, T.G., Gendreau, M. & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, **259**, 801–817. 74

Ratliff, H.D., Sicilia, G.T. & Lubore, S. (1975). Finding the n most vital links in flow networks. *Management Science*, **21**, 531–539. 12

Rocco S., C.M. & Ramirez-Marquez, J.E. (2010). A bi-objective approach for shortest-path network interdiction. *Computers & Industrial Engineering*, **59**, 232–240. 18, 20

Rocco S., C.M., Emmanuel Ramirez-Marquez, J. & Salazar A., D.E. (2010). Bi and tri-objective optimization in the deterministic network interdiction problem. *Reliability Engineering & System Safety*, **95**, 887–896. 18

Royset, J.O. & Wood, R.K. (2007). Solving the bi-objective maximum-flow network-interdiction problem. *INFORMS Journal on Computing*, **19**, 175–184. 18

Sadeghi, S., Seifi, A. & Azizi, E. (2017). Trilevel shortest path network interdiction with partial fortification. *Computers & Industrial Engineering*, **106**, 400–411. 18, 20

Salmeron, J., Wood, K. & Baldick, R. (2004). Analysis of electric grid security under terrorist threat. *IEEE Transactions on power systems*, **19**, 905–912. 2, 19

Salmeron, J., Wood, K. & Baldick, R. (2009). Worst-case interdiction analysis of large-scale electric power grids. *IEEE Transactions on Power Systems*, **24**, 96–104. 18

Scaparra, M.P. & Church, R.L. (2008). A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, **35**, 1905–1923. 20

Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons. 58

Schrijver, A. (2003). *Combinatorial optimization: polyhedra and efficiency*, vol. 24. Springer Science & Business Media. 26

Sefair, J.A. & Smith, J.C. (2016). Dynamic shortest-path interdiction. *Networks*, **68**, 315–330. 20

Shen, H. (1999). Finding the k most vital edges with respect to minimum spanning tree. *Acta Informatica*, **36**, 405–424. 16

Sullivan, K.M., Morton, D.P., Pan, F. & Cole Smith, J. (2014). Securing a border under asymmetric information. *Naval Research Logistics (NRL)*, **61**, 91–100. 18

Tanergüçlü, T., Karaşan, O.E., Akgün, İ. & Karaşan, E. (2019). Radio communications interdiction problem under deterministic and probabilistic jamming. *Computers & Operations Research*, **107**, 200–217. 19

Tsai, J., Yin, Z., Kwak, J.y., Kempe, D., Kiekintveld, C. & Tambe, M. (2010). Urban security: Game-theoretic resource allocation in networked domains. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*. 2

Vicente, L.N. & Calamai, P.H. (1994). Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, **5**, 291–306. 8

Von Stackelberg, H. (1934). Marktform und gleichgewicht; hill, r.; bazin, d.; urch, l., translators. 7

Wei, N., Walteros, J.L. & Pajouh, F.M. (2021). Integer programming formulations for minimum spanning tree interdiction. *INFORMS Journal on Computing*, **33**, 1461–1480. 3, 20, 77, 87, 99

Wollmer, R. (1964). Removing arcs from a network. *Operations Research*, **12**, 934–940. 2, 12, 20

Wollmer, R.D. (1963). *Some methods for determining the most vital link in a railway network*. Rand Corporation. 12

Wood, R.K. (1993). Deterministic network interdiction. *Mathematical and Computer Modelling*, **17**, 1–18. 2, 12, 13, 17, 18, 20

Wu, X. & Conejo, A.J. (2017). An efficient tri-level optimization model for electric grid defense planning. *IEEE Transactions on Power Systems*, **32**, 2984–2994. 18

Wu, Y., Chen, Z., Gong, H., Feng, Q., Chen, Y. & Tang, H. (2021). DefenderâĂŞattackerâĂŞoperator: Tri-level game-theoretic interdiction analysis of urban water distribution networks. *Reliability Engineering & System Safety*, **214**. 18

Yates, J. & Sanjeevi, S. (2013). A length-based, multiple-resource formulation for shortest path network interdiction problems in the transportation sector. *International Journal of Critical Infrastructure Protection*, **6**, 107–119. 20

