Tri-Level Mixed-Integer Linear Programming Problems: Solution Approaches and Applications

TRI-LEVEL MIXED-INTEGER LINEAR PROGRAMMING PROBLEMS: SOLUTION APPROACHES AND APPLICATIONS

BY

RAMY ABDALLAH, B.Sc., M.Sc.

A THESIS
SUBMITTED TO DEGROOTE SCHOOL OF BUSINESS
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2022) McMaster University
(DeGroote School of Business) Hamilton, Ontario, Canada

TITLE:	Tri-Level Mixed-Integer Linear Programming Problems: Solution Approaches and Applications
AUTHOR:	Ramy Abdallah B.Sc. (Electrical Engineering), M.Sc. (Physics and Engineering Mathematics) Alexandria University, Egypt
SUPERVISOR:	Dr. Elkafi Hassini and Dr. Wael El-Dakhakhni
NUMBER OF PAGES:	xx, 294

To my parents, all that I am or hope to be, I owe to you

Lay Abstract

Multi-level programming has been suggested as a suitable methodology for modelling the interaction between the different levels of decisions in organizations that follow a hierarchical structure. It has been used in practical contexts such as in determining pricing strategies, and energy management. Due to the rise of decentralized decision-making and the need for efficient algorithms, the overarching motivation of this thesis is to develop algorithms suitable for solving multi-level programming problems. We develop solution strategies for solving different classes of multi-level problems including novel heuristics, general-purpose solvers, and exact algorithms. We address different classes of multi-level problems and apply the proposed solution approach(es) on relevant applications. Although the developed techniques are inspired by specific practical applications, they can be applied in many other domains. We show numerically that our proposed solution approaches lead to better solution quality and are computationally more efficient. Furthermore, our proposals form a cornerstone for interesting theoretical and algorithmic developments in this area of research.

Abstract

Multi-level programming has been suggested as a suitable methodology for modelling the interactions between the different levels of decisions in organizations that follow a hierarchical structure. It has been applied in different fields such as revenue and energy management. Due to the rise of decentralized decision-making and the need for efficient algorithms, the overarching motivation of this thesis is to develop algorithms suitable for solving multi-level programming problems and testing them on practical applications. First, we conduct a bibliometric analysis of the literature to categorize the major topics of study and solution methodologies. In addition, we identify research gaps and future research directions. Second, we direct our focus to developing efficient algorithms for solving specific classes of linear tri-level programs. In Chapter 3, we propose three heuristic-based approaches; each heuristic type offers a trade-off between solution quality and computational time. To illustrate our solution approaches, we present an application for defending critical infrastructure to improve its resilience against intentional attacks. In Chapter 4 we study bi-level mixed-integer linear problems and a general class of tri-level programs by proposing a general-purpose algorithm capable of handling mixed-integer variables in both levels of a bi-level linear program and solving a general class of tri-level mixed-integer programs with a convex optimization problem being at the most lower-level. In chapter 5, we examine a three-level non-cooperative game with perfect information that can have a min-max-min or a max-min-max structure. We propose a heuristically-enhanced exact algorithm. We demonstrate the proposed algorithm on two applications: defending critical infrastructure and the capacitated lot-sizing problem with the capability of interdiction and fortification.

Acknowledgements

First and foremost, I am indebted to my supervisor, Professor Elkafi Hassini, for his trust, support, thoughtful suggestions, fruitful discussions, and his willingness to share his time, wisdom, and expertise despite all other academic and professional commitments. I am grateful for all that I learned from him. He continues to set an exemplary role model of a scholar.

To my co-supervisor, Dr. Wael El-Dakhakhni, thanks for your guidance and support during those productive years. I would like to express my sincere gratitude and appreciation to my supervisory committee members, Professor Mahmut Parlar, Professor Yasser Haddara, and my external examiner Professor Joe Naoum-Sawaya for their enthusiastic engagement in the discussions and insightful comments that made this work better than it otherwise would have been. My appreciation goes to the professors from whom I have learned a lot throughout the graduate courses required for the completion of my doctoral degree. I am grateful for the financial support from the Ontario Trillium Doctoral scholarship, Dr. Elkafi Hassini, and Dr. Wael El-Dakhakhni.

I cannot begin to express my thanks to my parents, and brother for supporting my

education for as long as I can remember. Without their love and guidance I would not have been where I am today. I will forever remember the countless hours they have spent listening to me on the phone, while being thousands of miles away, to ease and relieve my stress. I am extremely grateful to my partner in life, Yosra, for her constant love and support during the past two years of this program. It was most comforting and encouraging to be able to count on her presence and patience during all the challenges of this program. My sincere appreciation further goes to my brother- and parents-in-law whose support during this program was invaluable and most uplifting.

Last but not least, I would like to take this opportunity to thank all my friends and colleagues, past and present, and members of the McMaster community.

Abbreviations

$\mathcal{NP}$	non-deterministic polynomial time
BLPs	Bi-level programmes
TLPs	Tri-level programmes
RTLP	Reduced tri-level programme
ORMS	Operations research & management sciences
BMILP	Bi-level mixed-integer linear problem
B&B	branch-and-bound
Karush-Kuhn-Tucker	KKT
B&C	branch-and-cut
BILPs	Bi-level integer linear problems
ISI	International scientific indexing
WoS	Web of Science
SCP	Single country publications
MCP	Multiple country publications

LF	Leader-Follower relation
SL	Secondary-Leadership relation
PF	Primary-Followership relation
SF	Secondary-Followership relation
DOI	Digital object identifier
MLP	Multi-level programming
BOT	Build-operate-transfer
MBLP	Mixed-binary linear programmme
AD	Attacker-defender
DAD	Defender-attacker-defender
DAO	Defender-attacker-operator
MEA	Modified enumeration algorithm
C&C	Column-and-Cnstraint
HPP	High point problem
LLP	Lower-level problem
CSV	Comma-separated values
IG	Interdiction Game
FG	Fortification Game
OAR	Optimal attack-recourse
C&CG	Column-and-constraint generation
CLSIPF	Capacitated lot-sizing interdiction problem with fortification

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The increase in the digital transformation of organizational processes has facilitated more decentralized decision making. Distributed asymmetric coordination mechanisms is suggested as an approach to model the nonlinear and intractable nature of decisions in such environments (Lu et al. 2012). Mathematical programming has played an important role in devising coordination solutions since the early works on decomposition (Dantzig & Wolfe 1960, Benders 1962). In distributed optimization (Yang et al. 2019), agents cooperate to minimize a global function, which is a sum of local objective functions. Each agent performs local computation based on the information received, so that the optimization problem can be solved in a distributed manner, where all agents have the same control/power over the decision. This thesis discusses a different concept, which is decentralized decision-making where there is a hierarchy of power between decision entities and uses multi-level programming to model the decentralized decision-making process. Each decision entity has a place in the hierarchical structure and control over a set of decision variables that are not shared with other entities.

In order to model the interactions between different levels of decisions, multi-level programming has been suggested as a suitable methodology for capturing those interactions. Multi-level programming is concerned with nested optimization problem where the decision variables are controlled at different levels (usually by different decision makers) and they can impact all constraints and objectives functions at all levels (Avraamidou & Pistikopoulos 2022). Despite their complexity due to the inherent hierarchical structure of decentralized decision making processes, multi-level mathematical programs have been applied to model a wide range of decision problems in different fields, such as forestry (Parkatti et al. 2019), transportation and road planning (Gu et al. 2019), disaster management (Irohara et al. 2013), generation and transmission expansion planning (Hong et al. 2017), supply chain and waste management (Fathollahi-Fard et al. 2018), as well as defence, security and reliability assessment (Mahmoodjanloo et al. (2016), Lin & Bie (2018)). As we will show in Chapter 2, the majority of the literature in the area of multi-level programming is focused largely on bi-level programming, with some studies on tri-level programmes (TLPs) and a few on generalized multi-level programmes.

In this thesis, our goal is to push the research frontier in the area of multi-level programming from an algorithmic and applications perspectives. To do so, we target the tri-level programming area and at the same time make use and contribute to the area of bi-level programming. Moreover, we offer a unifying perspective on the research developments in tri-level programming.

The thesis is written in the form of a “sandwich” thesis. Thus, each chapter is written as a standing-alone research article, and as such each chapter has its own literature review section. Hence, it is expected that some chapters may share references. In particular, Chapter 2 and Chapter 4 share a detailed background on bi-level programs. It is used in Chapter 2 to pave the discussion towards tri-level programs, while in Chapter 4, this background is necessary for understanding the previously proposed general-purpose solvers and how it differs from our proposed implementation. Additionally, Chapter 2 and Chapter 3 share a discussion on resolving multiple optimal solutions and selection approaches in tri-level programs. The discussion in Chapter 2 is more generalized as it offers an overall view of the possible resolution strategies, while in Chapter 3, the discussion is directed toward the special class of tri-level programs under study. Lastly, each of the modelling chapters shares a common application, which is defending critical infrastructure, and as such the notations and definitions are included accordingly for the sake of the readers’ convenience.

Furthermore, all chapters fall under the umbrella of tri-level programs. In particular, in Chapter 2 we provide a bibliometric analysis of the literature on tri-level programs and introduces definitions and taxonomy of the tri-level linear program structure, while Chapter 3 proposes three heuristic-based solution algorithms generalized to work on a class of tri-level programs. In Chapter 4, we develop a general-purpose branch-and-bound solver suitable for solving bi-level mixed-integer linear programs and a more generalized class of tri-level problems than that of Chapter 3. Furthermore, in Chapter 5, we propose an exact decomposition-based approach that relies on penalty terms to tackle a special class of tri-level problems known as fortification games. Finally, in Chapter 6 we summarize our findings from this dissertation and provide directions for future research endeavours. In the next section, we will outline the organizational structure of this thesis and elaborate on the contributions of each chapter.

1.1 Thesis Organization

Chapter 2

The aim of Chapter 2 is to provide a summary for essential bi-level programming knowledge that is a prerequisite for conducting research in the area of tri-level programming. Additionally, the motivation behind Chapter 2 is to clarify common misconceptions by introducing definitions and classifying highly cited and co-cited research works pertaining to tri-level programs. Ultimately, the aim is to offer a synthesis of the literature and uncover the conceptual structure needed to further push the frontiers of multi-level programming in terms of theory, solution methods and applications. We achieve this goal by conducting a bibliometric analysis of the literature on multi-level programs. Furthermore, a meta-analysis is done using the R bibliometrix package (Aria & Cuccurullo 2017) to extract insights from the reviewed publications and their citations. Additionally, we attempt to clear some common misconceptions, provide some unifying definitions, and a taxonomy of TLPs.

Chapter 3

Multi-level programming, even for the simplest case of bi-levels, are strongly $\mathcal{NP}$-hard (Jeroslow 1985). Given the increasing number of practical applications that would benefit from modelling the decentralized decision-making process, there is a need for developing efficient solution algorithms in this area, and in particular for TLPs (Scaparra & Church 2008). To address this computational challenge, we develop three different heuristic-based approaches for solving a specific class of TLPs, in which the leader has direct control over some of the follower's decisions, with a common objective function shared at all levels. Each solution approach offers a trade-off between solution quality and computational time. To illustrate our solution approaches, we present an application for defending critical electric grid infrastructure to improve its resilience against intentional attacks. We also propose a modified implementation of a widely-adopted enumeration algorithm in this area, with a warm-starting solution technique that significantly enhanced the computational performance of the enumeration algorithm. We test our algorithms on three electrical transmission networks that vary in size and present the results of our numerical computations as well as some insights.

Chapter 4

Despite the need for general-purpose solvers for multi-level programs, there have not been enough efforts dedicated to the algorithm development of such solvers (Fischetti et al. 2017). This is mainly due to the challenging nature of multi-level programs that are proven to be $\mathcal{NP}$-hard even in their most simplest case of continuous bi-level linear programs. Recently, researchers have started to gain interest in developing bi-level general-purpose solvers to address the diverse applications that require flexible solvers that can be accustomed to customers' needs. This research work proposes a general-purpose algorithm capable of handling mixed-integer variables in both levels of a bi-level linear program. Moreover, it also solves a general class of tri-level mixed-integer programs with a convex optimization problem being at the most lower-level. In this Chapter, we generalize the class of tri-level programs that we discussed in Chapter 3. In particular, we allow the presence of mixed-integer variables in the first- and second- levels, in addition to having different objective functions across all levels. The class of tri-level programs handled by our branch-and-bound algorithm has been motivated by defending critical infrastructure applications (Brown et al. 2006), which has been found to the most impactful in tri-level programming by our literature review in Chapter 2. Protecting critical infrastructure ((Arroyo & Galiana 2005), (Arroyo 2010), (Akbari-Jafarabadi et al. 2017), Alvarez (2004)) has been extensively studied in the literature. In particular, we direct our focus to defending electrical power grids due to the interdependence of all other critical infrastructures on the reliable operation of the electrical transmission networks. The motivation for using a branch-and-bound approach for defending electrical power grids is three-folds:

- Finding alternative solutions that would enhance the set of options available for the leader (i.e., first decision-maker) which would protect the transmission network against worst-case scenarios in case of operational hidden constraints, that might impede the original fortification plan.

- Most recent research work ((Yuan et al. 2014),(Davarikia & Barati 2018), (Davarikia et al. 2020)) done on protecting critical infrastructure, especially on electrical transmission networks, used the column and constraint (**C&C**) generation algorithm or a variation of the Bender's algorithm (Wu & Conejo 2017) with no guarantee of reaching optimal solutions; these algorithms are known to require fine tuning of some parameters (e.g., gap between lower-bound and upper-bound, and penalty values) to converge; this tuning might differ from an electrical network to another. Tackling the problem of protecting electrical transmission networks with a reliable general-purpose branch-and-bound (**B&B**) algorithm would provide a benchmarking tool that can determine all optimal fortification strategies.

- The B&B algorithm would pave the way for efficient and exact solutions methods for protecting electrical transmission networks by offering insights on nodes' characteristics that contain optimal solutions.

Furthermore, in addition to the aforementioned contributions from the application perspective, we have the following contributions in the area of algorithmic developments for tri-level programming:

- We present a branch-and-bound algorithm with a new branching rule which can be used as a general-purpose bi-level mixed-integer linear program solver.
- We test our algorithm on a test-bed of randomly generated instances that have been previously presented in the literature (Xu & Wang 2014) for validation. We report on computational efficiency in addition to the numbers and types of relaxation problems solved to reach the optimal solution(s).

- We test our algorithm on a specific class of tri-level problems which can be reduced to a mixed-integer bi-level program; where we focus on the application of defending electrical transmission networks.
- Furthermore, in order to enrich the test bed of bi-level mixed-integer linear problems, we provide a Matlab live editor that converts any electrical transmission network to a bi-level mixed-integer program instance, in the context of enhancing the resilience of the electrical network under consideration.

It is important to note that, although, protecting critical infrastructure, and in particular electrical power grids is a special class of tri-level programs, the branch-and-bound algorithm can handle more general versions of tri-level programs. Furthermore, we use our Matlab-based tool to generate instances for two electrical transmission networks and report on their numerical results.

Chapter 5

This chapter examines a special class of tri-level optimization problems, which are also known as Stackelberg sequential games. In general terms, the three-level non-cooperative game, with perfect information, can either have min-max-min or max-min-max structure, where each level represents a player sharing a set of items with the next player, and optimizing a common objective function in opposite direction. These problems are notoriously difficult to optimize, because of the inherent tri-level structure which is crucial for modelling the players' interactions. The three-stage problem structure cannot be evaded, if the most lower-level problem is $\mathcal{NP}$-hard.

Nevertheless, even for the simplest case where the lower-level problem is a convex problem, and the tri-level problem can be reduced to a bi-level structure using KKT conditions or duality theory, the mathematical program is known to be strongly $\mathcal{NP}$-hard (Bard 1991). It should be noted that the solution approaches developed in Chapter 3 and Chapter 4 can only handle tri-level programs with convex lower-level problems. In this chapter, we propose a heuristically-enhanced exact algorithm for solving the aforementioned class of tri-level problems, where the most lower-level problem can be $\mathcal{NP}$-hard. The main idea of the algorithm relies on forming a single-level equivalent of the tri-level problem, where the feasible region is constructed in an iterative manner. Moreover, we rely on heuristics gained from the structural domain-knowledge of the application to enhance the formation of the feasible region. This idea can be implemented on various applications, and we demonstrate the effectiveness of our proposed solution on two applications. In particular, the first one is, the widely studied, application of defending critical infrastructure to improve its resilience against intentional attacks. In this context, we use a defender-attacker-operator model and apply it to electrical transmission networks, where the most lower-level is a convex optimization problem. The second application is the capacitated lot-sizing problem with the capability of interdiction and fortification; this modified version of the ubiquitous lot-sizing problem is characterized by having its most lower-level a mixed-binary program rendering the overall tri-level problem inherently very difficult to solve. We test our solution approaches on three electrical networks that varies in size, and randomly generated instances of lot-sizing problems. Furthermore, we present the results of our numerical computations as well as some insights.

Chapter 6

This Chapter includes our concluding remarks. We also propose research directions to expand on the research questions presented in this thesis.

1.2 Description of Thesis Contributions to Publications

This thesis has been written in the form of a “sandwich thesis.” Chapter 3 of this thesis has already appeared in *European Journal of Operational Research*, while Chapter 2, 4, and 5 are currently being revised by co-authors, and will be submitted afterwards.

1.3 Author’s Statement of Contribution

I am the author of this thesis and the first author of all works under revision/review, submitted or accepted for publication that are included in this thesis.

Bibliography

Akbari-Jafarabadi, M., Tavakkoli-Moghaddam, R., Mahmoodjanloo, M. & Rahimi, Y. (2017), ‘A tri-level r-interdiction median model for a facility location problem under imminent attack', *Computers & Industrial Engineering* **114**, 151–165.

Alvarez, R. E. (2004), Interdicting electrical power grids, PhD thesis, Monterey, California. Naval Postgraduate School.

Aria, M. & Cuccurullo, C. (2017), 'bibliometrix: An r-tool for comprehensive science mapping analysis', *Journal of informetrics* **11**(4), 959–975.

Arroyo, J. M. (2010), 'Bilevel programming applied to power system vulnerability analysis under multiple contingencies', *IET generation, transmission & distribution* **4**(2), 178–190.

Arroyo, J. M. & Galiana, F. D. (2005), 'On the solution of the bilevel programming formulation of the terrorist threat problem', *IEEE transactions on Power Systems* **20**(2), 789–797.

Avraamidou, S. & Pistikopoulos, E. (2022), *Multi-level Mixed-Integer Optimization: Parametric Programming Approach*, Walter de Gruyter GmbH & Co KG.

Bard, J. F. (1991), 'Some properties of the bilevel programming problem', *Journal of optimization theory and applications* **68**(2), 371–378.

Benders, J. (1962), 'Partitioning procedures for solving mixed-variables programming problems '', *Numerische mathematik* **4**(1), 238–252.

Brown, G., Carlyle, M., Salmerón, J. & Wood, K. (2006), 'Defending critical infrastructure', *Interfaces* **36**(6), 530–544.

Dantzig, G. B. & Wolfe, P. (1960), ‘Decomposition principle for linear programs’, *Operations research* **8**(1), 101–111.

Davarikia, H. & Barati, M. (2018), ‘A tri-level programming model for attack-resilient control of power grids’, *Journal of Modern Power Systems and Clean Energy* **6**(5), 918–929.

Davarikia, H., Barati, M., Al-Assad, M. & Chan, Y. (2020), ‘A novel approach in strategic planning of power networks against physical attacks’, *Electric Power Systems Research* **180**, 106140.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. & Mirjalili, S. (2018), ‘Hybrid optimizers to solve a tri-level programming model for a tire closed-loop supply chain network design problem’, *Applied Soft Computing* **70**, 701–722.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2017), ‘A new general-purpose algorithm for mixed-integer bilevel linear programs’, *Operations Research* **65**(6), 1615–1637.

Gu, Y., Cai, X., Han, D. & Wang, D. Z. (2019), ‘A tri-level optimization model for a private road competition problem with traffic equilibrium constraints’, *European Journal of Operational Research* **273**(1), 190–197.

Hong, S., Cheng, H. & Zeng, P. (2017), ‘Nk constrained composite generation and transmission expansion planning with interval load’, *IEEE Access* **5**, 2779–2789.

Irohara, T., Kuo, Y.-H. & Leung, J. M. (2013), From preparedness to recovery: a tri-level programming model for disaster relief planning, *in* 'International Conference on Computational Logistics', Springer, pp. 213–228.

Jeroslow, R. G. (1985), 'The polynomial hierarchy and a simple model for competitive analysis', *Mathematical programming* **32**(2), 146–164.

Lin, Y. & Bie, Z. (2018), 'Tri-level optimal hardening plan for a resilient distribution system considering reconfiguration and dg islanding', *Applied Energy* **210**, 1266–1279.

Lu, S. Y., Lau, H. Y. & Yiu, C. K. (2012), 'A hybrid solution to collaborative decision-making in a decentralized supply-chain', *Journal of Engineering and Technology Management* **29**(1), 95–111.

Mahmoodjanloo, M., Parvasi, S. P. & Ramezanian, R. (2016), 'A tri-level covering fortification model for facility protection against disturbance in r-interdiction median problem', *Computers & Industrial Engineering* **102**, 219–232.

Parkatti, V.-P., Assmuth, A., Rämö, J. & Tahvonen, O. (2019), 'Economics of boreal conifer species in continuous cover and rotation forestry', *Forest Policy and Economics* **100**, 55–67.

Scaparra, M. P. & Church, R. L. (2008), 'A bilevel mixed-integer program for critical infrastructure protection planning', *Computers & Operations Research* **35**(6), 1905–1923.

Wu, X. & Conejo, A. J. (2017), ‘An efficient tri-level optimization model for electric grid defense planning’, *IEEE Transactions on Power Systems* **32**(4), 2984–2994.

Xu, P. & Wang, L. (2014), ‘An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions’, *Computers & operations research* **41**, 309–318.

Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., Hong, Y., Wang, H., Lin, Z. & Johansson, K. H. (2019), ‘A survey of distributed optimization’, *Annual Reviews in Control* **47**, 278–305.

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

Chapter 2

Tri-Level Programming Problems: Taxonomy and Bibliometric Analysis

Abstract

There have been several reviews on bi-level programming problems, and their relevant applications and solution algorithms; however, there have not been considerable effort on reviewing and synthesizing the tri-level programming literature. This review distinguishes itself by focusing on those specific multi-level programs research problems that have appeared in the Web of Science. First, we start by discussing our research methodology and proceed with a bibliometric analysis of research articles to gain a birds-eye view of the available literature. From which we identify, and extract the most influential research work. We then proceed with a systematic review, and breakdown of the articles. Moreover, we introduce some definitions that will help in categorizing tri-level programs. Furthermore, we discuss the taxonomy of papers, solution approaches and applications related to tri-level Programs. In the end, we provide some recommendations for future research directions.

2.1 Introduction

Industrial multicriteria decisions and business management problems require compromising objectives of different interacting hierarchical decision entities. Multi-level decision-making techniques, inspired from the Stackelberg game theory (von Stackelberg 2011), have been established to tackle those compromises arising among decentralized decision-making entities through multi-level mathematical programming. The basic concept of multi-level programming is that, the first-level decision-maker (i.e., leader) sets her/his objective(s) and/or decision(s), or strategy set, and then each subordinate-level (i.e., follower(s)) chooses the decisions that best serve their interests. Those decisions are submitted and modified by the first-level decision-maker in consideration of the overall benefit of the hierarchical organization. It is important to note that multi-level mathematical programming operates under the assumption of perfect information from a game-theoretic framework; that is, each player when making any decision, is perfectly informed of all the events that have previously occurred, including the initial state of the game (e.g., the starting hands in a card game) (Mycielski 1992). It is also important to note the difference between perfect information and complete information; as the latter implies knowledge of each decision entity's utility functions, pay-offs and strategies, but players may not see all of the moves made by other players. Hence, a game with perfect information may or may not have complete information. Finally, we note that bi-level programming offers a more general modelling framework than the classic Stackelberg game, in that in the latter co-operation is prohibited.

In general, multi-level mathematical programs consider a class of optimization problems characterized by constraints, which contain optimization problems. The first appearance dates back to 1973, in a paper co-authored by Bracken & McGill (1973) discussing properties of functions that lead to convex programming problems. Since the 1980s, a vast amount of research has been devoted to address optimality conditions (Bard 1984, 1991, Dempe et al. 2006) and solution algorithms for solving linear (Hansen et al. 1992, Wen & Hsu 1992), non-linear (Edmunds & Bard 1991, Wang et al. 2005), and discrete problems (Bard & Moore 1990, Vicente et al. 1996) of a special case of multi-level programs consisting of two levels namely: leader and follower, which is also referred to as bi-level programming (Dempe 2002). Bi-level programmes (**BLPs**) are considered to be the simplest case of multi-level problems in general; nevertheless, even in their simplest forms, where decision variables are continuous, objective functions and constraints are linear, have been proved to be $\mathcal{NP}$-hard by Jeroslow (1985), Ben-Ayed & Blair (1990), and Bard (1991). Different solution algorithms have been developed to address bi-level programs such as descent algorithms, extreme points algorithms, complementary pivot algorithms, penalty algorithms, bundle algorithms, trust region, and smoothing methods (Dempe (2002), Bard (2013)).

Applications have been a crucial factor for the development of bi-level programming, as decentralized decision-making is becoming more ubiquitous. For instance, some intriguing applications involve the determination of optimal tariffs/prices for road tolls (Yin 2000), electricity prices (Carrión et al. 2009), optimal penalization of deviations for transported gas amounts (Berry et al. 1999), and cross-dock truck scheduling (Konur & Golias 2013).

Tri-level programmes (**TLPs**) inherit all the properties of BLPs, but also add to the hierarchical structure one upper-level along with its associated set of decision variables, constraints and objective function. In most studies on tri-level programming, it is implicitly assumed that an optimal solution of lower-levels' objectives for each decision made at the upper-levels is unique (Sarhadi et al. (2017), Yao et al. (2007), Alguacil et al. (2014), Schweitzer & Medal (2019)). This is generally not always true, as any non-strictly convex (concave) minimization (maximization) might have multiple optimal solutions. In a tri-level problem, the selection of alternative optima (i.e., degenerate solutions) at a particular level yields the same results for that level. However, each of the alternatives has a different impact on the overall problem Florensa et al. (2017). That is why it is important to determine the selection criteria for the upper-level decision-maker(s) among the different solution alternatives in the lower-levels, otherwise the tri-level model would be ill-posed. The aim of this study is to clarify similar misconceptions by introducing definitions and classifying highly cited and co-cited research work pertaining to TLPs. Our report can help researchers in understanding the conceptual structure needed to further push the frontiers of multi-level programming in terms of theories, solution methods and applications. To so we start by identifying articles that have been influential in shaping the literature on tri-level programs. The main contributions of this research can be summarized as follows:

- We provide a bibliometric analysis for multi-level programs with a focus on TLPs, by searching for possible keywords to pull relevant literature from the Web of Science core collection.
- A meta-analysis is done using the R bibliometrix package (Aria & Cuccurullo 2017) to extract useful knowledge from the data, and depict it through intuitive visualizations.
- We direct our attention to Operations Research & Management Science (**ORMS**) area for a systematic review.
- In an effort to clear some common misconceptions, and disseminate the literature, we provide some definitions to structure the taxonomy of TLPs.
- We provide a list of influential and necessary core knowledge as well as directions for future research.