Yuan, W., Zhao, L. & Zeng, B. (2014). Optimal power grid protection through a defender-attacker-defender model. *Reliability Engineering & System Safety*, **121**, 83–89. 18

Yuan, W., Wang, J., Qiu, F., Chen, C., Kang, C. & Zeng, B. (2016). Robust optimization-based resilient distribution network planning against natural disasters. *IEEE Transactions on Smart Grid*, **7**, 2817–2826. 2

Yuan, Y., Li, Z. & Ren, K. (2011). Modeling load redistribution attacks in power systems. *IEEE Transactions on Smart Grid*, **2**, 382–390. 2

Zhuang, J. & Bier, V.M. (2007). Balancing terrorism and natural disasters-defensive strategy with endogenous attacker effort. *Operations Research*, **55**, 976–991. 19

Résumé Etendu

La prévention est un aspect important pour tout système pour en assurer le bon fonctionnement. Les transports, les livraisons ou la production sont des processus où la prévention joue un rôle essentiel. Les catastrophes naturelles et les perturbations causées par l'homme sont des exemples de ce qui peut provoquer un dysfonctionnement de ces systèmes, ce qui, dans la plupart des cas, conduit à un fonctionnement indésirable, voire à l'absence totale de fonctionnement. Par exemple, une entreprise doit savoir comment réagir lorsque la panne d'une machine compromet sa chaîne logistique. Elle doit prévoir une absence de production, des délais, des pertes de profits, entre autres. Une analyse des opérations dans un cadre d'optimisation est indispensable.

La modélisation de ces situations dans le cadre d'un Jeu d'Interdiction (JI) est l'un des moyens les plus pratiques de garantir les meilleurs résultats. Un JI cherche à identifier une stratégie qui cause le plus de dommages aux opérations d'un processus. Il est alors possible de décider d'investir dans des mesures de sécurité supplémentaires aux points critiques ou de déployer des procédures alternatives une fois que le pire scénario que les opérations peuvent subir est connu.

Un JI implique deux entités non coopératives qui interagissent l'une avec l'autre. La première entité, appelée *l'interdicteur*, cherche à perturber les processus d'optimisation ou les paramètres de la seconde entité, appelée le *défenseur*. Dans le cadre normal d'un JI, l'interdicteur agit en premier, puis le défenseur réagit en toute connaissance des actions de l'interdicteur. L'interdicteur peut modifier les ressources du défenseur voire les détruire afin d'aggraver l'objectif du défenseur.

Les JI ont fait l'objet d'une forte attention en raison de leur grande application pratique. Les opérations militaires, la structure critique des réseaux électriques, le

maintien de l'ordre et la prévention des catastrophes naturelles sont quelques-uns des domaines d'applications des JIs.

L'un des problèmes les plus étudiés est celui du chemin le plus fiable. On modélise une situation dans laquelle un individu se déplace à travers un réseau entre une origine et un destination. Chaque lien du réseau est associé à une probabilité de le traverser sans être détecté. L'objectif de cet individu est de traverser le réseau sans être détecté pendant que l'interdicteur tente de l'intercepter. La détection se fait en interdisant des liens de telle sorte que les probabilités associées soient diminuées. Ce problème doit son nom au fait que le défenseur tente de trouver un chemin qui maximise ses chances d'arriver Ãă destination.

Un autre problème étudié est celui de l'Interdiction de Flux Maximum (IFM). Le défenseur souhaite calculer le flux maximum dans un réseau où des capacités sont imposés. L'objectif de l'interdicteur est de minimiser le flux maximum du réseau en éliminant certains de ses arcs. Cependant, l'interdicteur a des ressources limitées pour le faire. Une application directe du problème de l'IFM est apparue dans la dégradation des flux de drogue en Amérique du Sud.

Le problème d'Interdiction du Plus Court Chemin (IPCC) est un des problèmes qui a reçu le plus d'attention. Le plus court chemin dans un réseau entre les nœuds source et destination est calculé par le défenseur tandis que l'interdicteur, contraint par un budget, perturbe le réseau en modifiant le poids des arcs. Les formulations proposées peuvent gérer une interdiction partielle, c'est-à-dire une augmentation du poids des arcs ou une destruction complète de l'arc, c'est-à-dire une interdiction totale.