2.2 Background

The BLP problem is a special case of multi-level programs, when there are only two decision-makers. A generic BLP can be formulated as

$$
\begin{aligned}
\text{`}\max_{\boldsymbol{x}}\text{'} \quad & f_1(\boldsymbol{x}, \boldsymbol{y}) \\
\text{s.t.} \quad & g_1(\boldsymbol{x}, \boldsymbol{y}) \geq 0 \\
& \boldsymbol{y} \in \arg\min_{\boldsymbol{y}'} \quad f_2(\boldsymbol{x}, \boldsymbol{y}') \\
& \qquad\quad \text{s.t.} \qquad g_2(\boldsymbol{x}, \boldsymbol{y}') \geq 0
\end{aligned}
\tag{2.1}
$$

The above BLP is often referred to as a leader-follower model, where the leader takes the first move by controlling a set of decision variables $\boldsymbol{x}$ to maximize their objective function. The follower reacts to the leader's move by adjusting their own set of decision variables $\boldsymbol{y}$ to optimize the objective value. It is worth mentioning that $\boldsymbol{y}^{'}$ is just a dummy variable to replace $\boldsymbol{y}$ in the lower-level problem. The quotation marks in 2.1 are used to indicate the ambiguity in the formulation of the leader's problem. The ambiguity arises when the follower has to choose between more than one optimal reaction (i.e., the follower's problem is a non-strictly convex minimization problem that might have several alternative global optima, in which case, a problem is said to be *"degenerate"*). In order for the model to be well defined, the follower has to choose between alternative optima, which leads to two approaches; the optimistic approach (Dempe 2002) and the pessimistic approach (Aussel & Svensson 2019). Motivated by the need for modelling decentralized planning in many practical applications such as cross-dock truck scheduling (Konur & Golias 2013), facility location (Cao & Chen 2006), bi-level knapsack and capacitated lot-sizing (Lozano & Smith 2017), taxation and highway pricing (Labbé et al. 1998), interdiction games (Fischetti et al. 2019), defending critical infrastructure (Alvarez (2004), Alguacil et al. (2014), Fakhry et al. (2022)), and natural gas planning (Dempe et al. 2011), the work done by Bialas & Karwan (1984) was the first to call for the need of efficient and tractable algorithms for solving the bi-level mixed-integer linear problem (**BMILP**). Moore & Bard (1990) established that it is not possible to obtain tight upper-bounds from the natural relaxation of the bi-level problem. By providing examples and toy problems, they established that two of the three well known fathoming rules used in branch-and-bound (**B&B**) in single-level mixed-integer programming cannot be used in BMILP. In their influential work, Moore & Bard (1990) provided an implicit enumeration technique for finding bi-level feasible solutions, and a series of heuristics that offer a trade-off between quality and efficiency. In (Bard & Moore 1990), a B&B approach was suggested that makes use of exploiting the follower's Karush-Kuhn-Tucker (**KKT**) conditions; the algorithm enforces the underlying complementary slackness conditions suggested by Fortuny-Amat & McCarl (1981). Zeng & An (2014) also presented a computing scheme based on a decomposition strategy; by converting BMILP into a single-level reformulation and using an algorithm akin to the column-and-constraint generation algorithm. Xu & Wang (2014) presented an exact B&B algorithm with three simplifying assumptions for tractability. Kleniati & Adjiman (2014) presented an algorithm called branch-and-sandwich, in which two solution spaces corresponding to the first- and second-levels, are explored using a single B&B tree. In particular, two pairs of upper- and lower- bounds are computed: one for the objective function of the leader, and the other pair is for the follower's objective value. Motivated by recent efforts at that time, Fischetti et al. (2017) suggested a new branch-and-cut **B&C** algorithm for BMILP, in which they provided specific pre-processing strategies, valid linear inequalities, along with separation procedures. Recently, Tahernejad et al. (2020) presented a generalized B&C algorithmic framework for solving BMILPs; in which features from single-level and bi-level algorithms are combined. The aim was to produce a flexible and robust framework for solving a variety of different BMILPs. Furthermore, based on the fact that B&C

has proven to be more powerful than B&B in single-level mixed-integer optimization problems, Kleinert, Labbé, Plein & Schmidt (2021) were motivated to review existing cuts for linear bi-level problems, and introduced a new valid inequality that examines the strong duality constraint of the follower's level, and strengthened variants of the inequality derived from McCormick envelopes. Most recently, Liu et al. (2021) presented an enhanced branching rule based on the algorithm developed by Xu & Wang (2014); however, the new branching rule might discard bi-level feasible solutions if the lower-level problem possesses alternative optima, which may in-turn lead to bi-level feasibility (i.e., sub-optimality in BMILP).

From the perspective of bi-level integer linear programs **BILP**s, Bard (2013) presented an algorithm for the binary case for both leader and follower decision variables; this is done by converting the leader's objective function into a parametrized constraint and solving the re-formulated problem which produces a bi-level feasible solution. After which, improvements are gradually sought that eventually lead to the global optimum. DeNegre & Ralphs (2009) proposed a B&C approach for BILP, which improves on the B&B approach proposed by Bard & Moore (1990), by adding cutting planes that provide tighter bounds. It is worth mentioning that this approach does not require special branching strategies, and was implemented through publicly available linear solvers. Furthermore, using almost the same branching rules stated in (Xu & Wang 2014), but taking advantage of the integer requirements in BILP, Wang & Xu (2017) proposed the watermelon algorithm, in which a polyhedron is formed to encapsulate bi-level infeasible solutions. The complement of this polyhedron is then taken as disjunction hyperplanes in a B&B framework. Indeed, the area of including cuts or valid inequalities to bi-level programs is a fertile area for research, influential papers that discuss the use of valid inequalities and cuts include, but not limited to the work done by Fischetti et al. (2016), and Fischetti et al. (2018), which is based on relatively old developments in convexity cuts (Balas (1971), Glover (1973, 1974)).

Parametric programming approaches have also been used to solve bi-level quadratic and BMILPs such as the work done by Faísca et al. (2007), through inserting the rational reaction sets of the follower in the leader's problem, and transforming the bi-level problem into a set of independent quadratic, linear or mixed-integer linear problem that can be solved to optimality. Moreover, Mitsos (2010) proposed an algorithm for the global optimization of non-linear bi-level mixed-integer programs where it relies on a lower-bound obtained by solving mixed-integer non-linear programs, and generating a parametric upper-bound to the optimal solution function of the lower-level program.

The majority of the research done on bi-level programming deals with the optimistic case; that is, in case of a non-unique rational response (i.e., maximizes/minimizes payoffs) for the follower, the strategy that is in favour of the leader would be chosen. From a game-theory perspective, this is known as a strong Stackelberg game (Breton et al. 1988). On the other hand, if the follower picks a strategy that is against the leader's payoffs; this is considered a weak Stackelberg game (Loridan & Morgan 1996); which corresponds to a pessimistic two-level optimization problem

((Dempe 2002), (Liu et al. 2018)). It should be noted that Leitmann (1978) introduced the concept of a generalized Stackelberg game, accounting for non-unique followers' responses, after which Breton et al. (1988) introduced a formal definition for the strong-weak Stackelberg games. Furthermore, obtaining the optimality conditions for bi-level linear programming problem has been discussed in the literature under the assumption of uniqueness (Bard 1984), optimistic ((Dempe et al. 2006), (Gadhi & Dempe 2012), and pessimistic (Dempe et al. 2014) approaches. Since, bi-level programs are often re-formulated using KKT conditions for the lower-level problem– if it is a parametric convex optimization problem, resulting in a single-level mathematical program with complementary slackness conditions, the question of equivalence of both programs has been discussed in (Dempe & Dutta 2012) which turned out to depend on Slater's constraint qualification for the lower-level problem for the optimistic approach. The work done by Aussel & Svensson (2019) discusses the equivalence in the pessimistic situation.

For a comprehensive review of bi-level programming, solution approaches and practical applications, the interested reader may refer to the following reviews:

- Wen & Hsu (1991) recaps basic models, applications, solution approaches for the linear bi-level programming problems.
- Ben-Ayed (1993) gives a review of the features of linear bi-level programs, applications, algorithms and clarifies some confusing representations in the literature.

- Dempe (2003) provides some main directions of research highlighting re-formulated bi-level programs with complementary slackness conditions, difficulties arising from non-uniqueness of followers' optimal solutions, and on optimality conditions.

- The work done in (Colson et al. (2005, 2007)) gives an introductory survey of bi-level programs motivated by simple applications, main properties of different cases (e.g., linear-quadratic), and an overview of solution approaches.

- Lu et al. (2016) reviews multi-level decision-making with a focus on bi-level, however it discusses multi-objective and multi-follower situations.

- Liu et al. (2018) reviews the definitions, properties of the pessimistic bi-level optimization approach, and a follow-up with a discussion on solution approaches and some practical applications.

- Kleinert, Labbé, Ljubić & Schmidt (2021) reviews bi-level algorithmic approaches that make use of mixed-integer programming techniques with a focus on linear lower-level problems, followed-up by a review on solution approaches that solve mixed-integer bi-level problems with integer constraints in the follower's level, and end with a brief discussion on specific applications such as pricing and interdiction games. Future research questions pertaining to algorithmic development for bi-level optimization is provided.

Problems requiring a sequence of decisions in reaction to uncertainty realizations have been studied intensely over the past decades (Bakker et al. 2020). Multi-level optimization problems that deal with uncertainty have been often referred to as multi-stage problems in the literature. Since real world optimization problems often appear in a temporal context, the interplay between uncertainty and time is inherently important to any related decision-making process. The problem referred to requires a sequence of decisions which react to outcomes that evolve over time, and information on these outcomes is disclosed after the realization of a subset of probabilistic parameters (Birge & Louveaux 2011). Thus, multi-stage programmes differ from the classical Stackelberg multi-level programmes (von Stackelberg 2011) in the assumption of perfect and complete information. Since the information on the uncertain outcomes are being disclosed gradually, multi-stage optimization problems have incomplete information. Several reviews exist on optimization under uncertainty (Sahinidis 2004, Powell 2019), and application specific approaches such as project scheduling under uncertainty (Saharidis & Ierapetritou 2009), and supply chain network design (Govindan et al. 2017). Recently, Bakker et al. (2020) review the different methods for solving multi-stage optimization problems under uncertainty to pave the way for a holistic picture of sequential decision-making under uncertainty. In particular, Bakker et al. (2020) review the theoretical underpinnings of the different concepts for solving multi-stage optimization problems under uncertainty in the areas of mathematical programming and computer science. Indeed a large number of applications has been treated with more than one theoretical concept, e.g., robust optimization and stochastic dynamic programming. Yet few papers implement more than one concept at the same time (Bakker et al. 2020). It is worth mentioning multi-level programs, which is the focus of this study, have deterministic parameters and the underlying assumption is that players have perfect and complete information.

Now, we present a generic form of a TLP as follows:

$$\begin{aligned}
&\text{`}\min_{\boldsymbol{x}}\text{'} \quad f_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\\
&\text{s.t.} \quad g_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \geq 0\\
&\qquad \boldsymbol{y} \in \text{`}\arg\max_{\boldsymbol{y}'}\text{'} \quad f_2(\boldsymbol{x},\boldsymbol{y}',\boldsymbol{z})\\
&\qquad\quad \text{s.t.} \qquad g_2(\boldsymbol{x},\boldsymbol{y}',\boldsymbol{z}) \geq 0\\
&\qquad\qquad\qquad \boldsymbol{z} \in \arg\min_{\boldsymbol{z}'} \quad f_3(\boldsymbol{x},\boldsymbol{y}',\boldsymbol{z}')\\
&\qquad\qquad\qquad\quad \text{s.t.} \qquad g_3(\boldsymbol{x},\boldsymbol{y}',\boldsymbol{z}') \geq 0
\end{aligned} \tag{2.2}$$

The quotation marks in the first- and second- levels reflect the need to select a solution approach at the particular level where the marks are put. In particular, there is a need to resolve the solution approach at the first- and second- levels in case of multiple optima at the respective lower-levels, otherwise the TLP would be ill-posed. One of the contributions of this research is to provide a guideline on how a TLP can be classified, categorized, and most importantly how a solution approach or a strategy can be determined for a specific TLP. The rest of this research work is presented as follows: Section 2.3 presents the review and research methodology. Next, we present bibliometric analysis in Section 2.4. In Section 2.5 we introduce important definitions for TLPs. In Section 2.6 we discuss the most common algorithms used in TLP. Applications are presented in Section 2.7. Finally, future research directions and Conclusions are discussed in Sections 2.9 and 2.8.

2.3 Review and Research Methodology

In this section, we outline how we conducted our literature search in addition to the procedure used to filter the pool of research articles. We have classified the studied literature and present several summary statistics. Finally, inspired by Fahimnia et al. (2015), Mishra et al. (2018) and Ben-Daya et al. (2019), we include a bibliometric and network analysis of the reviewed pool of papers.

We have chosen the methodology of bibliometric analysis as it provides a quantitative and objective approach to analyze the literature by studying the citations and co-citations networks (Pilkington & Meredith 2009). To examine the current structure of research on TLPs, we performed citations analysis; this is a quantitative technique that provides a measure of the magnitude of influence of a research article in a specific field, which enables researchers to understand and identify the influential and major articles in a field. This is crucial to develop an understanding of how relevant an article is to the current research and how its popularity evolved through time. Co-citation analysis reveals the major research clusters within a field by tracing the connection between authors and their areas of research. It is worth mentioning that bibliometric analysis has been followed in fields that are related to ORMS such as information systems (Culnan 1986), strategic management (Nerur et al. 2008) and innovation (Cottrill et al. 1989). Furthermore, citation and co-citation analysis has been used to spot research trends and identify research gaps within ORMS (Pilkington & Fitzgerald 2006, Pilkington & Meredith 2009). Even for systematic literature review studies, citation and co-citation analysis have been used for pursuing an objective approach with the goal of narrowing down research domain classifications (Colicchia & Strozzi 2012, Fahimnia et al. 2015). In this research work, we adopt the same argument as of Pilkington & Meredith (2009) and Mishra et al. (2018) to identify highly cited research articles, in addition to revealing the structure of inter-relationships among articles which is the result of references' co-citations (i.e., articles that are usually cited together).

In order to focus on TLPs, we have chosen a combination of the words "tri-level optimization" (e.g., tri-level programming) or synonymous words such as "three-level optimization." In particular, we used in our search query a variation of the word "tri-level" with and without the hyphen. In addition to the word "three-level" with and without the hyphen. It is worth mentioning that using keywords to conduct a literature review has been used before in the work done by Eksoz et al. (2014) and Gunasekaran et al. (2015). We used Web of Science core collection to conduct our search and bibliometric analysis, since it includes only international scientific indexed (**ISI**) journals, that is generally accepted as a signal of journal quality (Garfield 1995, Chavarro et al. 2018). Using the following search query on Web of Science on *June 9th, 2022*: *tri-level optimization OR trilevel optimization OR "three-level optimization" OR tri-level programming OR trilevel programming OR "three-level programming"*; we ended up with 364 articles, book chapters and proceeding papers. We restricted our list to publications that appeared in ISI journals as the latter are considered as certified knowledge (Ramos-Rodríguez & Ruíz-Navarro 2004). After including only journal articles, we had 310 articles in the search database. We then refined our search and studied meticulously each paper to select the most relevant papers to TLPs. Mainly, we excluded papers that have the same keywords in the search query but not relevant to TLPs; we ended up with 277 articles in our literature database. In particular, articles related to neuroimaging, nanoscience nanotechnology, meteorology atmospheric sciences, metallurgy metallurgical engineering, materials science coatings films, linguistics, evolutionary biology, geosciences multidisciplinary, imaging science photographic technology, limnology, optics, physics condensed matter, physics fluids plasmas, political science, psychology, psychology biological, psychology clinical, regional urban planning, sociology, urban studies, ergonomics, engineering biomedical, education scientific disciplines, chemistry physical, radiology nuclear medicine medical imaging, materials science multidisciplinary, construction building technology, education educational research, environmental studies, instruments and instrumentation categories were excluded.

In the next section, we provide a bibliometric analysis for the 277 articles. Furthermore, in order to focus on papers published in ORMS journals, we carefully studied each paper to figure out whether it was published in an ORMS journal and we filtered out 53 papers that appeared in ORMS journals. We did so to offer a state of the art developments in the ORMS field and scope out relevant future research directions in that area. Additionally, we performed citation and co-citation analysis on those particular papers.

2.4 Bibliometric Analysis

We conducted a bibliometric analysis using the R programming (Tippmann 2015) language in R Studio, which is an integrated development environment for R (Allaire 2012). In particular, we used the bibliometrix package developed by Aria & Cuccurullo (2017) to conduct citation and co-citation analysis.

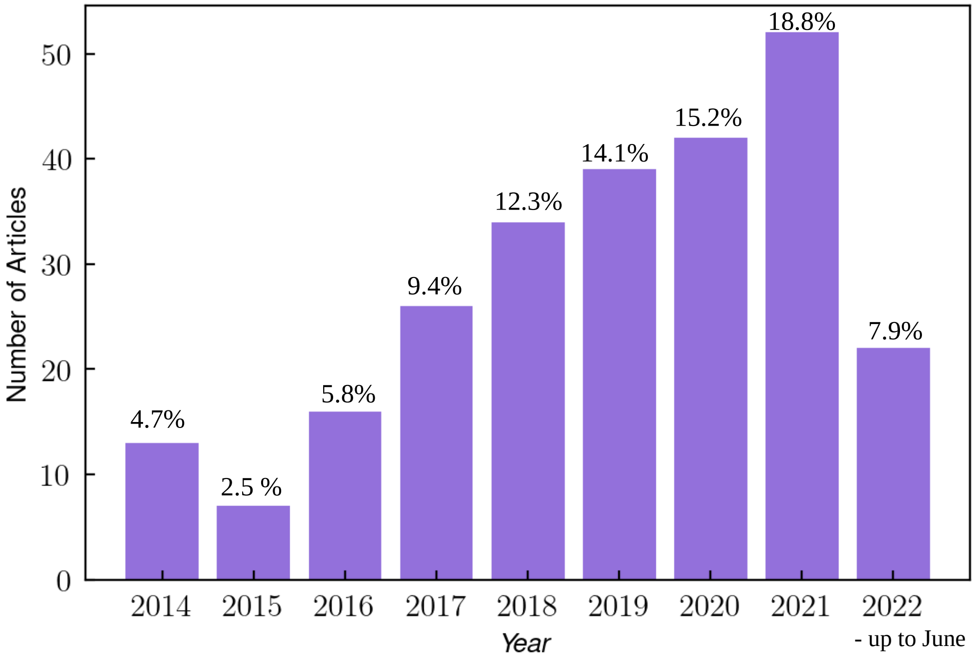


Figure 2.1: Number of Article Publications by Year.

Figure 2.1 shows the number of journal articles produced yearly in the core collection of Web of Science (**WoS**). Note that the year 2022 data is only up to the month of June. It can be clearly seen that the research work on TLPs is steadily increasing over the years, indicating the growing interest in the field.

Figure 2.2 is a time chart reflecting top-authors' productivity over time. The red line indicates the time span between the author's first published article and the last one pertaining to the topic of TLP. Furthermore, the number of articles published by a particular author in a year is proportional to the size of the ball depicted in that year; the largest ball size corresponds to 3 or more articles in a particular year. The total citation per year is also represented in the blue shade of the ball; we have three shade levels: light, medium, and heavy corresponding to 0, 10, and 20 total citations per year. In other words, if the ball representing an author is lightly shaded and small in size, then that particular author has 0 to 10 total citation per year and one article in that year.

In Table 2.1, we show the list of top authors that published three or more articles. Arroyo J.M. is the most productive with 10 articles, the majority of which pertain to electrical power networks such as electric power grid defence (Alguacil et al. 2014), transmission expansion planning (Moreira et al. 2014), energy reserve scheduling (Street et al. 2013, Moreira et al. 2015, Cobos et al. 2016, Cobos, Arroyo, Alguacil & Wang 2018), network expansion planning (Roldán et al. 2018, Munoz-Delgado et al. 2019) and the unit commitment problem (Cobos, Arroyo, Alguacil & Street 2018). It is worth mentioning that Arroyo co-authored six articles with Street A. who has seven articles in total, the second highest productive author in the collection. Moreover, the top five highest productive authors published mostly in electrical engineering journals with applications related to electrical power grid optimization. This highlights the interdisciplinary nature of TLP research. This review work aims to provide a unifying synthesis across these fields.

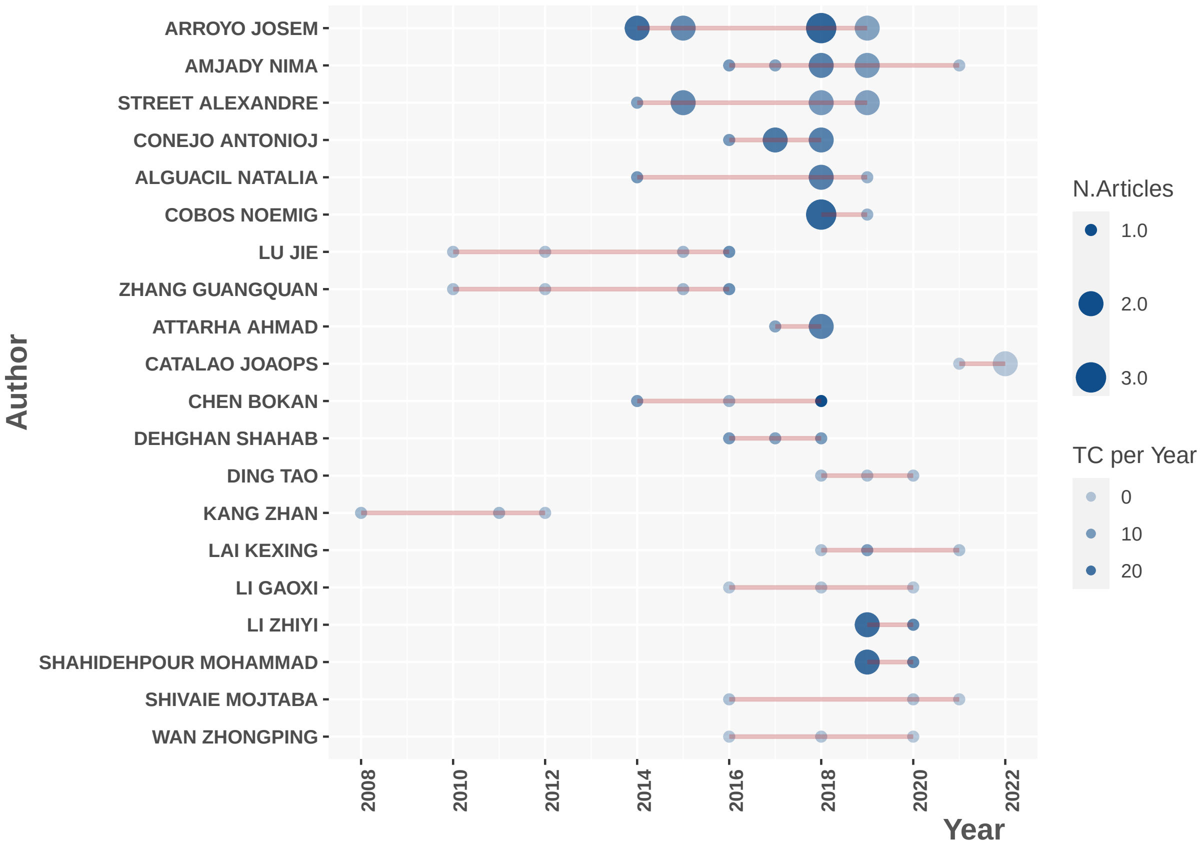


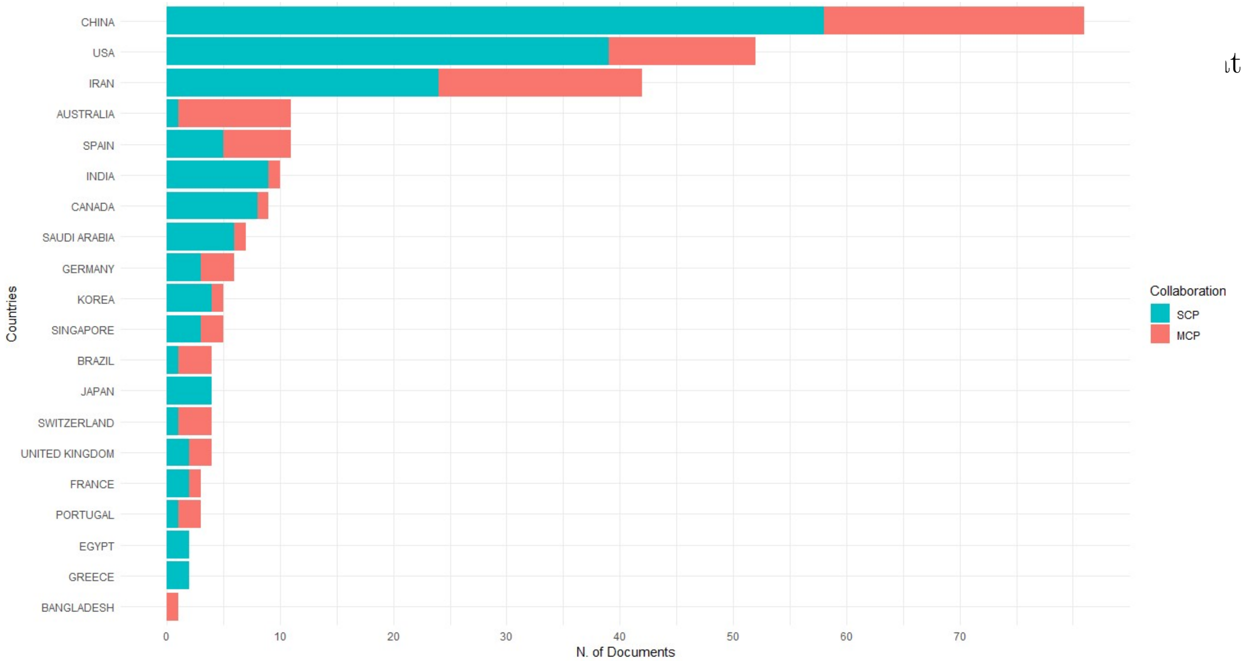
Figure 2.2: Top-Authors' Production over the Time.

Furthermore, a breakdown of the authors' countries is depicted in Figure 2.3. It can be clearly seen that China is the highest productive country in the field of TLPs, followed up by USA and Iran. In Figure 2.3, each country is represented by a bar depicting the overall publications, which are subsequently divided into single country publications (**SCP**) and multiple country publication (**MCP**). It is worth mentioning that China's SCP has surpassed USA's SCP and MCP combined; this is an indicator of the intensity by which China is trying to outperform USA in research. In addition, the fact that many Chinese and Iranian universities have instituted the

Author's Name	Record Count	% of 277
Arroyo JM	10	3.610
Amjady N	7	2.527
Street A	7	2.527
Conejo AJ	6	2.166
Li ZY	6	2.166
Lu J	5	1.805
Wang JH	5	1.805
Zhang GQ	5	1.805
Alguacil N	4	1.444
Catalao JPS	4	1.444
Cobos NG	4	1.444
Dehghan S	4	1.444
Liu Y	4	1.444
Attarha A	3	1.083
Chen BK	3	1.083
Ding T	3	1.083
Han JL	3	1.083
Hu YG	3	1.083
Kang Z	3	1.083
Lai KX	3	1.083
Li GX	3	1.083
Sadeghi H	3	1.083
Shafie-khah M	3	1.083
Shahidehpour M	3	1.083
Shivaie M	3	1.083
Tavakkoli-moghaddam R	3	1.083
Vidyarthi N	3	1.083
Wan ZP	3	1.083
Wang DZW	3	1.083
Wang FS	3	1.083
Wang LZ	3	1.083
Wu WH	3	1.083

Table 2.1: Top Authors of the Reviewed Publications.

ıte



SCP: Single Country Publications, MCP: Multiple Country Publications

Figure 2.3: Most Productive Countries.

In Table 2.2, we show a breakdown of the reviewed publications according to research areas. Engineering, which includes mostly electrical engineering, is the highest research area followed up by computer science and ORMS, respectively. These areas are known to utilize TLPs in their applications. This is also backed up by the journal titles of the selected articles shown in Table 2.3.

Table 2.3 lists the journals with the highest record counts of published articles in our search query; all journals are related to electrical engineering except for the European Journal of Operational Research, which is categorized as an ORMS journal. IEEE Transactions on Power Systems is the highest journal publishing peer-reviewed articles in the field of TLPs with a record count of 26 articles.

Research Area	Record Count	% of 277
Engineering	182	65.704
Computer Science	60	21.661
Operations Research Management Science	52	18.773
Energy Fuels	47	16.968
Business Economics	24	8.664
Mathematics	23	8.303
Science Technology Other Topics	18	6.498
Telecommunications	13	4.693
Transportation	12	4.332
Automation Control Systems	6	2.166
Environmental Sciences Ecology	5	1.805
Physics	5	1.805
Thermodynamics	5	1.805

Table 2.2: Research Areas of the Reviewed Publications.

Publication Titles	Record Count	% of 277
IEEE Transactions on Power Systems	26	9.386
Applied Energy	17	6.137
International Journal of Electrical Power Energy Systems	13	4.693
IEEE Transaction on Smart Grid	12	4.332
European Journal of Operational Research	10	3.610
IET Generation Transmission Distribution	9	3.249
IEEE Access	8	2.888
Reliability Engineering System Safety	8	2.888
IEEE Transaction on Sustainable Energy	7	2.527

Table 2.3: Journal Titles of the Reviewed Publications.

2.4.1 Citation analysis

In this section, we report on the influential works as well as the state of collaboration in the field of TLPs.

Influential Works

In an effort to identify influential research articles that shaped TLPs, we have compiled a list of the top 20 highly-cited research articles in our search pool as shown in Table 2.4. It is worth mentioning that we only included citations that are counted by WoS; these citations tend to be lower in count than that of Google Scholar. This is mainly due to two reasons: the first being a tendency for Google Scholar to inflate citation counts due to inclusion of non-scholarly resources (e.g., promotional pages, course reading lists, etc.), and the second is the tendency of WoS to have errors in citations provided by authors, and different citation styles used by journals; which might subsequently lead to poor indexing.

The most highly-cited research article by far in TLPs is the influential article by Brown et al. (2006) collecting 415 citation counts. This particular article was published in Interfaces, which is a known ORMS journal for practical applications of operations research. This study discusses defending critical infrastructure and was the first of its kind studying the defender-attacker and defender-attacker-defender models as a bi-level and tri-level mathematical programs, respectively. A list of the top 25 highly-cited reviewed research articles in ORMS journals is shown in Table 2.5. In the coming sections, we will discuss how those articles influenced and shaped TLPs in terms of applications and solution approaches.

Publication Author(s)- Year	Citation Counts	Research Area
(Brown et al. 2006)	415	Operations Research & Management Science
(Mendes et al. 2011)	158	Energy & Fuels
(Ma et al. 2016)	143	Engineering
(Koltsaklis & Dagoumas 2018)	128	Energy & Fuels-Engineering
(Liberatore et al. 2012)	128	Operations Research & Management Science
(Ruiz & Conejo 2015)	124	Operations Research & Management Science
(Roghanian et al. 2007)	111	Mathematics
(Yuan et al. 2014)	105	Engineering-Operations Research & Management Science
(Lin & Bie 2018)	101	Energy & Fuels-Engineering
(Alguacil et al. 2014)	100	Operations Research & Management Science-Engineering
(Chen et al. 2014)	95	Engineering
(Lu et al. 2016)	88	Computer Science- Operations Research & Management Science
(Moreira et al. 2014)	87	Engineering, Electrical & Electronic
(Street et al. 2013)	84	Engineering
(Yao et al. 2007)	79	Computer Science- Engineering
(Liu & Wang 2017)	77	Engineering- Operations Research & Management Science
(Wang et al. 2016)	75	Engineering
(Fang & Sansavini 2017)	75	Engineering- Operations Research & Management Science
(Dehghan et al. 2015)	73	Engineering
(Yan et al. 2018)	72	Energy & Fuels-Engineering

Table 2.4: Top 20 Highly-Cited Research Articles of the Reviewed Publications.

Publication Author(s)- Year	Citation Counts	Application Area
(Brown et al. 2006)	415	Defense Multi-Disciplinary
(Liberatore et al. 2012)	128	Defense Multi-Disciplinary
(Ruiz & Conejo 2015)	124	Electric Grid Applications
(Yuan et al. 2014)	105	Defending Electrical Systems
(Alguacil et al. 2014)	100	Defending Electrical Systems
(Lu et al. 2016)	88	Review Article
(Liu & Wang 2017)	77	Transportation
(Fang & Sansavini 2017)	75	Defending Electrical Systems
(Ouyang et al. 2017)	35	Defending Electrical Systems
(White 1997)	27	Theory Development
(Ghorbani-Renani et al. 2020)	26	Defending Electrical Systems
(Jin et al. 2015)	25	Transportation
(Sarhadi et al. 2017)	22	Transportation
(Fanzeres et al. 2019)	16	Auction Markets
(Han et al. 2016)	16	Defense Multi-Disciplinary
(Aussel et al. 2020)	15	Electric Grid Applications
(Ramamoorthy et al. 2018)	15	Defense Multi-Disciplinary
(Ke & Bookbinder 2018)	14	Discount Policies
(Florensa et al. 2017)	14	Capacity Planning
(Ding et al. 2018)	13	Defending Electrical Syste,s
(Wu et al. 2011)	13	Transportation
(Lei et al. 2018)	12	Defense Multi-Disciplinary
(Rahdar et al. 2018)	10	Inventory Planning
(Parajuli et al. 2017)	10	Defense Multi-Disciplinary
(Avraamidou & Pistikopoulos 2019)	8	Theory Development

Table 2.5: Top 25 Highly-Cited Reviewed Research Articles in Operations Research & Management Science Journals.

2.4.2 Co-citation and Collaboration Analysis

In order to identify research groups who are active in TLPs, we performed a collaboration analysis between authors as shown in Figure 2.4. This is beneficial in terms of understanding how research groups interact and add to the growing field of TLPs.

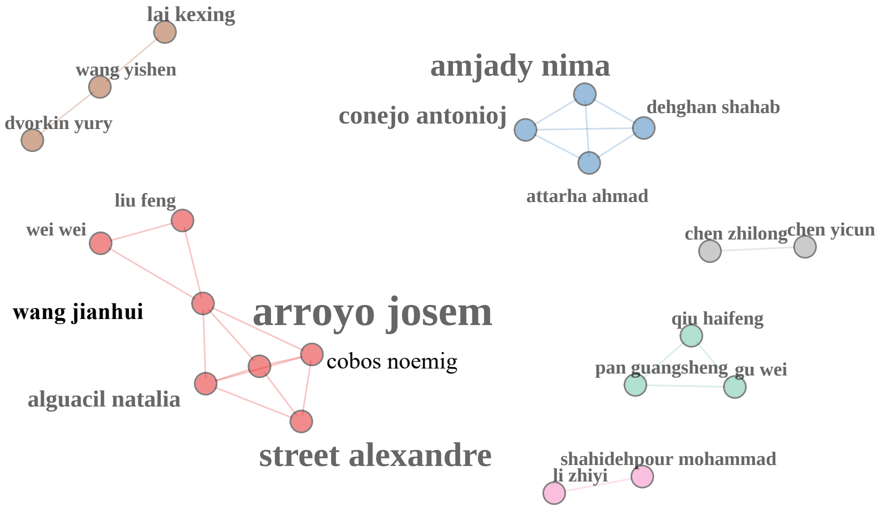


Figure 2.4: Author's Collaboration

Figure 2.4 shows that the largest research groups are the red and blue clusters. An edge between two nodes resembles a collaboration in a research article between two authors. Each node is accompanied with an author's name in lower case. The number of collaborations is directly proportional with author's name font size. In particular, the red research cluster representing Arroyo J.M. and Street A.'s research groups collaborating with Wang J.H.'s research group on applications related to electrical power networks. It is worth mentioning that Arroyo J.M. and Street A. are among the three highest productive authors in the collection, which explains the authors' names font size. Furthermore, the blue research cluster representing Conejo A.J., Amjady N., Deghan S. and Attarha A. are forming a research group collaborating on applications related to the optimization of micro-grids and electric power grids.

Figure 2.5 depicts the co-citation between authors in all reviewed articles resulting in three main clusters. The thickness of the edge between two cited authors indicates how frequently they are cited together, where the size of the sphere carrying the authors' name indicates how frequently the corresponding author is cited. Consequently, an author with many thick edges has to have a big sphere. The red cluster includes references that are usually cited in electrical engineering applications such as Fortuny-Amat & McCarl (1981), who introduced systematic linearization techniques for the reduction of TLPs. It is worth mentioning that "anonymous," which is highly cited in the red cluster, refers to the electrical power grid data used by authors to validate their proposed modelling and solution approaches on standard electrical networks. From Figure 2.5, we see that data is heavily cited by all three clusters. This also establishes the dominance of electrical engineering applications over the field of TLPs. The blue cluster in Figure 2.5 reflects the defender-attacker-defender theme. In particular, authors such as Arroyo J.M., Alguacil N., Scaparra M.P., Wood K., Brown G., Salmeron J. and Wu X. have discussed the defender-attacker-defender model in at least one article; this is why they are cited together, however they might have applied the model in different applications. Moreover, since Dempe (2002) and Bard (2013) are two of the most influential authors in bi-level programming, they are co-cited frequently in that cluster, as bi-level programming is a corner stone of TLPs. Lastly, the green cluster reflects methodological contributions needed for solving TLPs such as column-and-constraint generation and Benders decomposition.

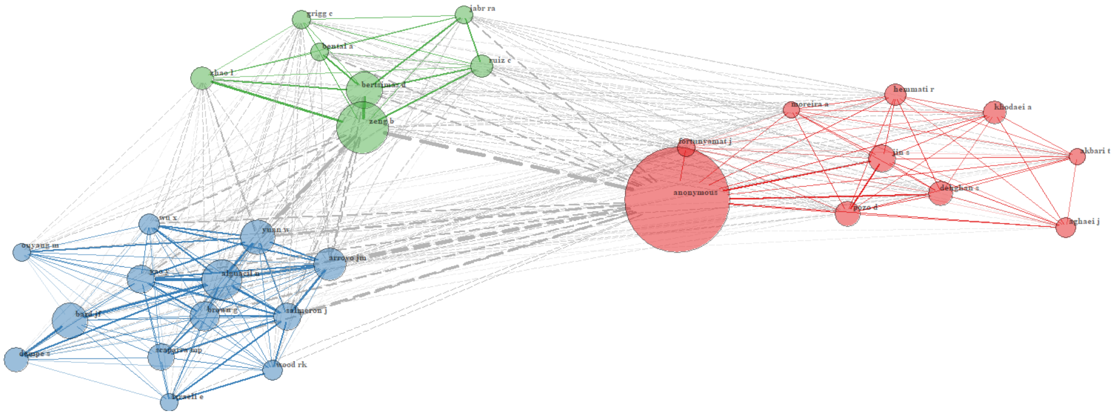


Figure 2.5: Author's Co-citation in the Reviewed Publications.

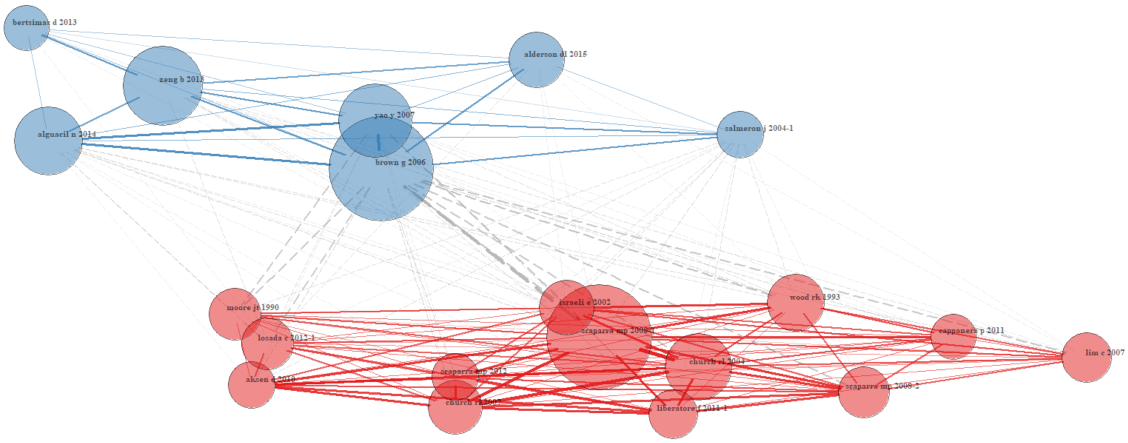


Figure 2.6: Co-citation References Network in the Operation Research & Management Science Reviewed Publications.

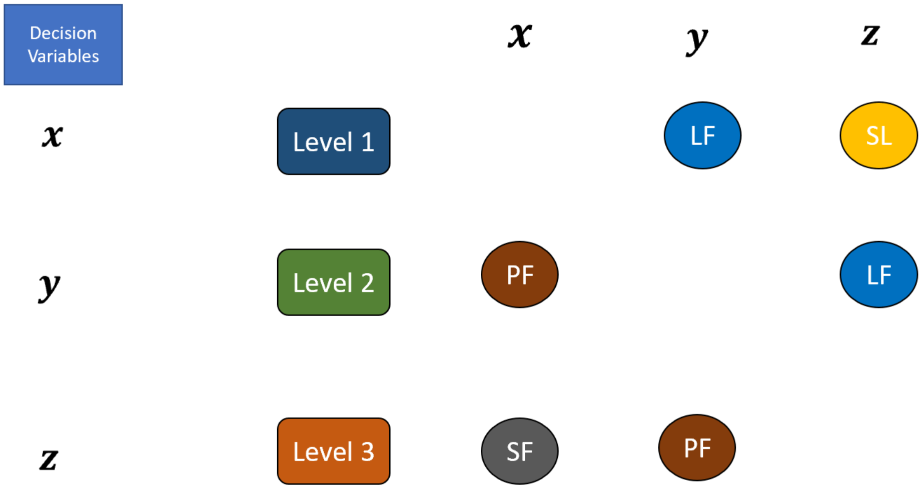
Figure 2.6 reflects co-citations between the most co-cited 20 references in the ORMS journal articles; this figure helps in understanding the references that are cited together in ORMS journal articles. Consequently, it forms the core knowledge contributing to the field of TLPs. In particular, Scaparra M.P. has co-authored 5 articles (Church & Scaparra 2007, Scaparra & Church 2008, Cappanera & Scaparra

2011, Scaparra & Church 2012, Liberatore et al. 2012) forming the core of the red cluster in Figure 2.6; these articles discuss bi-level and tri-level interdiction problems. This explains the association with the article done by Israeli & Wood (2002) that discusses shortest-path network interdiction, along with the research done by Moore & Bard (1990) proposing an algorithm for solving mixed-integer bi-level linear problems. At the core of the blue cluster, the article by Brown et al. (2006), that proposes bi-level and tri-level mathematical formulations for defending critical infrastructure, is co-cited with references (Salmeron et al. 2004*a*, Yao et al. 2007, Alguacil et al. 2014) which propose analysis for defending electric power grids. Lastly, the article by Zeng & An (2014) is co-cited with attacker-defender (Salmeron et al. 2004*b*) or defender-attacker-defender models (Yao et al. 2007, Alguacil et al. 2014) as it proposes a decomposition approach for solving bi-level mixed-integer problems.

2.5 Taxonomy of Papers

2.5.1 Interactions between Decision-Levels

This section introduces new definitions for identifying relationships among decision-makers in a TLP through direct interaction of decision variables of each entity/level with other levels. For the sake of clarity, we will be referring to the three levels in a TLP as the first-, second-, and third-levels, respectively. Figure 2.7 summarizes the definitions for direct interactions between decision levels in a TLP. An interaction between two levels means the presence of decision variables of a level in the objective function or constraints of the other level. Each interaction has been given an acronym, which is introduced in the definitions.



Decision Variables		$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$
$\boldsymbol{x}$	Level 1		LF	SL
$\boldsymbol{y}$	Level 2	PF		LF
$\boldsymbol{z}$	Level 3	SF	PF	

Figure 2.7: Definitions for Direct Interactions between Decision Entities in a TLP.

Definition 2.5.1 (Leader-Follower, **LF**)**.** *If the decision variables of an entity appear in the objective function and/or constraints of the immediate upper-level (i.e., hierarchically-higher) decision-maker, then that entity/level is said to have a LF relationship.*

In a TLP, the first-level (second-level) entity is said to have a LF relationship, if the second-level (third-level) decision variables appear in the objective function and/or constraints of the first-level (second-level). Figure 2.7 can be visualized as a 3 × 3 matrix, where the rows and columns are the decision vectors of each level. Decision vectors $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ correspond to the first-, second-, and third-levels respectively. Hence, a LF relationship, which is marked by blue in Figure 2.7, can occur in a TLP if $\boldsymbol{y}$ is present in the first-level, or if $\boldsymbol{z}$ is present in the second-level.

Definition 2.5.2 (Secondary-Leadership, **SL**)**.** *If the decision variables of the third-level entity (i.e., $\boldsymbol{z}$) in a TLP appear in the objective function and/or constraints of the first-level entity, then the first-level is said to have a SL relationship.*

Definition 2.5.3 (Primary-Follwership, **PF**)**.** *If the decision variables of an entity appears in the objective function and/or constraints of the immediate lower-level (i.e., hierarchically-lower decision-maker, then that entity/level is said to have a PF relationship .*

In a TLP, the second-level (third-level) entity is said to have a PF relationship, if the first-level (second-level) decision variables appear in the objective function and/or constraints of the second-level (third-level). Hence, a PF relationship, which is marked by brown in Figure 2.7, can occur in a TLP if $\boldsymbol{x}$ is present in the second-level, or if $\boldsymbol{y}$ is present in the third-level.

Definition 2.5.4 (Secondary-Followership, **SF**)**.** *If the decision variables of the top-level entity (i.e., $\boldsymbol{x}$) in a tri-level program appears in the objective and/or constraints of the third-level entity, then the third-level is said to have a SF relationship.*

These definitions can be used to formally describe the interactions between decision-makers in a TLP through abbreviating each level. For instance, if $\boldsymbol{y}$, and $\boldsymbol{z}$ are present in the first-level, then the interactions in the first level can be abbreviated with LF-SL. Furthermore, these definitions can also be used for describing the interactions in a bi-level program, in which case we can imagine a 2×2 matrix by omitting the last column and row in Figure 2.7. For example, the attacker-defender model can be described as “-/PF”, since there are no direct interactions with the first-level a

"-" has been used to describe the interactions with the first-level, the "/" means the abbreviations that are coming next belong to the second-level decision-maker, and finally "PF" which indicates that the second-level has a primary-followership. Consequently, we can abbreviate the interactions in any TLP through mentioning the abbreviations separated by "/" indicating the next level's interaction in a manner similar to describing queuing models. This notation can also be extended to multi-level programming.

2.5.2 Degeneracy

Degeneracy in Bi-Level Programming Problems

It is widely known that a non-strictly convex minimization problem might have several alternative global optima, in which case, a problem is said to be *degenerate*. Bi-level programs belong to problems of hierarchical optimization of the form:

$$\text{`}\min_{\boldsymbol{x}}\text{'}\ \{f_1(\boldsymbol{x},\boldsymbol{y})|\ \boldsymbol{x}\in X,\ \boldsymbol{y}\in S(\boldsymbol{x})\}, \tag{2.3}$$

where the first-level player (i.e., leader) aims to minimize their cost function f_1 with respect to the variable $\boldsymbol{x}$, taking into consideration the reaction $\boldsymbol{y}$ of the follower (i.e., second-level decision-maker). Here $S:\ X\subset\mathbb{R}^n\rightrightarrows\mathbb{R}^m$ is a set-valued mapping defined by:

$$S(\boldsymbol{x}):=\arg\min_{\boldsymbol{y}}\{f_2(\boldsymbol{x},\boldsymbol{y})|\ \boldsymbol{y}\in K(\boldsymbol{x})\}, \tag{2.4}$$

which describes sets of optimal solutions of the lower-level parametric problem:

$$\min_{\boldsymbol{y}}\{f_2(\boldsymbol{x},\boldsymbol{y})|\ \boldsymbol{y}\in K(\boldsymbol{x})\}, \tag{2.5}$$

for any given choice $\boldsymbol{x} \in X$ of the leader. The sets X and $K(\boldsymbol{x})$ are called first-level and second-level feasibility sets, respectively. For the sake of clarity, we restrict ourselves to the case where first- and second-level constraint sets are given explicitly as:

$$X \ := \{\boldsymbol{x}\in\mathbb{R}^n|g_1(\boldsymbol{x})\leq 0\} \text{ and } K(\boldsymbol{x}) \ := \{\boldsymbol{y}\in\mathbb{R}^m|g_2(\boldsymbol{x},\boldsymbol{y})\leq 0\}, \tag{2.6}$$

respectively, with $g_1 : \mathbb{R}^n \to \mathbb{R}^k$ and $g_2 : \mathbb{R}^n\times\mathbb{R}^m \to \mathbb{R}^p$. Moreover, $f_1 : \mathbb{R}^n\times\mathbb{R}^m \to \mathbb{R}$ and $f_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are single-valued first- and second-levels objective functions, respectively. The general bi-level program presented in 2.3 can be recast as a set-valued optimization problem:

$$\min_{\boldsymbol{x}\in X} f_1(\boldsymbol{x}, S(\boldsymbol{x})) \ := \bigcup_{\boldsymbol{y}\in S(\boldsymbol{x})}\{f_1(\boldsymbol{x},\boldsymbol{y})\}, \tag{2.7}$$

where the minimization is considered with respect to some ordering cone (Dempe et al. 2006). From the perspective of scalar optimization, problem 2.3 becomes a regular optimization problem:

$$\min\{f_1(\boldsymbol{x}, S(\boldsymbol{x}))|\boldsymbol{x}\in X\}, \tag{2.8}$$

provided that $S(\boldsymbol{x})$ is single-valued $\forall\ \boldsymbol{x} \in X$. In other words, for every $\boldsymbol{x} \in X$, solution of problem 2.5 can be uniquely determined (i.e., global optimum). The single quotation marks in problem 2.3 is used to indicate the ambiguity in determining the follower's solution in-case of non-uniqueness, which makes the bi-level program defined by 2.3 and 2.4, ill-posed (Dempe 2002). In order to reflect real-world problems, the follower should have the freedom of choice that reflects their best interest, whether it suits the first-level decision maker or not; this gives the rise to two solution approaches commonly know as: *optimistic* and *pessimistic* approaches. The *optimistic* approach that deals with ill-posed bi-level problems can be modeled as follows:

$$(\mathbf{P}_o)$$

$$\min_{\boldsymbol{x}} \quad \varphi_o(\boldsymbol{x}) \tag{2.9}$$

$$\text{s.t.} \quad \boldsymbol{x} \in X \tag{2.10}$$

$$\varphi_o(\boldsymbol{x}) = \quad \min_{\boldsymbol{y}} \quad f_1(\boldsymbol{x}, \boldsymbol{y}) \tag{2.11}$$

$$\text{s.t.} \quad \boldsymbol{y} \in S(\boldsymbol{x}) \tag{2.12}$$

It is worth noting that both approaches (i.e., *optimistic* and *pessimistic*) yield a TLP, since $S(\boldsymbol{x})$ is defined in 2.4 with possible multiple solutions for some $\boldsymbol{x} \in X$. Problem $(\mathbf{P}_o)$ simulates a situation where a cooperation between upper-and lower-levels is allowed. An interesting variation of problem $(\mathbf{P}_o)$ is:

$$(\mathbf{P}_o^{'})$$

$$\min_{\boldsymbol{x},\boldsymbol{y}} \quad f_1(\boldsymbol{x},\boldsymbol{y}) \tag{2.13}$$

$$\text{s.t.} \quad \boldsymbol{x} \in X \tag{2.14}$$

$$\boldsymbol{y} \in S(\boldsymbol{x}). \tag{2.15}$$

Problem $(\mathbf{P}_o^{'})$ can be seen as a regularization of $(\mathbf{P}_o)$, where well-posedness is implicitly assumed, given that the difficulty in objective function of $(\mathbf{P}_o)$ is essentially moved to the constraint set in $(\mathbf{P}_o^{'})$, where the upper-level player has full control over both upper- $(\boldsymbol{x})$ and lower-levels' $(\boldsymbol{y})$ decision variables (Zemkoho 2016). It is paramount to mention that a local optimum solution of $(\mathbf{P}_o)$ implies a local optimum solution of $(\mathbf{P}_o^{'})$, however the converse is not necessarily true (Dempe 2002). Nevertheless, problem $(\mathbf{P}_o)$ and $(\mathbf{P}_o^{'})$ are globally equivalent.

For the *pessimistic* approach, the bi-level program can represented as:

$(\mathbf{P}_p)$

$$\min_{\boldsymbol{x}} \quad \varphi_p(\boldsymbol{x}) \tag{2.16}$$

$$\text{s.t.} \quad \boldsymbol{x} \in X \tag{2.17}$$

$$\varphi_p(\boldsymbol{x}) = \quad \max_{\boldsymbol{y}} \quad f_1(\boldsymbol{x},\boldsymbol{y}) \tag{2.18}$$

$$\text{s.t.} \quad \boldsymbol{y} \in S(\boldsymbol{x}). \tag{2.19}$$

Problem $(\mathbf{P}_p)$ reflects the fact that leader and follower can be adversaries. Hence, it is necessary for the first-level decision-maker to withstand damages resulting from undesirable actions of the second-level decision-maker. In general, *pessimistic* and *optimistic* solution approaches have been defined to determine which lower optima will be selected that could be against or in favour-of the leader (Wiesemann et al. (2013);Zemkoho (2016);Dempe et al. (2014)). Moreover, some algorithms have been designed to find one type of solution or the other (Zemkoho (2016);Kleniati & Adjiman (2014)). Furthermore, some algorithms have been tuned to analyze different approaches with regards to specific applications. For example, Konur & Golias (2013) have used genetic algorithms to solve a bi-level program for cross-dock truck scheduling with truck arrival time uncertainty with three different strategies: (1) deterministic approach which assumes mid-arrival time windows as expected truck arrival times, (2) pessimistic approach which assumes worst truck arrival times, and (3) optimistic approach which assumes best truck arrival times. For small-scale numerical examples illustrating the differences between ($\mathbf{P}_p$) and ($\mathbf{P}_o$), the interested reader can refer to (Zemkoho 2016).

Degeneracy in Tri-Level Programs

In most studies on TLP, it is implicitly assumed that the optimal solutions of the lower-levels' objectives for each decision made at the upper-levels are unique (Yao et al. (2007), Alguacil et al. (2014), Sarhadi et al. (2017), Schweitzer & Medal (2019)). This is generally not always true, as any non-strictly convex (concave) minimization (maximization) might have multiple optimal solutions. In a tri-level problem, the selection of alternative optima (i.e., degenerate solutions) at a particular level yields the same results for that level. However, each of the alternatives has a different impact on the overall problem (Florensa et al. 2017). This is why it is important to determine the selection criteria for the upper-level decision-maker(s) among the different solution alternatives in the lower-levels, otherwise the tri-level model would be ill-posed. A tri-level program without selecting an appropriate selection approach (i.e., ill-posed) can be modelled as follows:

$$
\begin{aligned}
&\text{`}\min_{\boldsymbol{x}}\text{'} \quad f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) && (2.20)\\
&\text{s.t.} \quad g_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \geq 0 && (2.21)\\
&\qquad \boldsymbol{y} \in \text{`}\arg\max_{\boldsymbol{y}'}\text{'} \quad f_2(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) && (2.22)\\
&\qquad\qquad \text{s.t.} \quad g_2(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) \geq 0 && (2.23)\\
&\qquad\qquad\qquad \boldsymbol{z} \in \arg\min_{\boldsymbol{z}'} \quad f_3(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}') && (2.24)\\
&\qquad\qquad\qquad\qquad \text{s.t.} \quad g_3(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}') \geq 0 && (2.25)
\end{aligned}
$$

Similar to the representation of problem 2.3, single quotation marks on the '$\min$' and '$\arg\max$' are to indicate ill-posedness of the model (Zemkoho 2016). As each player is in-charge of its own decision variables, A strategy needs to be set-up for the first-level and second-level decision-makers in case of multiple global optimal solutions. In general, research work discussing degeneracy in tri-level programs is very scarce. Florensa et al. (2017) have introduced new definitions that are required for the analysis of degeneracy of tri-level programs. The authors proposed a tri-level mixed-integer linear program to model capacity planning decisions in a duopoly considering the conflicting interests of three rational decision-makers. The definitions proposed by Florensa et al. (2017) cleared a part of the ambiguity in the characterization of optimal solutions in tri-level programs; by providing extensions of *optimistic* bi-level programs and application-specific algorithms tailored to finding the corresponding *optimistic* optimal solutions.

Recently, Aussel et al. (2020) devised a tri-level single-leader-multi-follower model for demand-side management. First, the authors took advantage of the special structure of their model by converting the tri-level model into a bi-level problem using explicit formulas for the third-level and plugging them into the second-level problem. They established equivalence between the resulting bi-level program and the the original tri-level program by proving the uniqueness of the third-level solution in addition to validation of the Slater's constraint qualification according to (Dempe & Dutta 2012). Second, the resulting bi-level program was converted into a single-level using the classical transformation of replacing the followers' problems by their KKT conditions in the leader's problem. Indeed, another equivalence should be established in order to assure the equivalence between the original bi-level and the resulting single-level problem, *even if the lower-level problems of the leader problem are convex*, as proved by Dempe & Dutta (2012) for the *optimistic* case and Aussel & Svensson (2019) for the *pessimistic* case. Furthermore, since uniqueness of the most lower-level solution was proved, the authors classified their solution approaches for the tri-level demand side management model into three categories: (1) *optimistic approach* which is the same as the classical optimistic approach, where the solution that best serves the upper-level's objective function is selected, (2) *revisited optimistic* where the leader gets to maintain the optimal value of their objective function based on certain application-specific assumptions (i.e, no interactions/energy exchanges within the intermediary level), and (3) *semi-optimistic* approach, which is also based on certain application-specific assumptions, where the solution is characterized to be in the middle spectrum of the *optimistic* and *pessimistic* approaches.