Le problème de l'Interdiction de l'Arbre Couvrant Minimal (IACM) est défini de façon similaire aux deux problèmes précédents. L'interdicteur cherche à allouer ses ressources dans le but de modifier la topologie du réseau dans lequel le défenseur calcule l'arbre couvrant minimal de façon à ce que le poids de ce dernier subisse le plus grand incrément. Le nombre maximum d'arêtes que l'interdicteur peut interdire est imposé par une contrainte budgétaire. La contrainte budgétaire peut être définie comme une contrainte de cardinalité, le poids d'interdiction de chaque arête étant alors égal, ou comme contrainte de sac à dos, les poids d'interdiction étant différents. Le premier travail sur le problème IACM est un cas particulier où l'interdicteur élimine une seule arête du réseau. Les travaux ont été ensuite étendus au cas où l'interdicteur élimine un nombre arbitraire d'arêtes.

Différentes formulations mathématiques ont été présentées. Ces modèles tirent avantage de la structure du problème IACM. Les articles concernant le problème IACM fournissent une diversité de modèles et de méthodes de résolution, mais ils considèrent tous certains cas particuliers. Par exemple, il existe un algorithme efficace qui résout le problème IACM lorsque l'interdicteur élimine un nombre fixe d'arêtes. La variante d'interdiction partielle, c'est-à-dire lorsque l'interdicteur augmente seulement le poids des arêtes, a reçu peu d'attention. Des algorithmes qui résolvent cette variante existent mais une étude sur les formulations mathématiques est manquante.

A notre connaissance, aucun algorithme n'a été explicitement conçu pour résoudre le problème IACM général. Cette thèse propose des modèles et des algorithmes pour une généralisation du problème IACM. Cette dernière est obtenue en considérant les deux types d'interdiction, l'interdiction totale et l'interdiction partielle. De plus, l'interdicteur peut être limité par une contrainte de budget, modélisée comme une contrainte de sac à dos. La contrainte de cardinalité, qui est un cas particulier, est également considérée. Le travail présenté dans cette thèse est structuré comme suit.

Une revue de la littérature sur les problèmes IFM, IPCC et IACM est présentée au chapitre 2. Nous commençons par introduire le concept de jeux de Stackelberg et d'optimisation à deux niveaux. Ensuite, leur relation avec le JI est expliquée. Une formulation générale des problèmes d'interdiction de réseau déterministe est proposée. La littérature relative aux trois problèmes de réseau considérés est discutée. La discussion se concentre principalement sur les cas particuliers qui ont été traités et les différents algorithmes qui ont été conçus. Les variantes et les applications de chacun des problèmes de réseau sont également examinées.

Le chapitre 3 présente sept modèles mathématiques pour le problème IACM. Une première formulation venant de la littérature est rappelée. Un deuxième modèle original basé sur une formulation linéaire du problème ACM est présenté. Cette dernière formulation est un problème non linéaire de $\max - \min$. Elle est donc reformulée et linéarisée pour obtenir un problème linéaire à un seul niveau. De plus, des inégalités valides sont développées pour renforcer le modèle. Deux formulations supplémentaires sont proposées. Elles sont basées sur le problème dual du problème non linéaire $\max - \min$. Toutes les formulations traitent les deux types d'interdiction. Un dernier modèle qui ne considère que le

cas d'interdiction totale est présenté. Des instances sont générées comme celles de la littérature et utilisées pour comparer les efficacités de nos formulations.

Le problème IACM est ensuite résolu par un algorithme de Branch-and-Price dans le chapitre 4. La formulation la plus compacte définie au chapitre 3 est utilisée pour définir le problème maître restreint et le problème de pricing. Ce dernier est obtenu en appliquant une décomposition de Dantzig-Wolfe. Une famille de contraintes qui excluent les stratégies d'interdiction non pertinentes est développée et ajoutée au problème de pricing. Ensuite, un algorithme de génération de colonnes intégré dans un schéma Branch-and-Bound est présenté. Les expériences numériques sont réalisées sur l'ensemble d'instances générées au chapitre 3.