To set a framework for a decision criteria under degeneracy, selection approaches must be defined at hierarchically-higher decision-making levels (i.e., first- and second-levels) in TLPs. For that purpose, given the tri-level program presented in (2.20)-(2.25), a group of definitions have to be introduced building on the notations introduced in (Florensa et al. 2017). Figure 2.8 summarizes the definitions that are later introduced in this section. As explained earlier, degeneracy is resolved if multiple optimal solutions exist for the second- and third-levels. A resolution for the third-level's degeneracy is marked by a solid arrow, while a resolution for the second-level's degeneracy is marked by a dashed arrow in Figure 2.8. The double-lined arrow denotes a strategic resolution; this means that both second- and third-levels co-operate in-favour-of/against the first-level. The arrow in all cases points to the level in which the degeneracy is resolved whether it is in favour or against the decision-maker in that level. The black and red colours refer to optimistic and pessimistic resolutions, respectively. The first row in Figure 2.8 depicts the variation of optimistic approaches that occur in TLP. For example, a *sequentially optimistic* approach means a resolution of the third-level's degeneracy is in favour-of the second-level, while the second-level's degeneracy is resolved in favour-of the first-level. Thus, a solid black arrow is pointing to the second-level from the third-level, and a dashed black arrow is pointing to first-level from the second-level in Figure 2.8. Thus, arrows are black in the purely optimistic approaches (i.e., first row in Figure 2.8), red in the purely pessimistic approaches (i.e., second row in Figure 2.8), black and red in the mixed approaches (i.e., third row in Figure 2.8). The last row depicts mixed optimistic and pessimistic approaches. In particular, an approach that is textitsequential implies that degeneracy of the third-level is resolved in-favour-of/against the second-level, and degeneracy of the second-level is resolved in-favour-of/against the first-level. For instance, *sequentially optimistic-pessimistic* approach means degeneracy of the third-level is resolved in favour-of the second-level (i.e., solid black arrow), while degeneracy of the second-level is resolved against the first-level (i.e., red dashed arrow). Lastly, an approach that is *hierarchical* implies that degeneracy in second- and third-levels are resolved in favour-of/against the first-level; this is different from the *strategic* approach, as there is no cooperation between the second- and third-levels' decision-makers.

The rest of this section is dedicated for introducing the mathematical representations and definitions for each resolution approach in Figure 2.8, given the TLP presented in (2.20)-(2.25).

Definition 2.5.5 (Tri-Level Constraint Region, Ω)**.** *let Ω be the tri-level constraint region:*

$$\Omega = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in X \times Y \times Z : \;\; g_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \geq 0, \; g_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \geq 0, \; g_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \geq 0\} \tag{2.26}$$

Definition 2.5.6 (Third-Level Constraint Region, $\Omega_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y})$)**.** *Let $\Omega_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y})$ be the third-level constraint region for fixed $(\boldsymbol{x}, \boldsymbol{y}) \in X \times Y$:*

$$\Omega_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}) = \{\boldsymbol{z} \in Z : \; g_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \geq 0\} \tag{2.27}$$

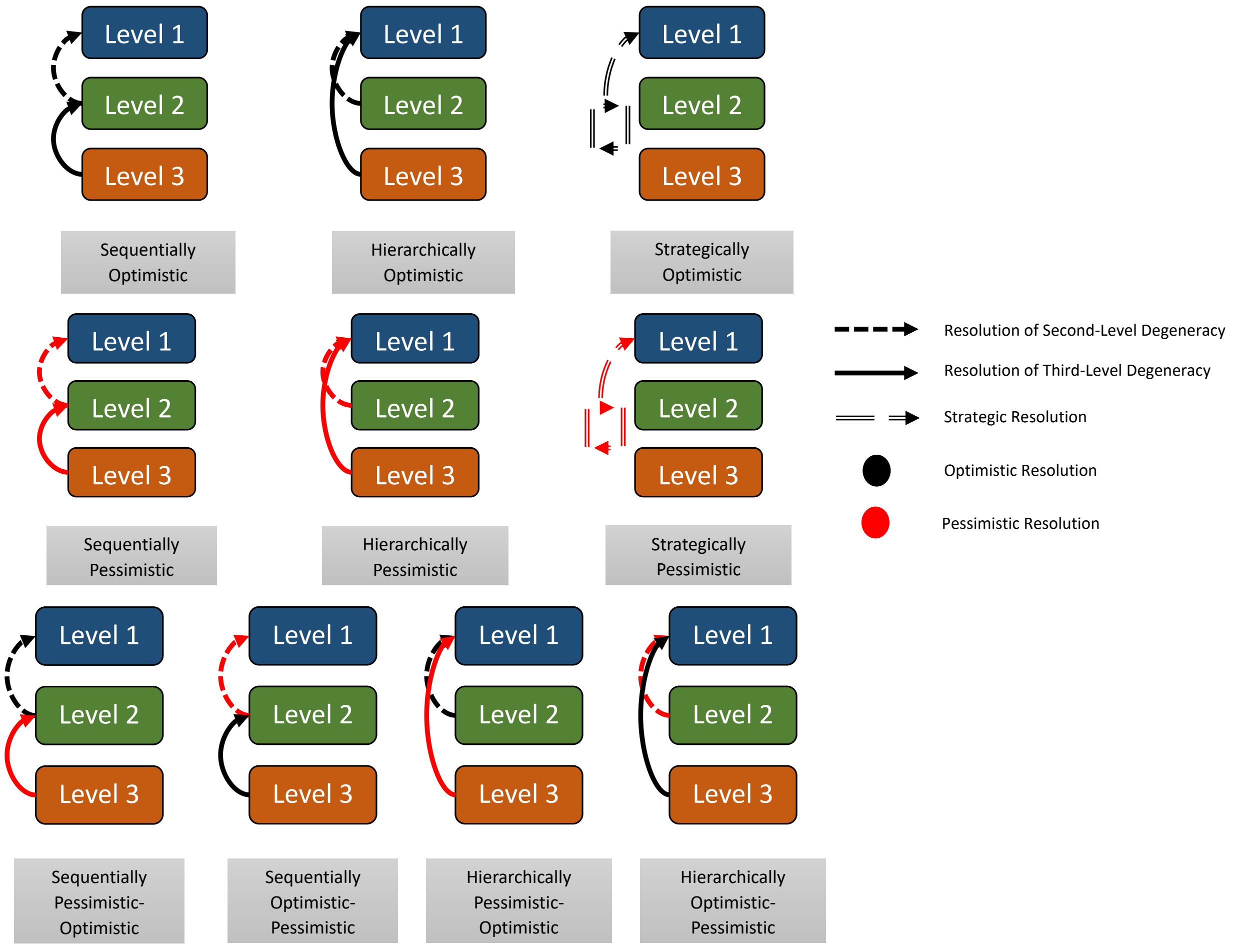

Level 1
Level 2
Level 3
Sequentially Optimistic
Hierarchically Optimistic
Strategically Optimistic
Sequentially Pessimistic
Hierarchically Pessimistic
Strategically Pessimistic
Sequentially Pessimistic-Optimistic
Sequentially Optimistic-Pessimistic
Hierarchically Pessimistic-Optimistic
Hierarchically Optimistic-Pessimistic
Resolution of Second-Level Degeneracy
Resolution of Third-Level Degeneracy
Strategic Resolution
Optimistic Resolution
Pessimistic Resolution

Figure 2.8: Degeneracy in Tri-Level Programs.

Definition 2.5.7 (Basic Rational Reaction Set of the Third-Level, $\Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})$). *Let $\Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})$ be the basic rational reaction set of the third-level for fixed $(\boldsymbol{x},\boldsymbol{y}) \in X \times Y$:*

$$\Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) = \arg\min_{z \in Z}\{f_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) : \; g_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \geq 0\} \tag{2.28}$$

or in a more compact form:

$$\Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) = \underset{\boldsymbol{z}\in\Omega_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})}{\arg\min}\{f_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\} \tag{2.29}$$

Definition 2.5.8 (Union of Second- and Third-Levels' Constraints Region, $\Omega_{\boldsymbol{y},\boldsymbol{z}}(\boldsymbol{x})$)**.** *Let $\Omega_{\boldsymbol{y},\boldsymbol{z}}(\boldsymbol{x})$ be the union of second- and third-level constraints region for fixed $\boldsymbol{x} \in X$:*

$$\Omega_{\boldsymbol{y},\boldsymbol{z}}(\boldsymbol{x}) = \{(\boldsymbol{y},\boldsymbol{z}) \in Y \times Z : \ \ g_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \geq 0,\ g_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \geq 0\} \tag{2.30}$$

Definition 2.5.9 (Basic Rational Reaction Set of the Second-Level, $\Psi_{\boldsymbol{y}}(\boldsymbol{x})$)**.** *Let $\Psi_{\boldsymbol{y}}(\boldsymbol{x})$ be the basic rational reaction set of the second-level for fixed $\boldsymbol{x} \in X$:*

$$\Psi_{\boldsymbol{y}}(\boldsymbol{x}) = `\underset{\boldsymbol{y}\in\Omega_{\boldsymbol{y},\boldsymbol{z}}(\boldsymbol{x})}{\arg\max}'\{f_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) : \boldsymbol{z} \in \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})\} \tag{2.31}$$

The single quotes for the basic rational reaction set of the second-level indicate its ill-definition since a selection approach needs to be defined in the case of degenerate solutions of the third-level. Nevertheless, it is necessary for representing the next definitions in a compact form, making the aforementioned definition well-defined.

Definition 2.5.10 (Sequentially Optimistic Reaction Sets, $\Psi^{So}_{\mathrm{II},\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})$, $\Psi^{So}_{\mathrm{I},\boldsymbol{y}}(\boldsymbol{x})$)**.** *The optimal solution to a tri-level program is said to be sequentially optimistic, if degenerate solutions in the third-level are resolved in favour-of the second-level, and multiple optima in the second-level are resolved in favour-of the first-level.*

- *The sequentially optimistic reaction set of the third-level (i.e., $\Psi^{So}_{\mathrm{II},\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})$) for* *fixed* $(\boldsymbol{x}, \boldsymbol{y}) \in X \times Y$ *can be represented as:*

$$\Psi^{So}_{\text{II},\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{z} \in \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})}{\arg\max} \{f_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\}. \tag{2.32}$$

- *The sequentially optimistic reaction set of the second-level for fixed* $\boldsymbol{x} \in X$*:*

$$\Psi^{So}_{\text{I},\boldsymbol{y}}(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \Psi_{\boldsymbol{y}}(\boldsymbol{x})}{\arg\min} \{f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) : z \in \Psi^{So}_{\text{II},\boldsymbol{z}}\}. \tag{2.33}$$

It is worth noting that the sub-script Roman number in $\Psi^{So}_{\text{II},\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y})$ indicates the level to which degeneracy is resolved for/against, and the superscript So indicates that it is a part of the sequentially optimistic approach (i.e., resolved in favour-of the second-level).

In an analogous way, we can represent the *sequentially pessimistic reactions sets* for the third- and second-levels.

- The *sequentially pessimistic* reaction set of the third-level for fixed $(\boldsymbol{x}, \boldsymbol{y}) \in X \times Y$ can be represented as:

$$\Psi^{Sp}_{\text{II},\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{z} \in \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})}{\arg\min} \{f_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\} \tag{2.34}$$

- The *sequentially pessimistic* reaction set of the second-level for fixed $\boldsymbol{x} \in X$:

$$\Psi^{Sp}_{\text{I},\boldsymbol{y}}(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \Psi_{\boldsymbol{y}}(\boldsymbol{x})}{\arg\max} \{f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) : z \in \Psi^{Sp}_{\text{II},\boldsymbol{z}}\} \tag{2.35}$$

Definition 2.5.11 (Hierarchically Optimistic Reaction Sets, $\Psi^{Ho}_{\text{I},\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}), \Psi^{Ho}_{\text{I},\boldsymbol{y}}(\boldsymbol{x})$)**.**

The optimal solution to a tri-level program is said to be hierarchically optimistic, if degenerate solutions in the third-level are resolved in favour-of the first-level, and multiple optima in the second-level are resolved in favour-of the first-level.

- *The hierarchically optimistic reaction set of the third-level for fixed* $(\boldsymbol{x}, \boldsymbol{y}) \in X \times Y$ *can be represented as:*

$$\Psi^{Ho}_{\mathrm{I},\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{z} \in \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})}{\arg\min} \{f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\}. \tag{2.36}$$

- *The hierarchically optimistic reaction set of the second-level for fixed* $\boldsymbol{x} \in X$*:*

$$\Psi^{Ho}_{\mathrm{I},\boldsymbol{y}}(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \Psi_{\boldsymbol{y}}(\boldsymbol{x})}{\arg\min} \{f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) : z \in \Psi^{Ho}_{\mathrm{I},\boldsymbol{z}}\}. \tag{2.37}$$

Similarly, the *hierarchically pessimistic reaction sets* can be defined as follows:

- The *hierarchically pessimistic* reaction set of the third-level for fixed $(\boldsymbol{x}, \boldsymbol{y}) \in X \times Y$ can be represented as:

$$\Psi^{Hp}_{\mathrm{I},\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{z} \in \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})}{\arg\max} \{f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\}. \tag{2.38}$$

- The *hierarchically pessimistic* reaction set of the second-level for fixed $\boldsymbol{x} \in X$:

$$\Psi^{Hp}_{\mathrm{I},\boldsymbol{y}}(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \Psi_{\boldsymbol{y}}(\boldsymbol{x})}{\arg\max} \{f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) : z \in \Psi^{Hp}_{\mathrm{I},\boldsymbol{z}}\}. \tag{2.39}$$

Definition 2.5.12 (Strategically Optimistic Reaction Set, $\Psi^{Stro}_{\mathrm{I},(\boldsymbol{y},\boldsymbol{z})}(\boldsymbol{x})$)**.**

The optimal solution to a TLP is said to be strategically optimistic, if degeneracy in the second- and third-levels are resolved such that the best first-level solution is obtained. The strategically optimistic reaction set of the second- and third-levels for fixed $\boldsymbol{x} \in X$ can be represented as:

$$\Psi^{Stro}_{\mathrm{I},(\boldsymbol{y},\boldsymbol{z})}(\boldsymbol{x}) = \{(\boldsymbol{y},\boldsymbol{z}) \in \Psi_{\boldsymbol{y}}(\boldsymbol{x}) \times \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) : \forall\ \tilde{\boldsymbol{y}} \in \Psi_{\boldsymbol{y}}(\boldsymbol{x}), \exists\ \ \tilde{\boldsymbol{z}} \in \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) : f_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \leq f_1(\boldsymbol{x},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{z}})\}. \quad (2.40)$$

We can represent the pessimistic counter-part definition as follows:

$$\Psi^{Strp}_{\mathrm{I},(\boldsymbol{y},\boldsymbol{z})}(\boldsymbol{x}) = \{(\boldsymbol{y},\boldsymbol{z}) \in \Psi_{\boldsymbol{y}}(\boldsymbol{x}) \times \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) : \forall\ \tilde{\boldsymbol{y}} \in \Psi_{\boldsymbol{y}}(\boldsymbol{x}), \exists\ \ \tilde{\boldsymbol{z}} \in \Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) : f_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \geq f_1(\boldsymbol{x},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{z}})\}. \quad (2.41)$$

Definition 2.5.13 (Sequentially Pessimistic Reaction Sets, $\Psi^{Spo}_{\mathrm{II},\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})$, $\Psi^{Spo}_{\mathrm{I},\boldsymbol{y}}(\boldsymbol{x})$)**.** *The optimal solution to a tri-level program is said to be sequentially pessimistic-optimistic (i.e., Spo), if degenerate solutions in the third-level are resolved against the second-level decision-maker (i.e., pessimistic), and multiple optima in the second-level are resolved in favour-of the first-level decision-maker (i.e., optimistic).*

- *The sequentially pessimistic-optimistic reaction set of the third-level for fixed $(\boldsymbol{x},\boldsymbol{y}) \in X \times Y$ can be represented as:*

$$\Psi^{Spo}_{\mathrm{II},\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) \equiv \Psi^{Sp}_{\mathrm{II},\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y}) = \underset{\boldsymbol{z}\in\Psi_{\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{y})}{\arg\min}\{f_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\}. \quad (2.42)$$

- *The sequentially pessimistic-optimistic reaction set of the second-level for fixed* $\boldsymbol{x} \in X$*:*

$$\Psi^{Spo}_{\mathrm{I},\boldsymbol{y}}(\boldsymbol{x}) = \underset{\boldsymbol{y}\in\Psi_{\boldsymbol{y}}(\boldsymbol{x})}{\arg\min}\{f_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) : z \in \Psi^{Spo}_{\mathrm{II},\boldsymbol{z}}\}. \tag{2.43}$$

Mathematical representations for the rest of the selection approaches in Figure 2.8 can be easily deducted in a similar manner to the definitions above. These definitions would help in clearing any ambiguity encountered when defining feasible regions and reaction sets of a TLP.

2.6 Algorithms and Solution Approaches

In this section we classify the algorithms and solution approaches that were used in the reviewed works into seven main categories with a special focus on articles that appeared in ORMS journals as shown in Table 2.6.

It is worth noting that some articles can be classified twice if they employed more than one approach. For instance, Fakhry et al. (2022) designed an enumeration approach, and three heuristic approaches. Additionally, we classify articles according to their reduction of the tri-level structure before implementing their respective solution algorithm in Table 2.7. First, we start with the classical and decomposition techniques as they are the most widely used to tackle TLPs. This is followed by a discussion on using enumeration, and equilibrium constraints techniques; those techniques often complement a heuristic or a special designed algorithm to provide a solution for the TLP. After which, we discuss machine learning, heuristic, and meta-heuristic methodologies.

Solution Approach	References	No. of Articles/ (Percentages)
Decomposition Techniques	Hasanzad & Rastegar (2022), Feng et al. (2021), Baggio et al. (2021), Ghorbani-Renani et al. (2020), Lazzaro & Carlyle (2019), Fanzeres et al. (2019), Fang & Zio (2019), Ramamoorthy et al. (2018), Rahdar et al. (2018), Ding et al. (2018), Sarhadi et al. (2017), Ouyang et al. (2017), Ouyang (2017), Florensa et al. (2017), Fang & Sansavini (2017), Ruiz & Conejo (2015), Yuan et al. (2014)	17 / (32.1%)
Enumeration	Wu et al. (2022), Fakhry et al. (2022), Parajuli et al. (2021) Sadati et al. (2020), Schweitzer & Medal (2019), Dey & Jenamani (2019), Sariddichainunta & Inuiguchi (2017), Parajuli et al. (2017), Han et al. (2016), Alguacil et al. (2014), Liberatore et al. (2012)	11 / (20.8%)
Equilibrium Constraints	Li (2021), Tian et al. (2021), Coniglio et al. (2021), Li et al. (2020), Aussel et al. (2020), Gu et al. (2019), Fanzeres et al. (2019), Lei et al. (2018), Ruiz & Conejo (2015), Wu et al. (2011)	10 / (18.9%)
Heuristics	Wu et al. (2022), Fakhry et al. (2022), Wu et al. (2021), Lazzaro & Carlyle (2019), Gu et al. (2019), Ke & Bookbinder (2018), Ashraf & Yuen (2017), Liu & Wang (2017), Han et al. (2016), Jin et al. (2015), Wu et al. (2011)	11 / (20.8%)
Meta-heuristics	Caçador et al. (2021), Sadati et al. (2020), Parvasi et al. (2019)	3 / (5.7%)
Machine Learning Techniques	Kaviani et al. (2018)	1 / (1.9%)

Table 2.6: Classification of Solution Approaches and Algorithms.

Classical and Decomposition Techniques

Decomposition techniques that divide the tri-level problem structure into a master-problem and a sub-problem are the most prevalent in solving tri-level programs; especially if integer decision variables exist in the most lower-level rendering the tri-level structure irreducible. These algorithms are always tailored to the special structure of the tri-level program, and resemble Benders decomposition or the column-and-constraint generation algorithms. For example, Hasanzad & Rastegar (2022) formulated a tri-level mixed-integer linear program that has binary variables in the lower-level, which prompted the use of a nested column-and-constraint generation algorithm to solve the model. The tri-level structure was decomposed into two problems: a single-level master-problem, and a two-level sub-problem transformed into

Structure	References	No. of Articles/ (Percentages)
Defender-Attacker-Defender Structure	Wu et al. (2022), Fakhry et al. (2022), Sadati et al. (2020), Parvasi et al. (2019), Ghorbani-Renani et al. (2020), Ouyang (2017), Fang & Sansavini (2017), Han et al. (2016), Jin et al. (2015), Yuan et al. (2014), Alguacil et al. (2014)	11/ (20.1%)
Reduced to Single-level Structure	Tian et al. (2021), Coniglio et al. (2021), Fanzeres et al. (2019), Lei et al. (2018)	4 / (7.5%)
Reduced to Bi-level Structure	Hasanzad & Rastegar (2022), Wu et al. (2022), Fakhry et al. (2022), Wu et al. (2021), Li (2021), Caçador et al. (2021), Li et al. (2020), Aussel et al. (2020), Schweitzer & Medal (2019), Fang & Zio (2019), Dey & Jenamani (2019), Ramamoorthy et al. (2018), Ouyang et al. (2017), Florensa et al. (2017), Han et al. (2016), Alguacil et al. (2014), Liberatore et al. (2012), Wu et al. (2011)	18 / (33.9%)
Maintained Tri-level Structure	Feng et al. (2021), Baggio et al. (2021), Sadati et al. (2020), Parvasi et al. (2019), Ghorbani-Renani et al. (2020), Lazzaro & Carlyle (2019), Gu et al. (2019), Rahdar et al. (2018), Ke & Bookbinder (2018), Ding et al. (2018), Ashraf & Yuen (2017), Parajuli et al. (2017), Liu & Wang (2017), Fang & Zio (2019), Han et al. (2016), Jin et al. (2015), Yuan et al. (2014)	17/ (32.1%)

Table 2.7: Classification According to Tri-level Structure.

a single-level using the duality theory approach. Feng et al. (2021) implemented a decomposition approach that reduces the tri-level structure into a bi-level main problem ($\min - \min$) and a bi-level sub-problem ($\max - \min$). The main problem was then relaxed into a single-level (min) problem, since the two levels shared the same objective function. Baggio et al. (2021) designed a row-and-column generation algorithm to solve the multi-level critical node problem. Ghorbani-Renani et al. (2020) implemented a decomposition approach that extends the standard covering decomposition approach initially proposed by Israeli & Wood (2002). Lazzaro & Carlyle (2019) implemented a procedure akin to Benders decomposition, except they introduced elimination constraints to prevent cycling solutions. A column-and-constraint generation algorithm was used in Ding et al. (2019), after converting the most two lower-levels into a single-level by taking the dual of the most lower-level problem. In Davarikia & Barati (2018), the tri-level problem was decomposed into a master-problem and sub-problem. The master-problem included the first- and third-levels; this is due to the special structure of the problem, as both levels were minimizing the same objective function. For the sub-problem, the authors used duality theory to cast the two lower levels into a single level; this is due to the linearity of the third-level. A column and constraint generation algorithm was then used to iterate between the master problem and sub-problem, till the solution converges. Primal cutting plane methods and Benders decomposition were both used in Ebrahimi & Amjady (2019). The rest of the references mentioned in Table 2.6 have proposed similar decomposition algorithms as the aforementioned research articles, with slight differences depending on the application under study.

Enumeration Techniques

The majority of the defender-attacker-defender models, grouped in Table 2.7, rely on the implicit enumeration idea proposed by Church & Scaparra (2007), which is the core of the red cluster in our co-citation analysis in Figure 2.6 (e.g., Parajuli et al. (2021), Schweitzer & Medal (2019), Dey & Jenamani (2019), Parajuli et al. (2017), Sarhadi et al. (2017), Alguacil et al. (2014), Liberatore et al. (2012)), if an enumeration methodology is used in the search for an exact solution. The enumeration algorithm relies on an observation stating that in order to prevent the worst-case scenario, at least one of the assets causing that scenario must be protected/defended. Wu et al. (2022) used a greedy algorithm for both the attacker and defender to enumerate possible defence and attack strategies. The quality of the solution is then determined by a variable neighbourhood search. Fakhry et al. (2022) designed a modified non-binary tree search with a warm-starting solution technique that significantly enhanced the run-time compared to the binary tree search proposed by

Church & Scaparra (2007). Sadati et al. (2020) proposed an exhaustive enumeration scheme where some patterns are excluded and only distinctive combinations are to be considered depending on the attack and defense budget. Sariddichainunta & Inuiguchi (2017) proposed a global optimality test based on an inner approximation method, and compared its computational efficiency to other test methods based on vertex enumeration. They show that an optimal solution exists at a vertex of a feasible region upon solving a special three-level programming problem. Han et al. (2016) enumerated all possible strategies for the defender and the attacker. However, they were constrained with small instances due to the large number of combinations.

Equilibrium Constraints Technique

Equilibrium constraints is a terminology used extensively in transportation (e.g., Tian et al. (2021)) and is synonymous to complementarity conditions. Simply put, these are conditions that guarantee that the party (e.g., passenger) under consideration will have no incentive deviating from the solution if it satisfies the complementarity constraint, hence equilibrium is achieved. Tian et al. (2021) reformulated the tri-level program into a mathematical program with equilibrium constraints. Two approaches were proposed to solve the model: 1) transforming the model into a single-level mixed-integer program, and 2) a surrogate optimization approach was suggested to tackle large problem instances. Li (2021) decomposed the tri-level structure into bi-level by replacing the most third-level with its complementarity constraints and used a game-theoretic framework by introducing constraints ensuring individual, group, and subgroup rationality are satisfied. Furthermore, a synchronous iterative method was used to solve the tri-level problem in Gu et al. (2019), where the second and third levels are equilibrium problems with equilibrium constraints. The second and third levels represent equilibrium between private firms (Cournot-Nash equilibrium) and users (user-equilibrium traffic problem), respectively. The authors used a synchronous iterative method that loops on toll-capacity adjustment for private firms and user equilibrium traffic assignment in a synchronous manner, in the sense that corresponding iterations are executed as a group. Moreover, the updating procedure is done using a successive average method. In Nemati et al. (2019), each transmission line is regarded as a virtual attacker. A co-operative game approach is formulated by including the KKT conditions for each virtual attacker that maximizes the network damages. The pareto equilibria conditions are then formulated and added to the equilibrium constraints. Thus, converting the bi-level problem into a single-level mixed-integer linear problem. It is worth mentioning that the authors were inspired by the exact solution method developed by Huppmann & Siddiqui (2018) for solving binary equilibrium problems. Aussel et al. (2020) used the classical method to decompose the tri-level structure into a bi-level problem by replacing the followers' problems by their KKT conditions in the second-level yielding a bi-level mathematical program with complementarity constraints. Lastly, Wu et al. (2011) formulated a tri-level leader-follower game with certain assumptions into a mixed-integer program with equilibrium constraints. They applied it to the build-operate-transfer projects in transportation, The authors showed that the optimal solution can be determined from the problem structure leading to an efficient heuristic algorithm. Other research articles that show under “Equilibrium Constraints” in Table 2.6 have implemented similar ideas to the above mentioned research work.

Heuristics Techniques

Heuristic techniques are widely used in solving tri-level programs because of their ability to provide near-optimal/optimal solutions with minimal computational effort. Wu et al. (2022) implemented a heuristic procedure that is similar in spirit to the active-set algorithm previously proposed by Zhang et al. (2009). The heuristic operates in a greedy manner by evaluating the marginal benefit of changing/mutating a candidate solution, and then solves a knapsack problem to update the solution. The candidate solution with the most decrease in system cost will be selected. Furthermore, Fakhry et al. (2022) proposed three different heuristic approaches that offer a trade-off between solution quality and computational time.The proposed heuristics are used to rank the binary variables in a pre-planning stage providing fast optimal/near-optimal solutions with much less computational effort. In particular, the *LPRank* heuristic solves a series of linear programs instead of having to solve a bi-level mixed-binary program. In contrast, the *HybridRank* heuristic triggers both linear and mixed-binary linear program solvers, depending on the change of objective function values through iterations, while the *MILPRank* heuristic only invokes a mixed-binary program solver, thereby yielding a better quality and taking relatively more time per instance. Additionally, Wu et al. (2021) implemented a variable neighbourhood search procedure that initializes a solution using a greedy algorithm, explores the neighbourhood searching for a better solution, and then repeats the cycle till all neighbourhoods are searched. A similar approach, was implemented in defending software-defined networks in (Ashraf & Yuen 2017), urban rail transit networks in (Jin et al. 2015), and build-operate-transfer project schemes in (Wu et al. 2011).

Lazzaro & Carlyle (2019) investigated heuristic and parametric programming approaches to identify the set of nested defences; which is a monotonic sequence of sets, where the set of defences for a particular budget scenario contains the set of defences for all smaller budget scenarios. Gu et al. (2019) proposed a heuristic algorithm to solve a tri-level mathematical program with equilibrium constraints in the second- (i.e., Cournot- Nash Equilibrium) and third- levels (traffic user equilibrium). Diagonalization and synchronous iterative methods were used to solve the programs with equilibrium constraints separately and sequentially while holding the decision variables of other players fixed in-turn, till the sequence converges. A similar idea, but in a different context, was proposed by Ke & Bookbinder (2018), where a heuristic algorithm was designed to start from the initial market equilibrium and then update each party/member's with its preferred decision till convergence. Liu & Wang (2017) formulated a tri-level model, that was first treated as black-box optimization, and then solved by an efficient surface-response-approximation model-based algorithm.

Meta-Heuristics Techniques

Meta-heuristic techniques have been used in solving TLPs such as genetic algorithms in Caçador et al. (2021) which allowed the reduction of a three-level optimization problem to a two-level problem. In general, evolutionary algorithms allow researchers to solver more complex and combinatorial problems with non-linear or non-convex objective functions. Some authors have opted for hybrid method. (Sadati et al. 2020) designed a meta-heuristic that is an amalgamation of a variable neighbourhood descent and tabu search techniques to solve the lower-level problem. The algorithm requires an initial solution which is generated using a greedy heuristic. Parvasi et al. (2019) proposed a combination of a genetic algorithm, simulated annealing. The genetic algorithm was used to form a population of candidate solutions and the simulated annealing was used as a neighbourhood search algorithm. In each iteration, the fitness function of the proposed chromosome was evaluated by substituting in the first level's objective function.

Machine Learning Techniques

Machine learning techniques have not been applied extensively, as researchers focused mainly on efficient decomposition techniques, heuristics, and combining both with enumeration methodologies. A disadvantage of machine learning algorithms that they require data for training which might need further processing. Recently, Kaviani et al. (2018) explored four different machine learning algorithms, which are logistic regression, support vector machine, random forest and lastly artificial neural networks. Based on the preliminary results, artificial neural networks had the highest accuracy among the different implemented algorithms. Principle component analysis was also used to reduce the dimensionality of the data that is fed to each machine learning algorithm before training. The overall technique provides acceptable solutions, and not necessarily optimal. It also provides a new lens to examine the interdiction problem.

2.7 Applications

Applications on tri-level programs span a wide range of spectrum due to the need of incorporating decentralized decision-making processes in different areas. In this discussion, we categorize research articles that have appeared in ORMS journals in Table 2.8, as we want to induce more research contributions in this area.

Application	References	No. of Articles/ (Percentages)
Defending Electrical Systems	Wu et al. (2022), Hasanzad & Rastegar (2022), Fakhry et al. (2022), Zhu et al. (2021) , Ding et al. (2018), Ouyang et al. (2017), Fang & Sansavini (2017), Yuan et al. (2014), Alguacil et al. (2014)	9/ (17%)
Defense Multi-disciplinary	Wu et al. (2021), Parajuli et al. (2021), Baggio et al. (2021), Sadati et al. (2020), Parvasi et al. (2019), Ghorbani-Renani et al. (2020), Schweitzer & Medal (2019), Lazzaro & Carlyle (2019) Fang & Zio (2019), Dey & Jenamani (2019), Ramamoorthy et al. (2018), Lei et al. (2018), Ashraf & Yuen (2017), Parajuli et al. (2017), Ouyang (2017), Han et al. (2016) Liberatore et al. (2012), Brown et al. (2006)	18/ (34%)
Transportation	Tian et al. (2021), Feng et al. (2021), Coniglio et al. (2021), Sadati et al. (2020), Gu et al. (2019), Ke & Bookbinder (2018), Sarhadi et al. (2017), Liu & Wang (2017), Jin et al. (2015), Wu et al. (2011)	10 / (18.9%)
Electric Grid Applications	Li (2021), Aussel et al. (2020), Ruiz & Conejo (2015)	3/ (5.7%)
Theory Development	Gabriel et al. (2021), Li et al. (2020), Avraamidou & Pistikopoulos (2019), Li & Wan (2018), Inuiguchi & Sariddichainunta (2016), White (1997), Sariddichainunta & Inuiguchi (2017)	7/ (13.2%)
Miscellaneous	Wang et al. (2022), Caçador et al. (2021), Fanzeres et al. (2019), Rahdar et al. (2018), Florensa et al. (2017)	5/ (9.4%)

Table 2.8: Classification According to Applications.

Papers that studied defending critical infrastructure applications were a major driving force for researchers' interest in TLPs. Consequently, this in-turn sparked theory and algorithm design to tackle the inherent difficulty that arises from even the simplest tri-level programs. Defending electrical networks has attracted many researcher; this is mainly due to the interdependence of other critical infrastructure on electricity such as transportation and health care systems, which might cause a chain effect of cascading failures. Many researcher have studied defending electrical networks against physical, and cyber attacks, and natural disasters (Wu et al. 2022, Hasanzad & Rastegar 2022, Fakhry et al. 2022, Zhu et al. 2021, Fang & Zio

2019, Ding et al. 2018, Ouyang et al. 2017, Alguacil et al. 2014, Yuan et al. 2014). Other critical infrastructure networks have also been investigated such as urban water distribution networks (Wu et al. 2021), facilities and supply networks (Parajuli et al. 2021, Dey & Jenamani 2019, Ramamoorthy et al. 2018, Parajuli et al. 2017, Liberatore et al. 2012), vehicle routing with depot protection (Sadati et al. 2020), service facility protection with a time horizon (Parvasi et al. 2019), a system of interdependent networks (Ghorbani-Renani et al. 2020, Fang & Zio 2019, Ouyang 2017), wireless local area network transmitters (Schweitzer & Medal 2019), general critical node problems (Baggio et al. 2021, Lazzaro & Carlyle 2019, Brown et al. 2006), software-defined networks (Ashraf & Yuen 2017), and air defence (Han et al. 2016). Furthermore, electric grid related applications such as transmission expansion Ruiz & Conejo (2015), demand side management, and coordination between smart distribution networks and multi-micro-grids(Li 2021) have attracted the attention of many researchers. For instance, Ebrahimi & Amjady (2019) discuss micro grid scheduling subject to uncertainties such as wind and solar generation, load demand and electricity price. The study suggests an adaptive robust micro grid scheduling model with recourse as a two-stage tri-level problem. KKT conditions of the third-level were added to the second-level. The authors then apply a primal cutting plane algorithm to solve the resulting bi-level problem. The authors defined two problems: *AdptRob*, and *ConvAc*, where the latter is solved by a primal cutting plane algorithm to determine the worst uncertainty case parameters. The former is then solved using Benders decomposition to obtain Micro Grid scheduling. Nemati et al. (2019) discuss a transmission expansion problem under physical intentional attacks. The network planner determines the optimal transmission expansion plan that minimizes the system load shed or operating costs in the first-level. In the second-level, the attacker tries to maximize the damages done to the network, while the network operator minimizes the effect done by maintaining the operational constraints of the electric network. Since the third-level is a linear programme, the second- and third-levels are converted into a single-level resulting in an overall bi-level optimization problem. A co-operative game-theoretic approach between multiple virtual attackers is then used to cast the bi-level structure into a single-level. This is done by enforcing Nash and Pareto equilibria conditions in the single-level model. Ding et al. (2019) studied long-term transmission system hardening, in which risk assessment is taken into consideration in the second-level objective function. Duality theory was applied to the third-level to convert the tri-level problem into a bi-level problem. The resulting non-linear objective function in the inner model was linearized using piecewise functions and a logarithmic transformation. The authors used a column-and-constraint generation algorithm to solve the linearized bi-level problem.

Koltsaklis & Dagoumas (2018) reviews generation expansion planning from different points of view, and its relation to tri-level programs comes from the fact that generation is coupled with transmission expansion. Misaghian et al. (2018) presented a novel framework for optimal scheduling of industrial micro-grids as a tri-level problem. In the first-level, the industrial micro-grid operator maximizes its revenue through minimizing the cost while taking into account the stochastic nature of the distributed energy resources. The main output from the first-level is the bids, which can be either to sell/buy electricity to/from the market. These bids are then fitted to a quadratic function to be used in the second-level, which are linearized using piecewise linear functions. The grid operator has an objective of minimizing the total expected cost of grid operation, while maintaining its operational constraints, and analysing (i.e., deciding to accept or reject) the received bids from the industrial micro-grid operator. In the third-level, based on the accepted bids, the micro-grid operator optimizes the operation cost and maintain operational constraints.

Transportation applications that involve decentralized decision-making have been discussed in the literature (Tian et al. 2021, Feng et al. 2021, Coniglio et al. 2021, Sadati et al. 2020, Gu et al. 2019). For instance, Gu et al. (2019) studied the road competition problem, where the first-level represents the government's perspective to maximize the social welfare of the transportation system. The second-level represents an oligopolistic competition problem, where each private firm seeks to compete in order to maximize its profits.The third-level models the road users problem as they seek to pay the least amount for travelling from the origin to the destination and maximize their benefits. Sarhadi et al. (2017) applied the defender-attacker-defender model to improve the resilience of railroad intermodal networks.

Researchers have directed their attention towards theory development in an effort to cope with the need for efficient algorithms to solve tri-level mathematical programs. For example, Li et al. (2020) studied the optimality conditions for a class of tri-level optimization problems, of which all levels are non-linear. Avraamidou & Pistikopoulos (2019) presented an algorithm for the global solution of tri-level mixed-integer linear problems based on multi-parametric theory. The need for more efficient algorithms is growing, as practical applications that are in need of decentralized decision-making continue to appear. This will in-turn call for algorithms to solve non-linear mixed-integer programs, an area that still needs more attention when compared to linear mixed-integer programs.

2.8 Future Research Directions

In this section we report on future research directions based on what we found from the literature review and our own work in this area. Based on our literature review we grouped future research directions into four categories as shown in Table 2.9.

Category	References	No. of Articles/ (Percentages)
Modelling and Applications	Wu et al. (2022), Wang et al. (2022), Hasanzad & Rastegar (2022), Zhu et al. (2021) Wu et al. (2021), Tian et al. (2021), Parajuli et al. (2021), Feng et al. (2021) Coniglio et al. (2021), Caçador et al. (2021), Baggio et al. (2021), Sadati et al. (2020) Parvasi et al. (2019), Ghorbani-Renani et al. (2020), Aussel et al. (2020), Schweitzer & Medal (2019) Lazzaro & Carlyle (2019), Gu et al. (2019), Fanzeres et al. (2019), Dey & Jenamani (2019) Ramamoorthy et al. (2018), Rahdar et al. (2018), Lei et al. (2018), Ashraf & Yuen (2017) Sarhadi et al. (2017), Ouyang et al. (2017), Ouyang (2017), Florensa et al. (2017) Han et al. (2016), Jin et al. (2015), Yuan et al. (2014), Liberatore et al. (2012) Wu et al. (2011)	33/ (62.26%)
Algorithm Development	Wu et al. (2022), Wang et al. (2022), Fakhry et al. (2022), Wu et al. (2021) Coniglio et al. (2021), Baggio et al. (2021), Sadati et al. (2020), Parvasi et al. (2019), Inuiguchi & Sariddichainunta (2016) Ghorbani-Renani et al. (2020), Aussel et al. (2020), Lazzaro & Carlyle (2019), Gu et al. (2019) Dey & Jenamani (2019), Ramamoorthy et al. (2018), Lei et al. (2018), Sadeghi et al. (2017) Parajuli et al. (2017), Ouyang (2017), Florensa et al. (2017), Han et al. (2016) Alguacil et al. (2014)	22/ (41.5%)
Optimization Under Uncertainty	Zhu et al. (2021), Tian et al. (2021), Parvasi et al. (2019), Avraamidou & Pistikopoulos (2019) Schweitzer & Medal (2019), Lazzaro & Carlyle (2019), Fanzeres et al. (2019), Ramamoorthy et al. (2018) Ramamoorthy et al. (2018), Rahdar et al. (2018), Ke & Bookbinder (2018), Parajuli et al. (2017) Ouyang et al. (2017), Ouyang (2017), Florensa et al. (2017), Han et al. (2016) Liberatore et al. (2012), Wu et al. (2011)	18/ (34%)
Theory Development	Coniglio et al. (2021), Li et al. (2020), Avraamidou & Pistikopoulos (2020), Gu et al. (2019) Fanzeres et al. (2019), Avraamidou & Pistikopoulos (2019), Li & Wan (2018), Sariddichainunta & Inuiguchi (2017)	8/(15.1%)

Table 2.9: Categories for Future Research.

Below we provide a detailed list of future research directions grouped by the categories in Table 2.9.

1. **Modelling and applications**: TLP is a discipline that largely grew out of specific applications. It is therefore not surprising that the majority of works

have called for more research on modelling and applications. We grouped these into four classes:

(a) *Competition*: Often, a modeller will resort to using TLP to capture conflicts between decision makers. In many instances there will also be competition between the decision makers. In such cases it is desired to reflect such interactions in the TLP. One challenge is how to reflect the competition in a TLP. Some effiorts have already been made in this area, but they are application-specific. For example, Florensa et al. (2017) modelled capacity competition between two producers in a market. They present a three-level Stackelberg game that resulted in a TLP with integer variables controlled by the two upper levels that represent the firms. Gu et al. (2019) studied road competition where the government leads in building toll roads (level one), with a social welfare objective, and competes with private developers (level two) to serve road users (level three). The resulting TLP has a continuous optimization problem in level one, a Cournot–Nash Equilibrium in level two, and a Wardrop traffic equilibrium in level three. Wang et al. (2022) suggested modeling extensions for its tri-level model (Government-Port-Ship) by including competitions between ports in the re-fueling markets (i.e., second-level), as well as the traditional marine fuel bunkering market. At the third-level, the ships operators are assumed to work independently, but in reality they will compete for cargoes in the transportation market. Hence, considering the competition between ships at the third-level would be an interesting extension. An interesting avenue of future research is to develop general frameworks for incorporating competitions in TLP.

(b) *Complex Systems*: Most applications of TLP focus on one sector or one system. Given the interdependence nature of systems, there is a need to incorporate such effects in TLPs. Examples include modelling failure propagation and their cascading effects (Wu et al. 2022), power and gas lines (Hasanzad & Rastegar 2022), different modes of transportation (Feng et al. 2021), and infrastructure interdependence (Ouyang 2017). Liberatore et al. (2012) suggested extending the model to capture complex features arising from the necessities of agencies and organizations that work in humanitarian logistics, probability of interdiction or failure, protection strategies that only mitigates the effects of disruption, and does not completely prevent it. Ashraf & Yuen (2017) suggested exploring hybrid strategies comprising of backups and recovery solutions to protect software-defined networks.

(c) *Dynamic Systems*: TLPs have generally grown as a generalization of BLPs that are in turn a generalization of Stackelberg games. While the latter capture the sequential nature of decisions, such as a leader and follower, they do not readily handle dynamic decision making, such as when the players are involved in bargaining. Thus, there is a need to develop models that can take into account situations where the decision makers iteratively resolve their conflicts (Aussel et al. 2020). Another extension involves adding the time dimension in TLPs, such as to represent the timing of the effect of an attack propagation in a network (Baggio et al. 2021).

(d) *Incomplete Information*: Most of the extant literature on TLPs has assumed all information about the three levels is known and shared between the decision makers, i.e., perfect and complete information. In many realistic situations, some of the information may not be available or is strategically hidden by one decision-maker. For example, Zhu et al. (2021) suggested incorporating the case of strategically misleading the adversary as an area of future interest. Wu et al. (2021) suggested modelling irrational attackers and incomplete information situations. Parajuli et al. (2021) called for modelling partial capacities loss and partial protection. Others called for focusing on modeling asymmetric information (Ouyang 2017, Lei et al. 2018).

2. **Algorithm development**: There are several streams of research on algorithms for TLPs. We grouped them into four categories:

(a) *Decision diagrams*: Fakhry et al. (2022) suggested the use of decision diagrams, which are based on recursive modelling, similar to that used in deterministic dynamic programming. This allows the formulation of a wide range of problems, in which linearity or convexity is no longer an issue (Hooker 2013). Applying decision diagrams to solve TLPs could be useful, even if it is applied on a single-level where convexity is a hurdle.

(b) *Decomposition*: As the need for more efficient solution approaches rises with the complexity of the models (Florensa et al. 2017), several researchers called for efficient and exact decomposition algorithms that are capable of handling large-sized problem instances (Wu et al. 2022, Fakhry et al. 2022, Wu et al. 2021, Ghorbani-Renani et al. 2020, Parvasi et al. 2019, Dey & Jenamani 2019), For instance, Ghorbani-Renani et al. (2020) suggested enhancing decomposition algorithm through adding valid inequalities. Fakhry et al. (2022) suggested developing generalized decomposition algorithms that are independent of electrical transmission networks' parameters in the context of DAD models.

(c) *Meta-heuristics*: Due to the computational complexity of TLPs, researchers have called for using meta-heuristics such as genetic algorithms (Wu et al. 2022, 2021, Sadati et al. 2020, Sarhadi et al. 2017, Parajuli et al. 2017). For instance, Sarhadi et al. (2017) and Parajuli et al. (2017) suggested developing meta-heuristic based tree search as warm-start solution technique instead of using implicit or explicit enumeration techniques.

(d) *Machine learning*: With the availability of more test cases, such as the ones proposed in Chapter 4, it is possible to generate enough solution data analytics, such as intermediate solution search data, to aid in performance machine learning, a field that is often called *learning to optimize*. Developing such methodologies can aid in solving larger classes of TLPs.

3. **Optimization under uncertainty**: Many environments where TLPs have been applied have inherent uncertainty in their parameters. It is thus not surprising that this is a promising area for future research from both the modelling

ad algorithmic development perspectives.

(a) *Modelling*: The uncertainty can stem from different sources such as demand (Wu et al. 2011), capacity (Ouyang 2017) or supply (Rahdar et al. 2018). In the event of availability of data, Fanzeres et al. (2019) suggested the development of data-driven methodologies to construct uncertainty sets directly from data. Schweitzer & Medal (2019) suggested stochastic programming interpretation of the transmitter jamming attacks in addition to extension of the attacker capabilities in the TLP. Lazzaro & Carlyle (2019) proposed modelling different defence strategies associated with different costs, including uncertainty in the attack and defence budgets, and defining a measurement index between optimal and nested defence strategies. Zhu et al. (2021) and Ouyang et al. (2017) suggested modelling uncertainties in the system (e.g., restoration time, and the amount of attack resources). Tian et al. (2021) suggested modelling uncertainties in the transit service operation problem, such as transit demand and time. Parvasi et al. (2019) suggested applying demand uncertainty to the model service facility protection, and modelling different degrees of protection to defend facilities with different costs. Ke & Bookbinder (2018) suggest incorporating demand uncertainty in the coordination of discount schemes for multiple supply chain members. Florensa et al. (2017) suggested extending their model to include stochastic parameters like demand forecasts and costs For the capacity expansion model. Inuiguchi & Sariddichainunta (2016) studied bi-level linear optimization problem with ambiguous follower's objective function coefficient vector, which can be reformulated as a TLP. For future research, they suggest modeling the follower's coefficient vector by a fuzzy set.

(b) *Algorithms*: Several ideas have been suggested for developing algorithms for TLPs. For example, Han et al. (2016) suggested improving the proposed heuristics using a neighbourhood search, applying a randomized emplacement strategy. Avraamidou & Pistikopoulos (2019) proposed using multi-parametric optimization for TLPs under uncertainty and quadratic mixed-integer adaptive robust optimization problems.

4. **Theory development**: With the multitude of applications of TLPs comes the urge to develop TLP-specific solution methodologies, rather than repackaging of BLP or MILP approaches. We see two main lines of research in this area:

    (a) *Multi-level Programming*: Li et al. (2020) studied optimality conditions for a specific class of tri-level programs in which all levels are non-linear. They suggested obtaining sufficient optimality conditions via auxiliary bi-level optimization problem for the same class of TLPs. Lei et al. (2018) suggested working on discovering special network structures where polynomial-time exact or approximate algorithms could be developed,

    (b) *Type of Equilibrium*: Coniglio et al. (2021) suggested investigating under which conditions the game that is played in the second level always admits a generalized Nash equilibrium. It would also be of interest to assess the impact on the whole tri-level problem of adopting, in the generalized Nash equilibrium problem, different equilibrium-selection strategies besides the optimistic one where a welfare-maximizing equilibrium is selected. Sariddichainunta & Inuiguchi (2017) suggested studying the equilibrium structure in multi-player in the lower-level problem of a TLP.

(c) *Structure of Tri-Level Program*: Most of the applications discussed in the literature deal with a linear tri-level program structure with a convex linear problem in the third-level. Additionally, having a mixed-integer problem in the third-level renders the tri-level problem not convertable to a bi-level programme. Researchers opted for using decomposition-based approaches for solving these problems, e.g., (Wu & Conejo 2017). These approaches are know to require fine tuning depending on the application under study. One could argue for a unified general framework for solving mixed-integer linear tri-level problems with a non-convex third-level problem. Furthermore, the non-linearity in the constraints (if exists) were dealt with using standard linearization techniques (Alguacil et al. 2014, Fakhry et al. 2022) or using meta-heuristic techniques. Moreover, using piecewise functions to approximate non-linear functions (Geißler et al. 2012) is a known approach for approximating non-linear functions. Nevertheless, solving tri-level optimally with inherent non-linear objective functions and/or constraints that cannot be linearized remains an open problem.

2.9 Conclusion

This article provides an essential and unifying background for multi-level programming with a focus on tri-level programmes. A bibliometric analysis has been conducted by searching for possible keywords to pull the relevant literature from the Web of Science core collection. The analysis of the results indicates the number of articles on TLPs is increasing at its fastest pace in the past few years soliciting the need for a review article. Furthermore, the study identified some of the most influential articles that defined the core knowledge for multi-level programming, through a meta-analysis extracting useful knowledge from the data, and depicting it through intuitive visualizations. In an effort to clear common misconceptions, we provide definitions to structure the interactions between decision-levels in TLPs, and resolve degeneracy. Additionally, we have provided a systemic review on articles that have appeared in the operations research and management science journals.

Bibliography

Alguacil, N., Delgadillo, A. & Arroyo, J. M. (2014), 'A trilevel programming approach for electric grid defense planning', *Computers & Operations Research* **41**, 282–290.

Allaire, J. (2012), 'Rstudio: integrated development environment for r', *Boston, MA* **770**(394), 165–171.

Alvarez, R. E. (2004), Interdicting electrical power grids, PhD thesis, Monterey, California. Naval Postgraduate School.

Aria, M. & Cuccurullo, C. (2017), ‘bibliometrix: An r-tool for comprehensive science mapping analysis’, *Journal of informetrics* **11**(4), 959–975.

Ashraf, U. & Yuen, C. (2017), ‘Capacity-aware topology resilience in software-defined networks’, *IEEE Systems Journal* **12**(4), 3737–3746.

Aussel, D., Brotcorne, L., Lepaul, S. & von Niederhäusern, L. (2020), ‘A trilevel model for best response in energy demand-side management’, *European Journal of Operational Research* **281**(2), 299–315.

Aussel, D. & Svensson, A. (2019), ‘Is pessimistic bilevel programming a special case of a mathematical program with complementarity constraints?’, *Journal of Optimization Theory and Applications* **181**(2), 504–520.

Avraamidou, S. & Pistikopoulos, E. N. (2019), ‘Multi-parametric global optimization approach for tri-level mixed-integer linear optimization problems’, *Journal of Global Optimization* **74**(3), 443–465.

Avraamidou, S. & Pistikopoulos, E. N. (2020), ‘Adjustable robust optimization through multi-parametric programming’, *Optimization Letters* **14**(4), 873–887.

Baggio, A., Carvalho, M., Lodi, A. & Tramontani, A. (2021), ‘Multilevel approaches for the critical node problem’, *Operations Research* **69**(2), 486–508.

Bakker, H., Dunke, F. & Nickel, S. (2020), ‘A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice’, *Omega* **96**, 102080.

Balas, E. (1971), 'Intersection cuts—a new type of cutting planes for integer programming', *Operations Research* **19**(1), 19–39.

Bard, J. F. (1984), 'Optimality conditions for the bilevel programming problem', *Naval research logistics quarterly* **31**(1), 13–26.

Bard, J. F. (1991), 'Some properties of the bilevel programming problem', *Journal of optimization theory and applications* **68**(2), 371–378.

Bard, J. F. (2013), *Practical bilevel optimization: algorithms and applications*, Vol. 30, Springer Science & Business Media.

Bard, J. F. & Moore, J. T. (1990), 'A branch and bound algorithm for the bilevel programming problem', *SIAM Journal on Scientific and Statistical Computing* **11**(2), 281–292.

Ben-Ayed, O. (1993), 'Bilevel linear programming', *Computers & operations research* **20**(5), 485–501.

Ben-Ayed, O. & Blair, C. E. (1990), 'Computational difficulties of bilevel linear programming', *Operations Research* **38**(3), 556–560.

Ben-Daya, M., Hassini, E. & Bahroun, Z. (2019), 'Internet of things and supply chain management: a literature review', *International Journal of Production Research* **57**(15-16), 4719–4742.

Berry, C. A., Hobbs, B. F., Meroney, W. A., O'Neill, R. P. & Stewart Jr, W. R. (1999), 'Analyzing strategic bidding behavior in transmission networks', *Utilities Policy* **8**(3), 139–158.

Bialas, W. F. & Karwan, M. H. (1984), 'Two-level linear programming', *Management science* **30**(8), 1004–1020.

Birge, J. R. & Louveaux, F. (2011), *Introduction to stochastic programming*, Springer Science & Business Media.

Bracken, J. & McGill, J. T. (1973), 'Mathematical programs with optimization problems in the constraints', *Operations Research* **21**(1), 37–44.

Breton, M., Alj, A. & Haurie, A. (1988), 'Sequential stackelberg equilibria in two-person games', *Journal of Optimization Theory and Applications* **59**(1), 71–97.

Brown, G., Carlyle, M., Salmerón, J. & Wood, K. (2006), 'Defending critical infrastructure', *Interfaces* **36**(6), 530–544.

Caçador, S., Dias, J. M. & Godinho, P. (2021), 'Portfolio selection under uncertainty: a new methodology for computing relative-robust solutions', *International Transactions in Operational Research* **28**(3), 1296–1329.

Cao, D. & Chen, M. (2006), 'Capacitated plant selection in a decentralized manufacturing environment: a bilevel optimization approach', *European journal of operational research* **169**(1), 97–110.

Cappanera, P. & Scaparra, M. P. (2011), 'Optimal allocation of protective resources in shortest-path networks', *Transportation Science* **45**(1), 64–80.

Carrión, M., Arroyo, J. M. & Conejo, A. J. (2009), 'A bilevel stochastic programming approach for retailer futures market trading', *IEEE Transactions on Power Systems* **24**(3), 1446–1456.

Chavarro, D., Ràfols, I. & Tang, P. (2018), 'To what extent is inclusion in the web of science an indicator of journal 'quality'?', *Research evaluation* **27**(2), 106–118.

Chen, B., Wang, J., Wang, L., He, Y. & Wang, Z. (2014), 'Robust optimization for transmission expansion planning: Minimax cost vs. minimax regret', *IEEE Transactions on Power Systems* **29**(6), 3069–3077.

Church, R. L. & Scaparra, M. P. (2007), 'Protecting critical assets: the r-interdiction median problem with fortification', *Geographical Analysis* **39**(2), 129–146.

Cobos, N. G., Arroyo, J. M., Alguacil, N. & Street, A. (2018), 'Network-constrained unit commitment under significant wind penetration: A multistage robust approach with non-fixed recourse', *Applied energy* **232**, 489–503.

Cobos, N. G., Arroyo, J. M., Alguacil, N. & Wang, J. (2018), 'Robust energy and reserve scheduling considering bulk energy storage units and wind uncertainty', *IEEE Transactions on Power Systems* **33**(5), 5206–5216.

Cobos, N. G., Arroyo, J. M. & Street, A. (2016), 'Least-cost reserve offer deliverability in day-ahead generation scheduling under wind uncertainty and generation and network outages', *IEEE Transactions on Smart Grid* **9**(4), 3430–3442.

Colicchia, C. & Strozzi, F. (2012), 'Supply chain risk management: a new methodology for a systematic literature review', *Supply Chain Management: An International Journal* .

Colson, B., Marcotte, P. & Savard, G. (2005), 'Bilevel programming: A survey', *4or* **3**(2), 87–107.

Colson, B., Marcotte, P. & Savard, G. (2007), 'An overview of bilevel optimization', *Annals of operations research* **153**(1), 235–256.

Coniglio, S., Sirvent, M. & Weibelzahl, M. (2021), 'Airport capacity extension, fleet investment, and optimal aircraft scheduling in a multilevel market model: quantifying the costs of imperfect markets', *OR Spectrum* **43**(2), 367–408.

Cottrill, C. A., Rogers, E. M. & Mills, T. (1989), 'Co-citation analysis of the scientific literature of innovation research traditions: Diffusion of innovations and technology transfer', *Knowledge* **11**(2), 181–208.

Culnan, M. J. (1986), 'The intellectual development of management information systems, 1972–1982: A co-citation analysis', *Management science* **32**(2), 156–172.

Davarikia, H. & Barati, M. (2018), 'A tri-level programming model for attack-resilient control of power grids', *Journal of Modern Power Systems and Clean Energy* **6**(5), 918–929.

Dehghan, S., Amjady, N. & Conejo, A. J. (2015), 'Reliability-constrained robust power system expansion planning', *IEEE Transactions on Power Systems* **31**(3), 2383–2392.

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

Dempe, S. (2003), 'Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints'.

Dempe, S. & Dutta, J. (2012), 'Is bilevel programming a special case of a mathematical program with complementarity constraints?', *Mathematical programming* **131**(1), 37–48.

Dempe, S., Kalashnikov, V. V. & Kalashnykova, N. (2006), Optimality conditions for bilevel programming problems, *in* 'Optimization with Multivalued Mappings', Springer, pp. 3–28.

Dempe, S., Kalashnikov, V. V., Pérez-Valdés, G. A. & Kalashnykova, N. I. (2011), 'Natural gas bilevel cash-out problem: convergence of a penalty function method', *European Journal of Operational Research* **215**(3), 532–538.