La principale contribution de cette thèse est présentée au chapitre 5. Un algorithme de décomposition de Benders est développé pour traiter le problème IACM. Le problème IACM est décomposé en un Problème Maître (PM) et un sous-problème. Une famille de contraintes pour le PM, appelée contraintes d'optimalité, est introduite et liftée. Une famille d'inégalités valides est également proposée pour renforcer la relaxation linéaire du PM. Le sous-problème, paramétré par la solution du PM, consiste uniquement en la détermination d'un ACM. L'algorithme de décomposition de Benders est implémenté à l'aide d'un solveur commercial. Le PM est résolu à l'optimalité une seule fois et le sous-problème est résolu dans des méthodes définies par l'utilisateur qui sont données au solveur. Quatre types d'instances sont générés. Les instances se différencient les unes des autres par le poids des arêtes et la densité du réseau. L'algorithme est évalué pour chacune des quatre configurations possibles en fonction du type d'interdiction et des contraintes budgétaires.

Le manuscrit se termine au chapitre 6 par les conclusions générales de la thèse. Les limites et les travaux futurs sont également discutés.

Models and Algorithms for the Minimum Spanning Tree Interdiction Problem

ABSTRACT

In this thesis, we study the Minimum Spanning Tree Interdiction (MSTI) problem. This problem is a two-player game between a network operator and an interdictor. The former aims to determine a Minimum Spanning Tree (MST) in a network. Constrained by a budget, the latter seeks to change the network topology to increase the weight of a MST. Two types of interdiction are considered: total and partial interdiction. A total interdicted edge is considered absent while the weight of a partial interdicted edge is augmented by a predefined amount. The interdictor's budget is modeled as a cardinality, each edge has the same interdiction weight, or knapsack constraint, the interdiction weights might be different. Seven mathematical formulations for the MSTI problem are devised. They proved to be efficient on small and medium-size graphs. A Branch-and-Price algorithm and a Benders Decomposition algorithm are designed for larger graphs. In addition, valid inequalities are proposed to strengthen the models and improve the efficiency of the proposed methods. Instances including up to 200 nodes and 19900 edges are solved to optimality.

Keywords: minimum spanning tree, network interdiction problems, interdiction problems, Benders decomposition, branch-and-price, integer programming, operations research, graph theory, algorithms.

Modèles et Algorithmes pour le Problème d'Interdiction de l'Arbre Couvrant de Poids Minimal

RÉSUMÉ

Dans cette thèse, nous étudions le problème de l'Interdiction de l'Arbre Couvrant Minimal (IACM). Ce problème est un jeu à deux joueurs entre un opérateur de réseau et un interdicteur. Le premier détermine un Arbre Couvrant Minimal (ACM) dans un réseau. Limité par un budget, le second cherche à changer la topologie du réseau pour augmenter le poids de l'ACM. Deux types d'interdiction sont considérés : l'interdiction totale et l'interdiction partielle. Une arête totalement interdite est considérée comme absente tandis que le poids d'une arête partiellement interdite est augmentée d'une quantité predéfinie. Le budget de l'interdicteur est modélisé par une contrainte de cardinalité, chaque arête a le même poids d'interdiction, ou une contrainte de sac à dos, les poids d'interdiction peuvent être différents. Sept formulations mathématiques du problème IACM ont été élaborées. Elles se sont avérées efficaces pour des graphes de petite et moyenne taille. Un algorithme de type "Branch-and-Price" et un algorithme basé sur la décomposition de Benders ont été conçus pour les graphes de plus grande taille. En outre, des inégalites valides sont proposées pour renforcer les modèles et améliorer les performances des algorithmes. Des instances de 200 sommets et 19900 arêtes ont ainsi pu être résolues optimalement.

Mots-clés: arbre couvrant de poids minimal, problèmes d'interdiction, problèmes d'interdiction de réseaux, decomposition de Benders, "branch-and-price", programmation en nombres entiers, recherche operationelle, théorie des graphes, algorithmes.