Dempe, S., Mordukhovich, B. S. & Zemkoho, A. B. (2014), 'Necessary optimality conditions in pessimistic bilevel programming', *Optimization* **63**(4), 505–533.

DeNegre, S. T. & Ralphs, T. K. (2009), A branch-and-cut algorithm for integer bilevel linear programs, *in* 'Operations research and cyber-infrastructure', Springer, pp. 65–78.

Dey, G. C. & Jenamani, M. (2019), 'Optimizing fortification plan of capacitated facilities with maximum distance limits', *Opsearch* **56**(1), 151–173.

Ding, T., Li, C., Chen, C., Chen, B. & Yang, Y. (2019), 'Risk assessment-based long-term transmission system hardening under prior probabilistic information', *IET Generation, Transmission & Distribution* **13**(1), 108–115.

Ding, T., Yao, L. & Li, F. (2018), 'A multi-uncertainty-set based two-stage robust optimization to defender–attacker–defender model for power system protection', *Reliability Engineering & System Safety* **169**, 179–186.

Ebrahimi, M. R. & Amjady, N. (2019), 'Adaptive robust optimization framework for day-ahead microgrid scheduling', *International Journal of Electrical Power & Energy Systems* **107**, 213–223.

Edmunds, T. A. & Bard, J. F. (1991), 'Algorithms for nonlinear bilevel mathematical programs', *IEEE transactions on Systems, Man, and Cybernetics* **21**(1), 83–89.

Eksoz, C., Mansouri, S. A. & Bourlakis, M. (2014), 'Collaborative forecasting in the food supply chain: a conceptual framework', *International journal of production economics* **158**, 120–135.

Fahimnia, B., Sarkis, J. & Davarzani, H. (2015), 'Green supply chain management: A review and bibliometric analysis', *International Journal of Production Economics* **162**, 101–114.

Faísca, N. P., Dua, V., Rustem, B., Saraiva, P. M. & Pistikopoulos, E. N. (2007), 'Parametric global optimisation for bilevel programming', *Journal of Global Optimization* **38**(4), 609–623.

Fakhry, R., Hassini, E., Ezzeldin, M. & El-Dakhakhni, W. (2022), 'Tri-level mixed-binary linear programming: Solution approaches and application in defending critical infrastructure', *European Journal of Operational Research* **298**(3), 1114–1131.

Fang, Y.-P. & Zio, E. (2019), 'An adaptive robust framework for the optimization of the resilience of interdependent infrastructures under natural hazards', *European Journal of Operational Research* **276**(3), 1119–1136.

Fang, Y. & Sansavini, G. (2017), 'Optimizing power system investments and resilience against attacks', *Reliability Engineering & System Safety* **159**, 161–173.

Fanzeres, B., Ahmed, S. & Street, A. (2019), 'Robust strategic bidding in auction-based markets', *European Journal of Operational Research* **272**(3), 1158–1172.

Feng, X., He, S., Chen, X. & Li, G. (2021), 'Mitigating the vulnerability of an air-high-speed railway transportation network: From the perspective of predisruption response', *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* **235**(3), 474–490.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2016), Intersection cuts for bilevel optimization, *in* 'International Conference on Integer Programming and Combinatorial Optimization', Springer, pp. 77–88.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2017), 'A new general-purpose algorithm for mixed-integer bilevel linear programs', *Operations Research* **65**(6), 1615–1637.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2018), 'On the use of intersection cuts for bilevel optimization', *Mathematical Programming* **172**(1), 77–103.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2019), 'Interdiction games and monotonicity, with application to knapsack problems', *INFORMS Journal on Computing* **31**(2), 390–410.

Florensa, C., Garcia-Herreros, P., Misra, P., Arslan, E., Mehta, S. & Grossmann, I. E. (2017), 'Capacity planning with competitive decision-makers: trilevel milp formulation, degeneracy, and solution approaches', *European Journal of Operational Research* **262**(2), 449–463.

Fortuny-Amat, J. & McCarl, B. (1981), 'A representation and economic interpretation of a two-level programming problem', *Journal of the operational Research Society* **32**(9), 783–792.

Gabriel, S. A., Leal, M. & Schmidt, M. (2021), 'On linear bilevel optimization problems with complementarity-constrained lower levels', *Journal of the Operational Research Society* pp. 1–11.

Gadhi, N. & Dempe, S. (2012), 'Necessary optimality conditions and a new approach to multiobjective bilevel optimization problems', *Journal of Optimization Theory and Applications* **155**(1), 100–114.

Garfield, E. (1995), 'Quantitative analysis of the scientific literature and its implications for science policymaking in latin america and the caribbean', *Bulletin of the Pan American Health Organization (PAHO); 29 (1), mar. 1995* .

Geißler, B., Martin, A., Morsi, A. & Schewe, L. (2012), Using piecewise linear functions for solving minlp s, *in* 'Mixed integer nonlinear programming', Springer, pp. 287–314.

Ghorbani-Renani, N., González, A. D., Barker, K. & Morshedlou, N. (2020), ‘Protection-interdiction-restoration: Tri-level optimization for enhancing interdependent network resilience’, *Reliability Engineering & System Safety* **199**, 106907.

Glover, F. (1973), ‘Convexity cuts and cut search’, *Operations Research* **21**(1), 123–134.

Glover, F. (1974), ‘Polyhedral convexity cuts and negative edge extensions’, *Zeitschrift für Operations Research* **18**(5), 181–186.

Govindan, K., Fattahi, M. & Keyvanshokooh, E. (2017), ‘Supply chain network design under uncertainty: A comprehensive review and future research directions’, *European journal of operational research* **263**(1), 108–141.

Gu, Y., Cai, X., Han, D. & Wang, D. Z. (2019), ‘A tri-level optimization model for a private road competition problem with traffic equilibrium constraints’, *European Journal of Operational Research* **273**(1), 190–197.

Gunasekaran, A., Subramanian, N. & Rahman, S. (2015), ‘Green supply chain collaboration and incentives: Current trends and future directions’.

Han, C. Y., Lunday, B. J. & Robbins, M. J. (2016), ‘A game theoretic model for the optimal location of integrated air defense system missile batteries’, *INFORMS Journal on Computing* **28**(3), 405–416.

Hansen, P., Jaumard, B. & Savard, G. (1992), ‘New branch-and-bound rules for linear bilevel programming’, *SIAM Journal on scientific and Statistical Computing* **13**(5), 1194–1217.

Hasanzad, F. & Rastegar, H. (2022), 'Application of optimal hardening for improving resilience of integrated power and natural gas system in case of earthquake', *Reliability Engineering & System Safety* **223**, 108476.

Hooker, J. N. (2013), Decision diagrams and dynamic programming, *in* 'International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research', Springer, pp. 94–110.

Huppmann, D. & Siddiqui, S. (2018), 'An exact solution method for binary equilibrium problems with compensation and the power market uplift problem', *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* **266**(2), 622–638.

Inuiguchi, M. & Sariddichainunta, P. (2016), 'Bilevel linear programming with ambiguous objective function of the follower', *Fuzzy Optimization and Decision Making* **15**(4), 415–434.

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

Jeroslow, R. G. (1985), 'The polynomial hierarchy and a simple model for competitive analysis', *Mathematical programming* **32**(2), 146–164.

Jin, J. G., Lu, L., Sun, L. & Yin, J. (2015), 'Optimal allocation of protective resources in urban rail transit networks against intentional attacks', *Transportation Research Part E: Logistics and Transportation Review* **84**, 73–87.

Kaviani, A., Thompson, R. G., Rajabifard, A. & Sarvi, M. (2018), 'Hybrid machine learning and optimisation method to solve a tri-level road network protection problem', *IET Intelligent Transport Systems* **12**(9), 1011–1019.

Ke, G. Y. & Bookbinder, J. H. (2018), 'Coordinating the discount policies for retailer, wholesaler, and less-than-truckload carrier under price-sensitive demand: A tri-level optimization approach', *International Journal of Production Economics* **196**, 82–100.

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**, 100007.

Kleinert, T., Labbé, M., Plein, F. & Schmidt, M. (2021), 'Closing the gap in linear bilevel optimization: a new valid primal-dual inequality', *Optimization Letters* **15**(4), 1027–1040.

Kleniati, P.-M. & Adjiman, C. S. (2014), 'Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. part i: Theoretical development', *Journal of Global Optimization* **60**(3), 425–458.

Koltsaklis, N. E. & Dagoumas, A. S. (2018), 'State-of-the-art generation expansion planning: A review', *Applied energy* **230**, 563–589.

Konur, D. & Golias, M. M. (2013), 'Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty', *Computers & Industrial Engineering* **65**(4), 663–672.

Labbé, M., Marcotte, P. & Savard, G. (1998), 'A bilevel model of taxation and its application to optimal highway pricing', *Management science* **44**(12-part-1), 1608–1622.

Lazzaro, G. & Carlyle, W. M. (2019), 'Finding the best system nested defense under budget uncertainty', *Military Operations Research* **24**(1), 5–22.

Lei, X., Shen, S. & Song, Y. (2018), 'Stochastic maximum flow interdiction problems under heterogeneous risk preferences', *Computers & Operations Research* **90**, 97–109.

Leitmann, G. (1978), 'On generalized stackelberg strategies', *Journal of optimization theory and applications* **26**(4), 637–643.

Li, G. & Wan, Z. (2018), 'On bilevel programs with a convex lower-level problem violating slater's constraint qualification', *Journal of Optimization Theory and Applications* **179**(3), 820–837.

Li, G., Wan, Z., Chen, J.-w. & Zhao, X. (2020), 'Necessary optimality condition for trilevel optimization problem', *Journal of Industrial & Management Optimization* **16**(1), 55.

Li, L. (2021), 'Coordination between smart distribution networks and multi-microgrids considering demand side management: A trilevel framework', *Omega* **102**, 102326.

Liberatore, F., Scaparra, M. P. & Daskin, M. S. (2012), 'Hedging against disruptions with ripple effects in location analysis', *Omega* **40**(1), 21–30.

Lin, Y. & Bie, Z. (2018), 'Tri-level optimal hardening plan for a resilient distribution system considering reconfiguration and dg islanding', *Applied Energy* **210**, 1266–1279.

Liu, H. & Wang, D. Z. (2017), 'Locating multiple types of charging facilities for battery electric vehicles', *Transportation Research Part B: Methodological* **103**, 30–55.

Liu, J., Fan, Y., Chen, Z. & Zheng, Y. (2018), 'Pessimistic bilevel optimization: a survey', *International Journal of Computational Intelligence Systems* **11**(1), 725.

Liu, S., Wang, M., Kong, N. & Hu, X. (2021), 'An enhanced branch-and-bound algorithm for bilevel integer linear programming', *European Journal of Operational Research* **291**(2), 661–679.

Loridan, P. & Morgan, J. (1996), 'Weak via strong stackelberg problem: new results', *Journal of global Optimization* **8**(3), 263–287.

Lozano, L. & Smith, J. C. (2017), 'A backward sampling framework for interdiction problems with fortification', *INFORMS Journal on Computing* **29**(1), 123–139.

Lu, J., Han, J., Hu, Y. & Zhang, G. (2016), 'Multilevel decision-making: A survey', *Information Sciences* **346**, 463–487.

Ma, S., Chen, B. & Wang, Z. (2016), 'Resilience enhancement strategy for distribution systems under extreme weather events', *IEEE Transactions on Smart Grid* **9**(2), 1442–1451.

Mendes, G., Ioakimidis, C. & Ferrão, P. (2011), 'On the planning and analysis of integrated community energy systems: A review and survey of available tools', *Renewable and Sustainable Energy Reviews* **15**(9), 4836–4854.

Misaghian, M., Saffari, M., Kia, M., Heidari, A., Shafie-khah, M. & Catalão, J. P. (2018), 'Tri-level optimization of industrial microgrids considering renewable energy sources, combined heat and power units, thermal and electrical storage systems', *Energy* **161**, 396–411.

Mishra, D., Gunasekaran, A., Papadopoulos, T. & Dubey, R. (2018), 'Supply chain performance measures and metrics: a bibliometric study', *Benchmarking: An International Journal* .

Mitsos, A. (2010), 'Global solution of nonlinear mixed-integer bilevel programs', *Journal of Global Optimization* **47**(4), 557–582.

Moore, J. T. & Bard, J. F. (1990), 'The mixed integer linear bilevel programming problem', *Operations research* **38**(5), 911–921.

Moreira, A., Street, A. & Arroyo, J. M. (2014), 'An adjustable robust optimization approach for contingency-constrained transmission expansion planning', *IEEE Transactions on Power Systems* **30**(4), 2013–2022.

Moreira, A., Street, A. & Arroyo, J. M. (2015), 'Energy and reserve scheduling under correlated nodal demand uncertainty: An adjustable robust optimization approach', *International Journal of Electrical Power & Energy Systems* **72**, 91–98.

Munoz-Delgado, G., Contreras, J. & Arroyo, J. M. (2019), 'Distribution system expansion planning considering non-utility-owned dg and an independent distribution system operator', *IEEE Transactions on Power Systems* **34**(4), 2588–2597.

Mycielski, J. (1992), 'Games with perfect information', *Handbook of game theory with economic applications* **1**, 41–70.

Nemati, H., Latify, M. A. & Yousefi, G. R. (2019), 'Tri-level transmission expansion planning under intentional attacks: virtual attacker approach–part i: formulation', *IET Generation, Transmission & Distribution* **13**(3), 390–398.

Nerur, S. P., Rasheed, A. A. & Natarajan, V. (2008), 'The intellectual structure of the strategic management field: An author co-citation analysis', *Strategic Management Journal* **29**(3), 319–336.

Ouyang, M. (2017), 'A mathematical framework to optimize resilience of interdependent critical infrastructure systems under spatially localized attacks', *European Journal of Operational Research* **262**(3), 1072–1084.

Ouyang, M., Xu, M., Zhang, C. & Huang, S. (2017), 'Mitigating electric power system vulnerability to worst-case spatially localized attacks', *Reliability Engineering & System Safety* **165**, 144–154.

Parajuli, A., Kuzgunkaya, O. & Vidyarthi, N. (2017), 'Responsive contingency planning of capacitated supply networks under disruption risks', *Transportation Research Part E: Logistics and Transportation Review* **102**, 13–37.

Parajuli, A., Kuzgunkaya, O. & Vidyarthi, N. (2021), 'The impact of congestion on protection decisions in supply networks under disruptions', *Transportation Research Part E: Logistics and Transportation Review* **145**, 102166.

Parvasi, S. P., Tavakkoli-Moghaddam, R., Bashirzadeh, R., Taleizadeh, A. A. & Baboli, A. (2019), 'Designing a model for service facility protection with a time horizon based on tri-level programming', *Engineering Optimization* .

Pilkington, A. & Fitzgerald, R. (2006), 'Operations management themes, concepts and relationships: a forward retrospective of ijopm', *International Journal of Operations & Production Management* .

Pilkington, A. & Meredith, J. (2009), 'The evolution of the intellectual structure of operations management—1980–2006: A citation/co-citation analysis', *Journal of operations management* **27**(3), 185–202.

Powell, W. B. (2019), 'A unified framework for stochastic optimization', *European Journal of Operational Research* **275**(3), 795–821.

Rahdar, M., Wang, L. & Hu, G. (2018), 'A tri-level optimization model for inventory control with uncertain demand and lead time', *International Journal of Production Economics* **195**, 96–105.

Ramamoorthy, P., Jayaswal, S., Sinha, A. & Vidyarthi, N. (2018), 'Multiple allocation hub interdiction and protection problems: Model formulations and solution approaches', *European Journal of Operational Research* **270**(1), 230–245.

Ramos-Rodríguez, A.-R. & Ruíz-Navarro, J. (2004), 'Changes in the intellectual structure of strategic management research: A bibliometric study of the strategic management journal, 1980–2000', *Strategic management journal* **25**(10), 981–1004.

Roghanian, E., Sadjadi, S. J. & Aryanezhad, M.-B. (2007), 'A probabilistic bi-level linear multi-objective programming problem to supply chain planning', *Applied Mathematics and computation* **188**(1), 786–800.

Roldán, C., Mínguez, R., García-Bertrand, R. & Arroyo, J. M. (2018), 'Robust transmission network expansion planning under correlated uncertainty', *IEEE Transactions on Power Systems* **34**(3), 2071–2082.

Ruiz, C. & Conejo, A. J. (2015), 'Robust transmission expansion planning', *European Journal of Operational Research* **242**(2), 390–401.

Sadati, M. E. H., Aksen, D. & Aras, N. (2020), 'A trilevel r-interdiction selective multi-depot vehicle routing problem with depot protection', *Computers & operations research* **123**, 104996.

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

Saharidis, G. K. & Ierapetritou, M. G. (2009), 'Resolution method for mixed integer bi-level linear problems based on decomposition technique', *Journal of Global Optimization* **44**(1), 29–51.

Sahinidis, N. V. (2004), 'Optimization under uncertainty: state-of-the-art and opportunities', *Computers & Chemical Engineering* **28**(6-7), 971–983.

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

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

Sarhadi, H., Tulett, D. M. & Verma, M. (2017), 'An analytical approach to the protection planning of a rail intermodal terminal network', *European Journal of Operational Research* **257**(2), 511–525.

Sariddichainunta, P. & Inuiguchi, M. (2017), 'Global optimality test for maximin solution of bilevel linear programming with ambiguous lower-level objective function', *Annals of Operations Research* **256**(2), 285–304.

Scaparra, M. P. & Church, R. (2012), 'Protecting supply systems to mitigate potential disaster: a model to fortify capacitated facilities', *International Regional Science Review* **35**(2), 188–210.

Scaparra, M. P. & Church, R. L. (2008), 'A bilevel mixed-integer program for critical infrastructure protection planning', *Computers & Operations Research* **35**(6), 1905–1923.

Schweitzer, D. & Medal, H. (2019), 'Wireless lan transmitter location under the threat of jamming attacks', *Computers & Operations Research* **106**, 14–27.

Street, A., Moreira, A. & Arroyo, J. M. (2013), 'Energy and reserve scheduling under a joint generation and transmission security criterion: An adjustable robust optimization approach', *IEEE transactions on Power Systems* **29**(1), 3–14.

Tahernejad, S., Ralphs, T. K. & DeNegre, S. T. (2020), ‘A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation’, *Mathematical Programming Computation* **12**(4), 529–568.

Tian, Q., Wang, D. Z. & Lin, Y. H. (2021), ‘Service operation design in a transit network with congested common lines’, *Transportation Research Part B: Methodological* **144**, 81–102.

Tippmann, S. (2015), ‘Programming tools: Adventures with r’, *Nature* **517**(7532), 109–110.

Vicente, L., Savard, G. & Judice, J. (1996), ‘Discrete linear bilevel programming problem’, *Journal of optimization theory and applications* **89**(3), 597–614.

von Stackelberg, H. (2011), *Market Structure and Equilibrium*, Springer Berlin Heidelberg, Berlin, Heidelberg.

Wang, C., Wei, W., Wang, J., Liu, F., Qiu, F., Correa-Posada, C. M. & Mei, S. (2016), ‘Robust defense strategy for gas–electric systems against malicious attacks’, *IEEE Transactions on Power Systems* **32**(4), 2953–2965.

Wang, L. & Xu, P. (2017), ‘The watermelon algorithm for the bilevel integer linear programming problem’, *SIAM Journal on Optimization* **27**(3), 1403–1430.

Wang, S., Qi, J. & Laporte, G. (2022), ‘Governmental subsidy plan modeling and optimization for liquefied natural gas as fuel for maritime transportation’, *Transportation Research Part B: Methodological* **155**, 304–321.

Wang, Y., Jiao, Y.-C. & Li, H. (2005), 'An evolutionary algorithm for solving non-linear bilevel programming based on a new constraint-handling scheme', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **35**(2), 221–232.

Wen, U.-P. & Hsu, S.-T. (1991), 'Linear bi-level programming problems—a review', *Journal of the Operational Research Society* **42**(2), 125–133.

Wen, U.-P. & Hsu, S.-T. (1992), 'Efficient solutions for the linear bilevel programming problem', *European Journal of Operational Research* **62**(3), 354–362.

White, D. (1997), 'Penalty function approach to linear trilevel programming', *Journal of Optimization Theory and Applications* **93**(1), 183–197.

Wiesemann, W., Tsoukalas, A., Kleniati, P.-M. & Rustem, B. (2013), 'Pessimistic bilevel optimization', *SIAM Journal on Optimization* **23**(1), 353–380.

Wu, D., Yin, Y. & Lawphongpanich, S. (2011), 'Optimal selection of build–operate-transfer projects on transportation networks', *Transportation Research Part B: Methodological* **45**(10), 1699–1709.

Wu, X. & Conejo, A. J. (2017), 'An efficient tri-level optimization model for electric grid defense planning', *IEEE Transactions on Power Systems* **32**(4), 2984–2994.

Wu, Y., Chen, Z., Dang, J., Chen, Y., Zhao, X. & Zha, L. (2022), 'Allocation of defensive and restorative resources in electric power system against consecutive multi-target attacks', *Reliability Engineering & System Safety* **219**, 108199.

Wu, Y., Chen, Z., Gong, H., Feng, Q., Chen, Y. & Tang, H. (2021), ‘Defender–attacker–operator: Tri-level game-theoretic interdiction analysis of urban water distribution networks’, *Reliability Engineering & System Safety* **214**, 107703.

Xu, P. & Wang, L. (2014), ‘An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions’, *Computers & operations research* **41**, 309–318.

Yan, Y., Zhang, C., Li, K. & Wang, Z. (2018), ‘An integrated design for hybrid combined cooling, heating and power system with compressed air energy storage’, *Applied Energy* **210**, 1151–1166.

Yao, Y., Edmunds, T., Papageorgiou, D. & Alvarez, R. (2007), ‘Trilevel optimization in power network defense’, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **37**(4), 712–718.

Yin, Y. (2000), ‘Genetic-algorithms-based approach for bilevel programming models’, *Journal of transportation engineering* **126**(2), 115–120.

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

Zemkoho, A. B. (2016), ‘Solving ill-posed bilevel programs’, *Set-Valued and Variational Analysis* **24**(3), 423–448.

Zeng, B. & An, Y. (2014), ‘Solving bilevel mixed integer program by reformulations and decomposition’, *Optimization online* pp. 1–34.

Zhang, L., Lawphongpanich, S. & Yin, Y. (2009), An active-set algorithm for discrete network design problems, *in* 'Transportation and Traffic Theory 2009: Golden Jubilee', Springer, pp. 283–300.

Zhu, H., Zhang, C., Ramirez-Marquez, J. E., Wu, S. & Monroy, R. (2021), 'The integration of protection, restoration, and adaptive flow redistribution in building resilient networked critical infrastructures against intentional attacks', *IEEE Systems Journal* **15**(2), 2959–2970.

Chapter 3

Heuristic-Based Approaches for Solving Mixed-Binary Tri-Level Programs

Abstract

Decentralized decision-making is becoming more ubiquitous in different organizations that often follow a hierarchical structure. To model these problems, multi-level programming has been suggested as a suitable methodology for modelling the interaction between the different levels of decisions. However, multi-level programming, even for the case of bi-levels, is known to be strongly $\mathcal{NP}$-hard. To address this computational challenge, we develop three different heuristic-based approaches for solving a specific class of tri-level programming problems, in which the leader has direct control over the follower's decisions to a certain extent, with a common objective function shared at all levels. As expected, each heuristic type offers a trade-off between solution quality and computational time. To illustrate our solution approach, we present an application for defending critical infrastructure to improve its resilience against intentional attacks. In this context we use a defender-attacker-defender model and apply it to electrical power grids. We also propose a modified implementation of a widely adopted enumeration algorithm in this area, with a warm-starting solution technique that significantly enhanced the computational performance of the enumeration algorithm. We test our solution approaches on three electrical transmission networks and present the results of our numerical computations as well as some insights.

The work in this chapter appeared in *European Journal of Operational Research* (Volume: 298, Number: 3, Pages: 1114 – 1131, 2022) (Fakhry et al. 2022). Minor modifications have been made to improve the clarity of this work. EJOR owns the copyright of the material in this chapter and it is permitted to be re-used in this thesis. The digital object identifier (**DOI**) link is https://doi.org/10.1016/j.ejor.2021.07.034 .

3.1 Introduction

Multi-level programming (**MLP**) problems, first introduced by Candler & Norton (1977), have been used to model decentralized planning problems that involve several decision-makers. This class of optimization problems has been recently receiving significant attention from researchers in different fields. The interest has grown mainly due to the applicability of MLP to a wide range of problems and the nature of decision-making process that often takes a hierarchical structure. MLP results in nested mathematical programs (lower-levels) having a subset of their decision variables affecting the optimal solution of other mathematical programs (upper-levels), which have their own set of decision variables influencing their objective function and higher-level programs. Bi-level programming (**BLP**) problem is a special case of MLPs, when there are only two decision-makers. A generic (**BLP**) can be formulated as

$$\begin{aligned} &\text{`}\max_{\mathbf{x}}\text{'} \quad f_1(\mathbf{x},\mathbf{y}) \\ &\text{s.t.} \quad g_1(\mathbf{x},\mathbf{y}) \geq 0 \\ &\qquad \mathbf{y} \in \arg\min \quad f_2(\mathbf{x},\mathbf{y}) \\ &\qquad\qquad \text{s.t.} \quad g_2(\mathbf{x},\mathbf{y}) \geq 0 \end{aligned} \tag{3.1}$$

The above BLP is often referred to as a leader-follower model, where the leader takes the first move by controlling a set of decision variables **x** to maximize their objective function. The follower reacts to the leader's move by adjusting their own set of decision variables **y** to optimize the objective value. The quotation marks in 3.1 are used to indicate the ambiguity in the formulation of the leader's problem. The ambiguity arises when the follower has to choose between more than one optimal reaction (i.e., the follower's problem is a non-strictly convex minimization problem that might have several alternative global optima, in which case, a problem is said to be *"degenerate"*). In order for the model to be well defined, the follower has to choose between alternative optima, which leads to two approaches, namely: optimistic approach (Dempe 2002) and pessimistic approach (Aussel & Svensson 2019). For more information on the foundations of BLP, the interested reader can refer to (Dempe 2002) and (Bard 2013). In this manner, BLP mimics a sequential two-person game, in which both players have perfect information, known as the static Stackelberg game in the field of game theory (von Stackelberg 2011). The interest in solving BLP as optimization problems has started mainly in the 1980s (Vicente & Calamai 1994). It was later established that BLP, even in their simplest forms, where decision variables are continuous and functions and constraints are linear, are $\mathcal{NP}$-hard and that the feasible region is non-convex (Bard 1991).

Several solution approaches for **BLP** have been proposed in the literature. Moore & Bard (1990) provided an algorithm that relies on a branch and bound method to solve mixed-integer linear BLP, in which decision variables of both levels are mixed-integer. Bard & Moore (1992) provided an algorithm that solves linear BLP where decision variables in both levels are binary. Their idea relied on including the leader's objective function as a "parameterized constraint", and use the resultant problem to produce a solution, which in turn is used to obtain a feasible solution to the original BLP. Recently, an exact algorithm has been developed to solve the mixed-integer linear BLP, with fewer restrictions and assumptions (Xu & Wang 2014). Colson et al. (2005) and Colson et al. (2007) provide more comprehensive reviews on solution algorithms and applications of BLP.

Tri-level programming (**TLP**) problems not only inherits all the properties of BLP, but also add to the hierarchical structure one upper-level along with its associated set of decision variables, constraints and objective function. In most studies on tri-level programming, it is implicitly assumed that an optimal solution of lower-levels' objectives for each decision made at the upper-levels is unique (Sarhadi et al. (2017);Yao et al. (2007); Alguacil et al. (2014);Schweitzer & Medal (2019)). This is generally not always true, as any non-strictly convex (concave) minimization (maximization) might have multiple optimal solutions. In a tri-level problem, the selection of alternative optima (i.e., degenerate solutions) at a particular level yields the same results for that level. However, each of the alternatives has a different impact on the overall problem Florensa et al. (2017). That is why it is important to determine the selection criteria for the upper-level decision-maker(s) among the different solution alternatives in the lower-levels, otherwise the tri-level model would be ill-posed. A TLP without selecting an appropriate selection approach represented by quotation marks in a manner similar to BLP, can be modeled as follows:

$$
\begin{array}{lllll}
\text{`}\min_{\mathbf{x}}\text{'} & f_1(\mathbf{x},\mathbf{y},\mathbf{z}) & & & \\
\text{s.t.} & g_1(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0 & & & \\
 & \mathbf{y} \in \text{`}\arg\max\text{'} & f_2(\mathbf{x},\mathbf{y},\mathbf{z}) & & \\
 & \quad\text{s.t.} & g_2(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0 & & \\
 & & \mathbf{z} \in \arg\min & f_3(\mathbf{x},\mathbf{y},\mathbf{z}) & \\
 & & \quad\text{s.t.} & g_3(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0 &
\end{array}
\tag{3.2}
$$

The most complex and general form is when each decision-maker has mixed-integer decision variables. Below we state several definitions which we will make use of in our subsequent analysis given the TLP presented in 3.2.

Definition 3.1.1. *S be the **tri-level constraint region**:*

$$S = \{(\mathbf{x},\mathbf{y},\mathbf{z}) \in X \times Y \times Z : \ g_1(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0, \ g_2(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0, \ g_3(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0\} \tag{3.3}$$

Definition 3.1.2. *$S_{\mathbf{z}}(\mathbf{x},\mathbf{y})$ be the **third-level constraint region** for fixed $(\mathbf{x},\mathbf{y}) \in X \times Y$:*

$$S_{\mathbf{z}}(\mathbf{x},\mathbf{y}) = \{z \in Z : \ g_3(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0\} \tag{3.4}$$

Definition 3.1.3. *$M_{\mathbf{z}}(\mathbf{x},\mathbf{y})$ be the **basic rational reaction set of the third-level** for fixed $(\mathbf{x},\mathbf{y}) \in X \times Y$:*

$$M_{\mathbf{z}}(\mathbf{x},\mathbf{y}) = \arg\min_{\mathbf{z} \in Z}\{f_3(\mathbf{x},\mathbf{y},\mathbf{z}) : \ g_3(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0\} \tag{3.5}$$

or in a more compact form:

$$M_{\mathbf{z}}(\mathbf{x},\mathbf{y}) = \operatorname*{arg\,min}_{\mathbf{z}\in S_{\mathbf{z}}(\mathbf{x},\mathbf{y})}\{f_3(\mathbf{x},\mathbf{y},\mathbf{z})\} \tag{3.6}$$

Definition 3.1.4. $S_{\mathbf{y},\mathbf{z}}(\mathbf{x})$ *be the* ***union of second- and third-level constraints region*** *for fixed* $\mathbf{x} \in X$*:*

$$S_{\mathbf{y},\mathbf{z}}(\mathbf{x}) = \{(\mathbf{y},\mathbf{z}) \in Y \times Z : \ \ g_2(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0,\ g_3(\mathbf{x},\mathbf{y},\mathbf{z}) \geq 0\} \tag{3.7}$$

Definition 3.1.5. $M_{\mathbf{y}}(\mathbf{x})$ *be the* ***"basic rational reaction set of the second-level"*** *for fixed* $\mathbf{x} \in X$*:*

$$M_{\mathbf{y}}(\mathbf{x}) = `\operatorname*{arg\,max}_{\mathbf{y}\in S_{\mathbf{y},\mathbf{z}}(\mathbf{x})}'\{f_2(\mathbf{x},\mathbf{y},\mathbf{z}) : \mathbf{z} \in M_{\mathbf{z}}(\mathbf{x},\mathbf{y})\} \tag{3.8}$$

The quotation marks for the basic rational reaction set of the second-level is to indicate its ill-definition, in-case of degenerate solutions of the third-level. Nevertheless, to represent the next definitions in a compact form, we take the aforementioned definitions to be well-defined.

Definition 3.1.6. *The optimal solution to a tri-level program is said to be* ***sequentially pessimistic*** *(Sp), if degenerate solutions in the third-level are resolved against the second-level decision-maker, and multiple optima in the second-level are resolved against the first-level decision-maker.*

- *The* ***sequentially pessimistic*** *reaction set of the third-level for fixed* $(\mathbf{x},\mathbf{y}) \in$

$X \times Y$ can be represented as:

$$M^{Sp}_{\mathrm{II},\mathbf{z}}(\mathbf{x},\mathbf{y}) = \underset{\mathbf{z}\in M_{\mathbf{z}}(\mathbf{x},\mathbf{y})}{\arg\min}\{f_2(\mathbf{x},\mathbf{y},\mathbf{z})\} \tag{3.9}$$

The subscript II *is used to indicate that degeneracy is resolved with respect to the second level of TLP, while the superscript Sp represents that the resolution is determined in a pessimistic way (i.e., against the second-level decision-maker).*

- *The **sequentially pessimistic** reaction set of the second-level for fixed $\mathbf{x} \in X$:*

$$M^{Sp}_{\mathrm{I},\mathbf{y}}(\mathbf{x}) = \underset{\mathbf{y}\in M_{\mathbf{y}}(\mathbf{x})}{\arg\max}\{f_1(\mathbf{x},\mathbf{y},\mathbf{z}) : z \in M^{Sp}_{\mathrm{II},\mathbf{z}}\} \tag{3.10}$$

Definition 3.1.7. *The **Inducible Region** of a **sequentially pessimistic** TLP is*

$$IR = \{(\mathbf{x},\mathbf{y},\mathbf{z}) \in X \times Y \times Z : (\mathbf{x},\mathbf{y},\mathbf{z}) \in S, (\mathbf{y},\mathbf{z}) \in M^{Sp}_{\mathrm{I},\mathbf{y}}(\mathbf{x})\}. \tag{3.11}$$

Definition 3.1.8. ***Optimal solution** set of a **sequentially pessimistic** TLP is*

$$OP = \{(\mathbf{x},\mathbf{y},\mathbf{z}) : \mathbf{x} \in \arg\min\{f_1(\mathbf{x},\mathbf{y},\mathbf{z}) : (\mathbf{x},\mathbf{y},\mathbf{z}) \in IR\}\}. \tag{3.12}$$

The computational complexity of MLPs increases significantly, if the hierarchical structure consists of more than two levels (Blair 1992). From a game-theoretic perspective, a TLP can be regarded as a leader-follower-leader game. In such a game, the leader gets to pick two sets of actions: one before the follower reveals their move, and another after the revelation of follower's move. The game can also be generalized where all players are different.

There have also been several approaches for solving TLP. Bard (1984) provided an algorithm to solve the linear and continuous case of TLP. White (1997) modified the aforementioned algorithm by introducing penalty functions Anandalingam (1988) and Sinha (2001) have used Karush-Khun-Tucker **(KKT)**transformations to find local optimal solutions for linear TLP. Recently, Han et al. (2016) devised a particle swarm optimization algorithm for solving BLP and TLP when the decision variables are continuous for all decision-makers, and a specific set of assumptions has to be satisfied for the objective functions of the leader and followers. In general, TLP are strongly $\mathcal{NP}$-hard and the existing traditional solution approaches either do not guarantee optimality, or are computationally expensive. Subsequently, as per Scaparra & Church (2008), there is still a lack of efficient algorithms for solving three-stage problems.

To address this need, in this paper we present three main contributions:

- We propose three different heuristic-based approaches for solving a specific class of mixed-binary linear TLPs. Each has a trade-off between solution quality and computational time.

- We propose a modified exact enumeration algorithm for a specific class of TLP and present our related numerical computations.

- The solution approaches are implemented to solve a TLP to enhance the resilience of critical infrastructure. In particular, we test our algorithms on three electrical transmission networks, that vary in size, to improve the robustness and reliability of power grids.

The rest of the paper is organized as follows. In Section 3.2 we introduce the details of our class of TLP, practical applications that fall within that class, and provide two approaches to reduce the TLP. Our three solution approaches are discussed in Section 3.3. In Section 3.4 we present an application of TLP as a defender-attacker-defender model for electric power grid security and a modified exact enumeration algorithm, along with our numerical results. Finally, conclusions are presented Section 3.5.

3.2 Class of TLP Being Studied

In this section we define and analyze the class of TLP problems that will be studied. We will also outline some general application areas.

3.2.1 Generic TLP Model and Assumptions

In this paper we present and study the following class of sequentially pessimistic mixed-binary linear TLP:

$$\begin{aligned}
\min_{\mathbf{x}} \quad & f_1(\mathbf{x},\mathbf{y},\mathbf{z}) = f(\mathbf{z}) \\
\text{s.t.} \quad & g_1(\mathbf{x}) \geq 0 \\
& \mathbf{x} \in \{0,1\} \\
& \mathbf{y} \in \arg\max \quad f_2(\mathbf{x},\mathbf{y},\mathbf{z}) = f(\mathbf{z}) \\
& \qquad \text{s.t.} \quad g_2(\mathbf{x},\mathbf{y}) \geq 0 \\
& \qquad\qquad \mathbf{y} \in \{0,1\} \\
& \qquad\qquad \mathbf{z} \in \arg\min \quad f_3(\mathbf{x},\mathbf{y},\mathbf{z}) = f(\mathbf{z}) \\
& \qquad\qquad\qquad \text{s.t.} \quad g_3(\mathbf{y},\mathbf{z}) \geq 0
\end{aligned} \tag{3.13}$$

In TLP (3.13), we make the following assumptions to simplify degeneracy resolution as explained in detail in Section 3.4.2.

Assumption 3.2.1. *The hierarchical structure has one-way input decisions with binary variables* $\mathbf{x}$ *and* $\mathbf{y}$ *that are shared with the next lower-level constraints as shown in Figure 3.1.*

Assumption 3.2.2. *All three decision-makers have the same objective; that is a function of the third-level continuous decision variables* $\mathbf{z}$*.*

Assumption 3.2.3. *First-level decision-maker can enforce a specific value (i.e., zero or one) on the second-level decision-maker w.r.t. to a budget (or knapsack)*

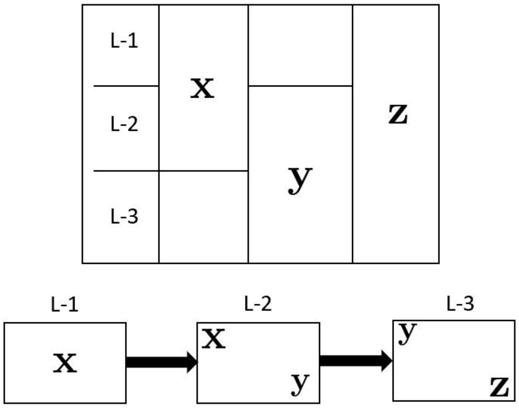


Figure 3.1: Decision Variables shared across levels.

constraint; this, in-turn, implies that the first-level decision variables have the same cardinality as the second-level decision variables.

Assumption 3.2.4. *The third level has a convex objective function defined over a convex set of constraints with continuous decision variables.*

3.2.2 Generic Application Examples

Despite the complexity of MLPs, they are used in many practical applications, due to the inherent hierarchical structure of most decision-making processes, such as forestry (Parkatti et al. 2019), transportation and road planning (Gu et al. 2019), disaster management (Irohara et al. 2013), generation and transmission expansion planning (Hong et al. 2017), supply chain and waste management (Fathollahi-Fard et al. 2018), as well as defence, security and reliability assessment [Mahmoodjanloo et al. (2016), Lin & Bie (2018)]. More specifically, TLP (3.13) can be used in different applications. One of the prominent applications, that we will focus on to demonstrate our solution approaches, is defending critical infrastructure, which is also known as Defender-Attacker-Defender model (Brown et al. 2006). Specific details regarding the application's literature review, modelling and solution algorithms are provided in Section 3.4.

Another application is the utilization of governmental resources; consider a set of new infrastructure projects that needs to be constructed as a part of a government's provincial plan to improve social welfare. The government has complete control over these projects, whether to fully/partly fund, or outsource them to private agencies. If the government chose to use public funds to construct a particular project, the private agencies would only have other ones to bid on; thus the government (leader) has direct control over the private agencies' (follower) choice(s). The government would seek to minimize the overall cost of using that infrastructure on the people/users (or maximize the social welfare) subject to a certain budget of public funds (Level 1), while the private agencies would seek to maximize their profits (Level 2) subject to certain (market/regulatory) constraints, which is reflected as a cost from the users' perspective. The lower-level problem represents the users, as they seek to minimize the incurred costs from using infrastructure to satisfy their needs (Level 3). This application can be viewed as a modified version of what is known in the literature as Build-Operate-Transfer (**BOT**) model Gu et al. (2019). A third generic application is in budgeting or asset management such as maintaining or renovating a specific set of existing infrastructure. The leader has full control of allocating the budget to certain infrastructure (Level 1) to improve the overall utility and decrease operational and maintenance costs, while followers compete to maximize their profits from renovating the selected infrastructure (Level 2). The operator of each basic facility (Level 3) would seek to maximize its utility by minimizing its operational and maintenance cost. A similar hierarchical structure can be applied when universities allocate a budget to fund existing or new research programs to improve its brand and overall rank.

3.2.3 General Analysis

The generic form defined in (3.13), can be reformulated in a matrix form as follows: (**TLP**)

$$
\begin{aligned}
&(\textbf{TLP-L1})\\
&\qquad \min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{z}^*\\
&\qquad \text{s.t.} \quad A_1\mathbf{x} \le \mathbf{b}_1\\
&\qquad \qquad \mathbf{x} \in \{0,1\}^{n_1}\\
&(\textbf{TLP-L2})\\
&\qquad \mathbf{c}^T\mathbf{z}^* = \max_{\mathbf{y}} \quad \mathbf{c}^T\mathbf{z}'\\
&\qquad \qquad \text{s.t.} \quad A_2(\mathbf{x},\mathbf{y})^T \le \mathbf{b}_2\\
&\qquad \qquad \qquad \mathbf{y} \in \{0,1\}^{n_2}\\
&(\textbf{TLP-L3})\\
&\qquad \mathbf{c}^T\mathbf{z}' = \min_{\mathbf{z}} \quad \mathbf{c}^T\mathbf{z}\\
&\qquad \qquad \text{s.t.} \quad A_3(\mathbf{y},\mathbf{z})^T \le \mathbf{b}_3\\
&\qquad \qquad \qquad \mathbf{z} \in \mathbb{R}^{n_3}
\end{aligned}
\tag{3.14}
$$

where $A_1 \in \mathbb{R}^{m_1 \times n_1}, A_2 \in \mathbb{R}^{m_2 \times (n_1+n_2)}, A_3 \in \mathbb{R}^{m_3 \times (n_2+n_3)}, \mathbf{b}_1 \in \mathbb{R}^{m_1}, \mathbf{b}_2 \in \mathbb{R}^{m_2}, \mathbf{b}_3 \in \mathbb{R}^{m_3}, \mathbf{c} \in \mathbb{R}^{n_3}, \mathbf{z}$ is a vector of continuous decision variables of size n_3, $\mathbf{x}$ and $\mathbf{y}$ are both binary decision vectors of size n_1 and n_2, respectively. Since the lower-level problem (**TLP-L3**) is linear and the decision variables are continuous, it is customary to reduce the two levels into a single-level problem. This can be done using two approaches: duality theory and KKT conditions (Arroyo 2010). Depending on how the constraints are formulated and the nature of the objective function, one approach may be computationally superior to the other (Arroyo 2010). Nevertheless, even in BLP, replacing the lower-level problem by its KKT optimality conditions does not necessarily yield a solution for the initial BLP; an equivalence needs to be established even if the lower-level problem is convex. Recent research, by Aussel & Svensson (2019) for the pessimistic case and Dempe & Dutta (2012) for the optimistic case, has discussed the conditions under which an equivalence can be established. It is worth noting that replacing the most lower-level in Problem **TLP** with either the KKT or duality approaches is dependent on satisfying the aforementioned conditions in BLP.

The duality theory approach adds the primal constraints, dual constraints and strong duality constraint of the lower-level problem in the upper-level problem; thus converting it into a single-level. This approach may introduce non-convex bilinear terms; which can be linearized depending on the nature of the terms. The KKT approach involves including the KKT conditions as constraints in the upper-level problem; however this increases the computational burden since the complementary slackness conditions introduce non-linear constraints. Nevertheless, there is a systematic way of linearizing those constraints by introducing binary variables (proportional to the number of primal constraints) by applying the reformulation mentioned in (Fortuny-Amat & McCarl 1981). One must be cautious when applying that reformulation, specifically, in choosing the Big-M value that represents an upper bound for the lower-level dual variables. This value has to be chosen relative to the values that dual variables can hold, in order not to eliminate any feasible solutions which can consequently affect the overall result. The two above mentioned approaches will be used to show that the TLP in (3.14) can be reduced to a BLP without loss of generality.

Dual Reformulation

The form defined in (3.14) can be reduced to a BLP using the duality theory approach (Alguacil et al. 2014) as follows:

(**RTLP-Dual**)

(**RTLP-Dual-L1**)

$$\min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{z}^* \tag{3.15}$$

$$\text{s.t.} \quad A_1\mathbf{x} \leq \mathbf{b}_1 \tag{3.16}$$

$$\mathbf{x} \in \{0,1\}^{n_1} \tag{3.17}$$

(RTLP-Dual-L2)

$$\mathbf{c}^T\mathbf{z}^* = \max_{\mathbf{y},\mathbf{z},\mathbf{u}} \mathbf{c}^T\mathbf{z} \tag{3.18}$$

$$
\begin{aligned}
\text{s.t.} \quad & A_2(\mathbf{x},\mathbf{y})^T \leq \mathbf{b}_2 && (3.19)\\
& A_3(\mathbf{y},\mathbf{z})^T \leq \mathbf{b}_3 && (3.20)\\
& -A_{3\mathbf{z}}^T \mathbf{u} = \mathbf{c}^T && (3.21)\\
& (A_{3\mathbf{y}}\mathbf{y} - b_3)^T \mathbf{u} = \mathbf{c}^T \mathbf{z} && (3.22)\\
& \mathbf{y} \in \{0,1\}^{n_2},\ \mathbf{z} \in \mathbb{R}^{n_3},\ \mathbf{u} \in {\mathbb{R}_+}^{m_3} && (3.23)
\end{aligned}
$$

where $\mathbf{u}$ is the vector of dual variables associated with the third level constraints, $A_{3\mathbf{y}}$ and $A_{3\mathbf{z}}$ are sub-matrices of A_3 consisting of the first n_2 and last n_3 columns, respectively.

KKT Reformulation

By including the KKT conditions of the third level into the second level, the resulting reduced TLP is **RTLP-KKT**. For more information on mathematical programs with complementarity constraints, the interested reader can refer to (Dempe 2002) and (Bard 2013).

(**RTLP-KKT**)

(**RTLP-KKT-L1**)

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{z}^* && (3.24)\\
\text{s.t.} \quad & A_1 \mathbf{x} \leq \mathbf{b}_1 && (3.25)\\
& \mathbf{x} \in \{0,1\}^{n_1} && (3.26)
\end{aligned}
$$

(RTLP-KKT-L2)

$$\mathbf{c}^T\mathbf{z}^* = \max_{\mathbf{y},\mathbf{z},\mathbf{u}} \mathbf{c}^T\mathbf{z} \tag{3.27}$$

$$\text{s.t.} \quad A_2(\mathbf{x},\mathbf{y})^T \leq \mathbf{b}_2 \tag{3.28}$$

$$A_3(\mathbf{y},\mathbf{z})^T \leq \mathbf{b}_3 \tag{3.29}$$

$$\mathbf{c}^T + \mathbf{u}^T A_{3\mathbf{z}} = 0 \tag{3.30}$$

$$\mathbf{u}^T(A_3(\mathbf{y},\mathbf{z})^T - \mathbf{b}_3) = 0 \tag{3.31}$$

$$\mathbf{y} \in \{0,1\}^{n_2},\ \mathbf{z} \in \mathbb{R}^{n_3},\ \mathbf{u} \in \mathbb{R_+}^{m_3} \tag{3.32}$$

It is worth mentioning that constraints (3.22) and (3.31)can be linearized using traditional methods (e.g., see (Fortuny-Amat & McCarl 1981),(Floudas 1995)). By examining the two reformulations: **RTLP-Dual** and **RTLP-KKT**, the following optimality and feasibility results can be obtained. Details on how these results were obtained are provided in Appendix A.

- *Problems **RTLP-Dual** and **RTLP-KKT** are equivalent, consequently if* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*, \mathbf{u}^*)$ *is optimal for **RTLP-Dual**, then it has to be optimal for **RTLP-KKT**.*

- *If* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*, \mathbf{u}^*)$ *is optimal for **RTLP-Dual** or **RTLP-KKT**, then* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in OP$ *is also optimal solution for Problem **TLP**.*

- *If* $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{z}}, \overline{\mathbf{u}})$ *is feasible for **RTLP-Dual** or **RTLP-KKT**, then* $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{z}}) \in S$, $(\overline{\mathbf{y}}, \overline{\mathbf{z}}) \in S_{\mathbf{y},\mathbf{z}}(\mathbf{x})$ *and* $\overline{\mathbf{z}} \in M_{\mathbf{z}}(\mathbf{x},\mathbf{y})$ *for Problem **TLP**.*

3.3 Solution Approach

In this section, we present the three different heuristic approaches to solve problem **TLP**. It should be noted that we are going to refer to problems **RTLP-Dual** and **RTLP-KKT** as **RTLP**, since both must yield the same solutions. Without loss of generality, and for the sake of simplicity and presentation, we will assume that $m_1 = 1$, $n_1 = n_2$ and $m_2 = n_1 + 1$, where the set of constraints (3.28) can be represented as follows:

$$\mathbf{y} \leq \mathbf{x} \tag{3.33}$$

$$A_2^{'}\mathbf{y} \leq b_2 \tag{3.34}$$

where $A_2^{'} \in \mathbb{R}^{1\times n_2}$. In Section 3.3.4 we discuss how the proposed heuristics can be extended to handle multi-dimensional budget constraints (i.e., $m_1 > 1$ and $m_2 > n_1 + 1$).

The proposed heuristics are used to rank the binary variables in a pre-planning stage in a **TLP** (i.e., not just for a particular instance); this should provide fast optimal/near-optimal solutions with less computational effort. Additionally, they offer a trade-off between solution quality and computational effort, depending on the application. In particular, the *LPRank* heuristic achieves computational efficiency by solving a series of linear programs instead of having to solve a bi-level mixed-binary linear program, thereby reducing the time required to solve a particular instance. In contrast, *HybridRank* triggers both linear and mixed-binary linear program solvers, depending on the change of objective function value through iterations. *MBLPRank* only invokes a mixed-binary linear program solver, thereby yielding a better quality and taking relatively more time per instance. Moreover, the heuristics can be customized to provide a tie-breaking rule between binary decision variables, depending on the application under consideration. This criterion can be used to resolve degenerate solutions that occur in multi-level programs.

3.3.1 LPRank Heuristic

The pseudo-code for this approach is provided in Heuristic.3.1. We begin by reading the problem's parameters for each level. Step 3 initializes an upper bound for **TLP**. Steps 4 to 5 develop an upper bound for **TLP** by solving **RTLP-L2** as a mixed-binary linear program (MBLP) without constraint (3.33), i.e., removing all of the first-level decision variables, and relaxing the right hand side of (3.34), thus obtaining an upper bound for **TLP**. It should be noted that solving **RTLP-L2** as MBLP could be costly depending on the number of decision variables and how the binary variables are integrated in the constraints. However, this computationally expensive operation can be avoided by exploiting the specific problem structure; this will be further explained and demonstrated in the next sections.

The main objective of Steps 7 to 21 is to rank the impact of each binary variable ($y_i \in \mathbf{y}$) through measuring its effect on the objevtive value of **RTLP**, *obj.val.*, by incrementing the allowable *Budget* b_2 in the right hand side of constraint (3.34)). First, a stopping criteria is established to break out of the for loop, whenever the objective value hits the stored upper bound. Second, for each additional unit of *Budget*, the model **RTLP-L2** is solved as a linear program (LP) by activating (setting it to 0 or 1) one binary variable (y_i) at a time and storing the value of *obj.val.* and the corresponding index i. These indices will be then arranged in a descending order according to their effect on *obj.val.*. The binary variable with the highest effect will be fixed for the next iteration. In the next iteration, *Budget* will be incremented by one unit, a check if the objective has hit the upper bound will be done, and then **RTLP-L2** will be solved as an LP for each binary variable that is not activated/fixed. Each binary variable will be then ranked according to its effect on *obj.val.*. This continues until we reach *Desired Budget*. In other words, each time the *Budget* is incremented by one unit, the number of times we solve **RTLP-L2** as an LP is decreased by one. For example, if we have one unit of *Budget* available and the number of binary variables is n_1, **RTLP-L2** will be solved n_1 times as LP. After ranking each binary variable, one will be chosen and fixed/activated. In the next iteration, after incrementing the *Budget*, **RTLP-L2** will be solved $n_1 - 1$ times.

If two or more binary variables have the same effect on *obj.val.* for a certain *Budget* value, we break the tie by decrementing the *Budget* by one unit and solve **RTLP-L2** as an LP. We then sort the binary variables with the highest effect on *obj.val.* with respect to an operational preference that has a direct effect on *obj.val.*. This operational preference is application-specific and will be further demonstrated in the next sections. The binary variable with the highest effect on both *obj.val.* and the operational preference will be activated for the next iteration.

The upper-level decision variables ($x_i \in \mathbf{x}$) will be chosen to be activated according to two factors: first, the allocated budget for the first level (i.e., b_1 in constraint (3.16)), which controls how many binary variables can be activated by the first level

Algorithm 3.1 LP Ranking

```
 1: procedure LPRank
 2:     Read problem data: $A_1, b_1, \mathbf{c}, A_2, b_2, A_3, b_3$
 3:     UB $\leftarrow \infty$.
 4:     Solve RTLP-L2 model as MBLP without any control from upper-level (i.e.,
        remove constraint (3.33)) and relaxed Budget ($b_2 = n_1$).
 5:     UB $\leftarrow$ obj. val..
 6:     Initialize Repository.
 7:     for $j = 1 :$ Desired Budget do
 8:         if obj. val.$(j)$ = UB then
 9:             Break.
10:         end if
11:         for $i = 1 : n_1$ do
12:             if $y_i \notin$ Repository then
13:                 Set $y_i \in \mathbf{y} \leftarrow 0$.
14:                 Solve RTLP-L2 model without constraint 3.33 as an LP by setting
                    $b_2 \leftarrow j$.
15:                 Store obj.val. and the corresponding $y_i$'s index.
16:             end if
17:         end for
18:         Sort $y_i$'s indices in a descending order according to obj.val.
19:         if Two or more binary variables ($y_i$) have the same highest obj.val. then
20:             $b_2 \leftarrow j - 1$.
21:             Solve RTLP-L2 model as LP without constraint 3.33.
22:             Sort $y_i$ indices in a descending order w.r.t an operational preference
                (application-oriented).
23:             Set $y_i \leftarrow 0$ for $i$ (index) with the highest obj.val. and operational
                effect.
24:         else Set $y_i \leftarrow 0$ for $i$ (index) with the highest obj.val..
25:         end if
26:         Store $y_i$ index, $j$ (Budget value), obj.val. in Repository.
27:     end for
28:     Set $x_i \in \mathbf{x}$ w.r.t sorted list and available budget.
29:     Repeat steps 7 to 21 while enforcing constraint 3.33 after $\mathbf{x}$ has been revealed
        to determine $\mathbf{y}$.
30: end procedure
```

decision-maker. Second, in order to suppress the effect of the second level decision-maker, the leader would choose to activate/control the binary variables with the highest effect on *obj.val.*, which are stored in the repository for each incremental budget unit.

In order to determine the second level decision variables, steps 7 to 21 are repeated while enforcing constraint (3.33). In other words, repeating those steps would determine the binary variables with the highest effect on *obj.val.* after the first level decision variables have been revealed. A detailed flowchart for *LPRank* is depicted in Figure B.1 in Appendix B. In Proposition 3.3.1 we show that the LPRank heuristic finds feasible solutions to **TLP** (3.14) as long as the input instance for **TLP** has feasible solutions.

Proposition 3.3.1. *Assuming that **TLP** admits feasible solutions and an equivalence exists between **TLP** and **RTLP**, the LPRank heuristic will yield a solution for **TLP** that is feasible for the tri-level constraint region S.*

Proof. Applying *LPRank* heuristic first solves **RTLP-L2** as an LP with successive increments of *Budget* value through systematically activating binary variables of the second-level decision-maker (i.e., enforcing a zero/one value), until the desired value is reached. Thus constraint (3.34) is satisfied. Moreover, the obtained solution satisfies **RTLP-L2** except for constraint set (3.33). Step 22 ensures that the first level decision variables (i.e., $x_i \in \mathbf{x}$) satisfy constraint (3.16) by activating first-level decision variables (i.e., enforcing a zero/one value) w.r.t. the sorted list. Finally, Step 23 enforces constraint (3.33) and resolves **RTLP-L2** as an LP with successive increments of *Budget* value, thus ensuring that the obtained solution satisfies the constraint region of **RTLP**. Since **TLP** is equivalent to **RTLP**, we conclude that the same solution should also be feasible for the tri-level constraint region S of **TLP**. □

It is worth mentioning that **TLP-L3** has no direct interaction with first-level decision variables, $\mathbf{x}$, hence finding a solution $\mathbf{z}^*$ that belongs to the **basic rational reaction set of the third-level**, $M_{\mathbf{z}}(\mathbf{x}, \mathbf{y})$, is guaranteed for fixed $(\mathbf{x}, \mathbf{y}) \in X \times Y$. Additionally, the last two steps in the heuristic guarantee the feasibility of the first-level budget constraint (3.16) and leader-follower constraints (3.33), obtaining a solution that is feasible for the second-level decision-maker and the tri-level constraint region S.

The *LPRank* heuristic enables us to solve **RTLP** as an LP and find feasible near-optimal/optimal solutions in an efficient manner rather than solving it as a bi-level MBLP for each budget value b_1 and b_2 in constraints (3.16) and (3.34), respectively. However, *LPRank* is limited by its inability to consider the simultaneous effect of activating several decision variables, which is the case if solved as an MBLP. Moreover, in applications where *obj.val.* has a monotonic decreasing (or increasing) pattern, *LPRank* can get stuck with the same *obj.val.* for several iterations despite incrementing the *Budget* value. This is due to the following factors: (1) In each iteration, *LPRank* chooses to activate the binary variable with the highest effect on *obj.val.* (or highest effect on *obj.val.RTLP* and an operational preference). Consequently, for the next iterations (i.e., upon incrementing the *Budget* value), the previously activated binary variables are fixed. Hence, *LPRank* might get anchored due to its inability to quantify the effect of activating several binary variables at the same time.

(2) *LPRank* can yield the same *obj.val.* due to the monotonic pattern established by the nature of the application under consideration. This means that even if we solved **RTLP-L2** as an MBLP, it will yield the same *obj.val.* despite the incremental increase of the *Budget* value.

3.3.2 HybridRank Heuristic

The pseudo-code for this approach is given in Heuristic 3.2. This version is specifically designed to remedy the problems mentioned in *LPRank* by solving **RTLP-L2** as an LP in the exact same way previously explained except for a certain condition. If *obj.val.* is the same for two successive iterations, it signals that LP ranking could be anchored (might be due to the nature of the application) after the incremental budget increase. Thus **RTLP-L2** is solved as MBLP to consider the simultaneous effect of activating several binary variables at the same time. Binary variables in this approach are ranked based on solving **RTLP-L2** as LP or MBLP, making it a *HybridRank* heuristic.

After reaching the desired *Budget* value, Step 25 determines the unique binary variables indices for each value. A list is made determining the count of each index, which are subsequently sorted in a descending order in Steps 26 and 27. This list quantifies the frequency by which each binary variable is activated for different budget values. Step 28 then sets/activates the upper-level decision variables ($x_i \in \mathbf{x}$) depending on the available budget in constraint (3.16) and to suppress the effect of the second level decision-maker by controlling the binary variables with the highest number of counts in the sorted list. Step 29 repeats steps 7 to 24 by enforcing constraint (3.33) to determine the binary variables with highest effect ($y_i \in \mathbf{y}$) on *obj.val.* after the first-level decision variables have been revealed.

HybridRank heuristic overcomes the drawback of *LPRank* by its ability to consider the effect of activating several binary variables simultaneously through solving **RTLP-L2** as a MBLP. Hence, if *obj.val.* is the same after incrementing the *Budget* value, then this is due to the nature of the application under consideration. This comes at a computational cost of possibly solving several MBLPs, depending on the application and the effect of activating the binary decision variables in **RTLP-L2**.

It is important to note that *LPRank* might yield the same solution quality with less computational effort as *HybridRank*, if *obj.val.* varies with activating different binary variables, making the advantage of *HybridRank* over *LPRank* is application-specific; this will be evident in the numerical results detailed in Subsection 3.4.5. A detailed flowchart outlining different procedures of the *HybridRank* heuristic is depicted in Figure B.2 in Appendix B.

Proposition 3.3.2 compares the running times between LPRank and HybridRank.

Proposition 3.3.2. *The running time of HybridRank is bounded below by the running time of LPRank.*

Proof. Consider the best running time situation in *HybridRank*. This occurs when solving **RTLP-L2** yields different *obj.val.RTLP* values for every incremental unit increase in the *Budget* value. Hence, the *HybridRank* heuristic will have the same running time as that of *LPRank*. Otherwise, if the *obj.val.RTLP* is the same for two successive iterations, **RTLP-L2** will be solved as MBLP and so increasing the computational burden for *HybridRank*. □

Algorithm 3.2 LP-MBLP Ranking

```
 1: procedure HybridRank
 2:     Read problem data: $A_1, b_1, \mathbf{c}, A_2, b_2, A_3, b_3$
 3:     Set UB $\leftarrow \infty$.
 4:     Solve RTLP-L2 model as MBLP without any control from upper-level (i.e remove constraint 3.33) and relaxed budget ($b_2 = n_1$).
 5:     UB $\leftarrow$ obj. val..
 6:     Initialize Repository.
 7:     for $j = 1 :$ Desired Budget do
 8:         if obj.val.($j$) = UB then
 9:             Break.
10:         end if
11:         for $i = 1 : n_1$ do
12:             if $y_i \notin$ Repository then
13:                 Set $y_i \in \mathbf{y} \leftarrow 0$.
14:                 Solve RTLP-L2 model without constraint 3.33 as LP by setting $b_2 \leftarrow j$.
15:                 Store obj.val. and the corresponding $y_i$'s index.
16:             end if
17:         end for
18:         Sort $y_i$'s indices in a descending order according to obj.val.
19:         if Two or more binary variables ($y_i$) have the same highest obj.val. then
20:             $b_2 \leftarrow j - 1$.
21:             Solve RTLP-L2 model as LP without constraint 3.33.
22:             Sort $y_i$'s index in a descending order w.r.t an operational preference (application-oriented).
23:             Set $y_i \leftarrow 0$ for $i$ (index) with the highest obj.val. and operational effect.
24:         else Set $y_i \leftarrow 0$ for $i$ (index) with the highest obj.val..
25:         end if
26:         Store $y_i$ index, Problem type: LP, $j$ (Budget value), obj.val. in Repository.
27:         if obj.val.($j$) = obj.val.($j - 1$) then
28:             Solve RTLP-L2 model without constraint 3.33 as MBLP.
29:         end if
30:         Update $y_i$ indices, Problem type: MBLP, $j$ (Budget value), obj.val. in Repository.
31:     end for
32:     Determine unique $y_i$ indices in Repository for all values of $j$ (i.e., from $j = 1 :$ Budget ).
33:     Count each unique index in Repository.
34:     Sort indices according to their count in a descending order.
35:     Set $x_i \in \mathbf{x}$ w.r.t sorted list and available budget.
36:     Repeat steps 7 to 24 with enforcing constraint 3.33 after $\mathbf{x}$ has been revealed to determine $\mathbf{y}$.
37: end procedure
```

3.3.3 *MBLPRank* Heuristic

Algorithm 3.3 MBLP Ranking

```
1:  procedure MBLPRank
2:      Read problem data: $A_1, b_1, \mathbf{c}, A_2, b_2, A_3, b_3$
3:      UB $\leftarrow \infty$.
4:      Solve RTLP-L2 model as an MBLP without any control from the upper-level
        (i.e., remove constraint 3.33) and relaxed budget ($b_2 = n_1$).
5:      UB $\leftarrow$ obj. val..
6:      Initialize Repository.
7:      for $j = 1$ : Desired Budget do
8:          if obj.val.($j$) = UB then
9:              Break.
10:         end if
11:         Solve RTLP-L2 model without constraint 3.33 as MBLP by setting $b_2$
            $\leftarrow j$.
12:         Store $y_i$ indices, $j$, obj.val. in Repository.
13:     end for
14:     Determine unique $y_i$ indices in Repository for all values of $j$ (i.e., from $j =$
        $1$ : Budget ).
15:     Count each unique index in Repository.
16:     Sort indices according to their count in a descending order.
17:     Set $x_i \in \mathbf{x}$ w.r.t. sorted list and available budget.
18:     Solve RTLP-L2 model with enforcing constraint 3.33 as an MBLP after $\mathbf{x}$
        has been revealed to determine $\mathbf{y}$.
19: end procedure
```

The pseudo-code for this approach is provided in Heuristic 3.3. *MBLPRank* focuses more on solution quality rather than computational time. It is designed for MBLPs with a modest number of integer variables, or if the computational burden of an MBLP can be reduced. First, *MBLPRank* starts by reading the problem parameters and finding an upper bound in Steps from 1 to 5. The repository is then initialized in Step 6. Steps 7 to 11, solve **RTLP-L2** as MBLP for each *Budget* value, with a specific condition to break out of the loop if the upper bound is hit.

Steps 12 to 14 determine the unique binary variables indices for each *Budget* value, after which a list is made with the count of each index. The indices are then sorted in a descending order. Step 15 then sets/activates the upper-level decision variables ($x_i \in \mathbf{x}$) to suppress the effect of the second level decision-maker by controlling the binary variables with the highest number of counts in the sorted list, depending on the available budget in constraint (3.16). Step 16 solves **RTLP-L2** as MBLP by enforcing constraint (3.33) to determine the second level decision variables ($y_i \in \mathbf{y}$). Flowchart for *MBLPRank* is provided in Figure B.3 in Appendix B.

3.3.4 Multi-dimensional Knapsack Constraints

While it is true that most applications of TLPs involve a single budget constraint (e.g., see (Sarhadi et al. 2015), (Alvarez 2004), and (Scaparra & Church 2008)), there may be occasions where a need arises for multiple budget constraints. Here we extend the proposed heuristics to handle multi-dimensional knapsack constraints. One way of dealing with multi-dimensionality in the second-level is to solve for each budget constraint and rank the binary variables according to their effect on the objective value, *obj.val.*, where each budget constraint would have a *Repository* holding the count of unique index y_i sorted in a descending order. It is worth mentioning that the first and second levels should have the same number of budget constraints (i.e., size($\mathbf{b_1}$)= size($\mathbf{b_2}$). *Repositories* for all budget constraints are combined and sorted in a descending order. The binary variable with the highest rank is activated and then a feasibility check is done on all budget constraints. If it is feasible, we activate the next binary variable in the *Repository* list until we reach infeasibility for the budget constraints in the second-level. In other words, the binary variables for the second-level are activated in a greedy manner w.r.t. all budget constraints. According to the sorted list of the second-level, the first-level decision-maker would activate the first-level decision variables, **x**), in a greedy manner w.r.t. the first-level budget constraints until infeasibility is achieved. A pseudo code (Heuristic C.1) and a flow chart (Figure C.1) are included in Appendix B. The aforementioned approach can be applied, in a similar way, to the rest of the proposed heuristics in this paper.

3.4 Defending Critical Infrastructure: Application on Electrical Power Transmission Networks

BLP and TLP have been used extensively in determining critical infrastructure links/element for various types of networks such as electrical transmission, transportation and supply chain networks [Brown et al. (2006), Babick (2009), Arroyo (2010)]. Specifically, BLP and TLP are used for modelling attacker-defender (AD) problems and defender-attacker-defender models (DAD) or, in some literature, defender-attacker-operator (DAO). Network flow problems are a typical application of DAD models such as minimum cost flow (Babick 2009), constrained shortest path problems (Lazzaro 2016) or minimizing the cost of load shedding in an electric power transmission network (Wu & Conejo 2017).

The DAD model is a form of a three-player sequential Stackelberg game. First, the defender plans on fortifying the system by hardening the critical infrastructure, or in some other cases designing new additions to improve the latter's overall resilience. Second, an attacker seeks to inflict as much damage as possible to degrade infrastructure's functionality despite the employed protective plans. Third (and final stage of the model) includes the system operator (or can be seen as the defender who is taking another sequential move after the attacker) minimizing the inflicted damage done by the attacker through maintaining the operational constraints of the attacked infrastructure.

Electric grid security has been recently becoming a major concern for governments, network planners and operators, due to the dependency/interdependency of other critical infrastructures such as communications, transportation, water systems, healthcare and public health sectors that pose a grave risk. A blow to the electric grid network could cause cascading effects on other sectors, leading to a massive catastrophe that spirals out of control. Specifically, targeted attacks on the power system components highly jeopardize the overall stability of the power grids. The transmission system of the power grid represents a weak-unprotected part, allowing it to be easily targeted by those malicious attacks. In April 2013, an attack on a substation, in California, resulted in the damage of 17 transformers. The cost of the load shed was substantial, as it took 27 days to repair them and bring them back to service (Smith 2014). In March 2019, a "first-of-its-kind" power grid attack occurred in the western united states (Sussman 2019), where internet-facing firewalls were rebooting. Consequently, each reboot cut-off communication between generation sites and control centre. The power grid cyberattack continued for approximately 10 hours. This prompted the U.S. government to announce a surprising move by introducing the Securing Energy Infrastructure Act (SEIA), to secure power grids by using “retro” technologies (O’Flaherty 2019). The main objective of this bill is to thwart cyber-adversaries, by replacing automated systems with low-tech redundancies, like manual procedures controlled by human operators making cyberattacks much more strenuous. Attacking the critical components of the power grid may cause cascading outages and possibly a complete blackout. Consequently, identifying the critical components of the grid which represents a high potential target for terrorist physical/cyber attacks, is crucial for its safe operation and of equal importance to following adequate protection plans.

TLP in the context of defending electrical power grids has received lots of attention from researchers in the past few years. The DAD model within the context of defending electrical power grids has the same generic structure in 3.13. The first level represents the defender/planner who is trying to minimize load shedding in the network, and the decision variables are binary subject to linear budgetary constraint representing the defence resources. The second level is the attacker who is in turn trying to inflict the maximum possible damage to the network by attacking the most critical lines. The decision variables in this level are also binary, while the constraints are linear and involve the decision variables of both the first and second levels. The third level represents the operator model, who is trying to maintain the operational constraints of the network and to minimize the inflicted damage. The decision variables in this level are continuous, while the objective function and constraints are linear. However, the constraints are a function of the second level and third level decision variables.

As previously mentioned, the generic form stated in 3.13 can be reduced to two levels using either the duality theory approach 3.15 or KKT conditions 3.24. The resulting problem is BLP, where the first level represents the planner's decisions, while the second level now represents AD model fused into a single-level.

The following sections will be dedicated to present the DAD model in the context of electric power grid security, propose a modified enumeration algorithm that provides optimal solutions and compare it to our proposed solution approaches on three electrical transmission networks of different sizes.

3.4.1 Defender-Attacker-Defender Model for Electric Power Grid Security

Table 3.1: Mathematical Notations for DAD model

Indices and Sets	
Symbol	Description
J	Set of generators.
J_n	Set of generators connected to bus n.
L	Set of transmission lines.
N	Set of buses.
j	Generator index.
l	Transmission line index.
n	Bus index.
Paramters	
A_{nl}	Element of the incidence matrix equals 1 if bus n is the sending end of line l, -1 if bus n is the receiving end of line l, and 0 otherwise.
B_l	Imaginary part of admittance of line l.
K	Attack Budget (Number of Lines).
D	Defence Budget (Number of Lines).
P_n^d	Demand at bus n.
$\bar{P}_l^f$	Maximum power flow in line l.
$\bar{P}_j^g$	Maximum power a generator can produce.
$\underline{P}_j^g$	Minimum power a generator can produce.
$\bar{\delta}$	Maximum power angle for a bus.
$\underline{\delta}$	Minimum power angle for a bus.
Decision Variables	
v_l	Binary variable set to 0 if line l is attacked and 1 otherwise.
z_l	Binary variable set to 1 if line l is defended and 0 otherwise.
P_j^g	Output power from generator j.
P_l^f	Power flow in line l.
δ_n	Power angle for bus n.
Φ_n	Load shed at bus n.

We will use the notation in Table 3.1 to model the DAD problem for electric grid security. We have opted for using the common notation in this field so that it will be easier for the reader to compare our model with existing studies in this area.

The DAD model can be formulated as follows:

(**DAD**)

(**DAD-L1**)

$$\min_{\mathbf{z}} \quad \sum_{n \in N} \Phi_n^* \tag{3.35}$$

$$\text{s.t.} \quad \sum_{l \in L} z_l \leq D \tag{3.36}$$

$$z_l \in \{0,1\}, \ \forall \, l \in L \tag{3.37}$$

(**DAD-L2**)

$$\max_{\mathbf{v}} \quad \sum_{n \in N} \Phi_n^{'} = \sum_{n \in N} \Phi_n^* \tag{3.38}$$

$$\text{s.t.} \quad \sum_{l \in L} (1 - v_l) \leq K \tag{3.39}$$

$$z_l \leq v_l, \quad \forall \, l \in L \tag{3.40}$$

$$v_l \in \{0,1\}\,, \quad \forall \, l \in L \tag{3.41}$$

(**DAD-L3**)

$$\sum_{n \in N} \Phi_n^{'} \in \left\{ \arg\min_{\boldsymbol{\delta}, \mathbf{P^g}, \mathbf{P^f}, \boldsymbol{\Phi}} \sum_{n \in N} \Phi_n \right\} \tag{3.42}$$

$$\text{s.t.} \quad P^f_l = B_l v_l \sum_{n \in N} A_{nl} \delta_n, \quad \forall\, l \in L \quad : (\mu_l) \tag{3.43}$$

$$\sum_{j \in J_n} P^g_j - \sum_{l \in L} A_{nl} P^f_l + \Phi_n = P^d_n, \quad \forall\, n \in N : (\lambda_n) \tag{3.44}$$

$$-\bar{P}^f_l \leq P^f_l \leq \bar{P}^f_l, \quad \forall\, l \in L : (\underline{\omega}_l, \bar{\omega}_l) \tag{3.45}$$

$$-\bar{\delta} \leq \delta_n \leq \bar{\delta}, \quad \forall\, n \in N \left(\underline{\gamma}_n, \bar{\gamma}_n\right) \tag{3.46}$$

$$\underline{P}^g_j \leq P^g_j \leq \bar{P}^g_j, \quad \forall\, j \in J : \left(\underline{\theta}_j, \bar{\theta}_j\right) \tag{3.47}$$

$$0 \leq \Phi_n \leq P^d_n, \quad \forall\, n \in N : (\underline{\alpha}_n, \bar{\alpha}_n) \tag{3.48}$$

The variables in parenthesis from (3.43) to (3.48) are the dual variables; each defined with respect to the corresponding constraint and grouped in Table 3.2. The upper-level is associated with planner/defender, middle-level pertains to the attacker's move, and the lower-level is concerned with the reaction of network operator to minimize load shedding based on the attacker's decision. The planner's objective is to minimize the load shed represented by (3.35), or in other words increase the robustness of the overall network. Constraint (3.36) is the defence budget in terms of transmission lines that can be protected, where z_l is a binary variable set to 1 if line l is defended and 0 otherwise. Constraints (3.37) enforce the binary nature of z_l. Equation (3.38) maximizes the total load shed that results from attacking line l, where v_l is a binary variable set to 0 if line l is interdicted and 1 otherwise. The budgetary constraint for the attacker is modeled by (3.39), where K represents the total resources available (in terms of transmission lines), while constraints (3.41) are the binary constraints for the middle-level decision variables. Constraints (3.40) enforces the assumption that if a line l is defended, then it cannot be attacked. In other words, if z_l is 1 then v_l is must also be 1. The lower-level problem models the network operator's reaction to minimize the load shed based on the attack scenario represented in (3.42). The dc-power flow for each line is modeled in (3.43), whereas the power balance equations in each bus is represented by (3.44) (or node balance equations in other contexts). Constraints (3.45)-(3.48) are the upper and lower bounds for the lower-level decision variables. Constraints (3.48) ensure that the load shed in each consumer sector does not exceed the load at that electric bus.

Table 3.2: Mathematical Notations for the Dual of the Operator model

λ_n	Variable associated with power balance constraint at bus n.
μ_l	Variable associated with the power flow constraint at line l.
$\underline{\omega}_l$	Variable associated with lower bound of power flow at line l.
$\bar{\omega}_l$	Variable associated with upper bound of power flow at line l.
$\underline{\gamma}_n$	Variable associated with lower bound of power angle at bus n.
$\bar{\gamma}_n$	Variable associated with upper bound of power angle at bus n.
$\bar{\omega}_l$	Variable associated with upper bound of power flow at line l.
$\underline{\theta}_j$	Variable associated with lower bound of generator j.
$\bar{\theta}_j$	Variable associated with upper bound of generator j.
$\underline{\alpha}_n$	Variable associated with lower bound of load shed at bus n.
$\bar{\alpha}_n$	Variable associated with upper bound of load shed at bus n.

3.4.2 Equivalence between TLP and RTLP

In order to be able to reduce the original TLP, we need to establish equivalence to the reduced version of the problem (i.e., RTLP). We will direct our analysis to the Defender-Attacker-Defender (**DAD**) model, as equivalence can differ from an application to another. First, we will establish the conditions under which the solution of lower-level problem of DAD is unique. Hence, applying KKT or duality approaches is necessary and sufficient to guarantee the optimality of the lower-level problem (i.e., defender or operator problem), provided that it has a convex objective function defined over a convex set of constraints. Second, we make use of two assumptions of the class of TLP being studied; first, we have a common objective functions shared across all levels with continuous decision variables **z**, and second the first-level decision maker has a direct control over the second-level decision variables through a budget/limit, which implies both levels have the same number of decision variables. Consider the third-level of Problem **DAD**, then for a given vector of attack variables (**v**), the defender's/operator's problem becomes:

(**DAD-L3**)

$$\min_{\boldsymbol{\delta},\mathbf{P^g},\mathbf{P^f},\boldsymbol{\Phi}} \quad \sum_{n\in N} \Phi_n$$

$$\text{s.t.} \quad P_l^f = B_l v_l^* \sum_{n\in N} A_{nl}\delta_n, \quad \forall\, l \in L \quad : (\mu_l) \tag{3.49}$$

$$\sum_{j\in J_n} P_j^g - \sum_{l\in L} A_{nl} P_l^f + \Phi_n = P_n^d, \quad \forall\, n \in N : (\lambda_n) \tag{3.50}$$

$$-\bar{P}_l^f \le P_l^f \le \bar{P}_l^f, \quad \forall\, l \in L : (\underline{\omega}_l, \bar{\omega}_l) \tag{3.51}$$

$$-\bar{\delta} \leq \delta_n \leq \bar{\delta}, \quad \forall\, n \in N \left(\underline{\gamma}_n, \bar{\gamma_n}\right) \tag{3.52}$$

$$0 \leq P^g_j \leq \bar{P}^g_j, \quad \forall\, j \in J : \left(\bar{\theta}_j\right) \tag{3.53}$$

$$0 \leq \Phi_n \leq P^d_n, \quad \forall\, n \in N : (\underline{\alpha}_n, \bar{\alpha_n}) \tag{3.54}$$

Using the work of Ríos-Mercado et al. (2002) on gas networks, and Krebs et al. (2018) on DC power flow networks, we show the uniqueness of solution problem (**DAD-L3**), based on the following assumption for a given attack vector ($\mathbf{v}$).

Assumption 3.4.1. *Generator productions* ($\mathbf{P^g}$) *and load sheds* ($\boldsymbol{\Phi}$) *satisfy* (3.53),(3.54) *and*
$\sum_{n\in N}\left(\sum_{j\in J_n} P^g_j + \Phi_n - P^d_n\right) = 0$. *Moreover, the phase angle* δ_r *at an arbitrary node* $r \in N$ *is fixed.*

It is worth mentioning that these assumptions do not add any extra constraints to (**DAD-L3**) except for setting a reference node angle. By summing (3.50) for all nodes, we obtain $\sum_{n\in N}\left(\sum_{j\in J_n} P^g_j + \Phi_n - P^d_n\right) = 0$. Hence using Theorem 2 of Ríos-Mercado et al. (2002), the following result can be directly applied.

Theorem 3.4.1. *Let* $r \in N$ *be a reference node with a constant voltage angle* δ_r*, and Assumption 3.4.1 holds, then if a solution* ($\mathbf{P^f}, \boldsymbol{\delta}$) *of system* (3.49)-(3.52) *exists, it is unique for a given attack vector* ($\mathbf{v}$)*.*

Furthermore, we can restrict ourselves to study the conditions under which the uniqueness of the entire solution of Problem (**DAD-L3**) for a given attack vector ($\mathbf{v}$) through the following Theorem.

Theorem 3.4.2. *Let generator productions* ($\mathbf{P^g}$) *and load sheds* ($\boldsymbol{\Phi}$) *in Problem **(DAD-L3)** be unique for a given attack vector* ($\mathbf{v}$) *and $r \in N$ be a reference node with a constant voltage angle δ_r. Then, the entire solution of Problem **(DAD-L3)** is unique.*

Proof. Assumption 3.4.1 holds as generator productions ($\mathbf{P^g}$) and load sheds ($\boldsymbol{\Phi}$) are unique for Problem (**DAD-L3**). The solution ($\mathbf{P^f}, \boldsymbol{\delta}$) of system (3.49)-(3.52) corresponds to the unique generator production ($\mathbf{P^g}$) and load shed ($\boldsymbol{\Phi}$) vectors, such that ($\mathbf{P^g}, \boldsymbol{\Phi}, \mathbf{P^f}, \boldsymbol{\delta}$) is a solution of of Problem (**DAD-L3**). Moreover, the existence of a solution is trivial because load sheds $\boldsymbol{\Phi}$ can account for disrupted demand for any given attack vector ($\mathbf{v}$), and the problem is bounded. Hence, applying Theorem 3.4.1 yields unique flows ($\mathbf{P^f}$) and phase angles ($\boldsymbol{\delta}$) with respect to node r. Thus, the solution of Problem (**DAD-L3**) is unique. □

We have proved partial uniqueness of solution for Problem (**DAD-L3**) using Theorem 3.4.1 and conditions under which the entire solution of Problem (**DAD-L3**) using Theorem 3.4.2 is unique. Consequently, applying KKT approach to reduce the TLP (**DAD**) will yield an equivalent bi-level model, as a direct result of Theorem 3.4.1 and Theorem 3.4.2 on Problem (**DAD-L3**), which has a convex objective function defined over a convex set of constraints. Hence, applying KKT approach or equivalently the duality approach will result in a unique set of continuous decision variables. Thus, no selection approach (i.e. pessimistic or optimistic) is required for the second-level.

Furthermore, the class of TLP under study has two main assumptions:

- A common objective function shared across all levels with continuous decision variables $\mathbf{z}$.
- The first-level decision maker has a direct control over the second-level binary decision variables through a budget/limit. This implies that both levels have the same number of binary decision variables.

In case of multiple optima in the second-level (i.e., the attacker problem), selecting any of the degenerate solutions will not affect the objective function, either from the attacker's or the defender's perspective, as objective function is shared across all levels. Furthermore, any of the attackers' choices will not cause infeasibility to the defender's problem (i.e., first level) for two reasons: first, second-level decision variables have no direct interaction with first-level objective function and constraints, and second, first-level decision maker has a direct control over second-level decision variables. Hence, even if a selection approach is determined (e.g., pessimistic), it will not affect the first-level problem.

3.4.3 Single-Level Attacker-Defender Model

In this section our goal is to reduce the AD model to a single level problem.

Dual of the Operator Model

First we convert the primal lower-level problem representing the operator's/defender's model to its dual counterpart, in order to merge the AD model into a single-level.

The dual problem of the lower-level DAD model is formulated as follows:

$$\begin{aligned}
\max_{\substack{\mu_l,\lambda_n,\underline{\gamma}_n,\bar{\gamma}_n,\underline{\theta}_j,\bar{\theta}_j,\\ \underline{\omega}_l,\bar{\omega}_l,\underline{\alpha}_n,\bar{\alpha}_n}} \quad & \sum_{n\in N}\left(\lambda_n-\bar{\alpha}_n\right)P_n^d-\sum_{n\in N}\left(\underline{\gamma}_n+\bar{\gamma}_n\right)\bar{\delta} \\
& -\sum_{l\in L}\left(\underline{\omega}_l+\bar{\omega}_l\right)\bar{P}_l^f+\sum_{j\in J}\underline{\theta}_n\underline{P}_j^g-\sum_{j\in J}\bar{\theta}_n\bar{P}_j^g
\end{aligned} \tag{3.55}$$

$$\text{s.t.} \sum_{l\in L} v_l B_l \mu_l A_{nl}-\underline{\gamma}_n+\bar{\gamma}_n=0, \quad \forall\, n\in N:(\delta_n) \tag{3.56}$$

$$\lambda_{n|j\in J_n}+\underline{\theta}_j-\bar{\theta}_j\le 0, \quad \forall\, j\in J:(P_j^g) \tag{3.57}$$

$$\mu_l-\sum_{n\in N}A_{nl}\lambda_n+\underline{\omega}_l-\bar{\omega}_l=0, \quad \forall\, l\in L:(P_l^f) \tag{3.58}$$

$$-\lambda_n-\underline{\alpha}_n+\bar{\alpha}_n\le 1, \quad \forall\, n\in N:(\Phi_n) \tag{3.59}$$

$$\underline{\theta}_j,\bar{\theta}_j,\underline{\omega}_l,\bar{\omega}_l,\underline{\alpha}_n,\bar{\alpha}_n,\underline{\gamma}_n,\bar{\gamma}_n\ge 0,\ \forall\, l\in L,\ \forall\, n\in N,\ \forall\, j\in J \tag{3.60}$$

The variables written in parenthesis are the primal variables corresponding to the dual constraints. As mentioned earlier, using the duality theory approach results in bilinear terms in the dual constraints (3.56) that can be linearized using traditional methods (Floudas 1995).

Single Level Model

In order to convert the middle and lower levels in the DAD model into a single-level program, the duality theory approach will be used. The primal constraints of the lower-level problem will be included, along with the dual constraints, and an equality between the objective function's value of the primal and dual problems to satisfy the strong duality condition. The following equations summarize the single-level AD model.

(AD)

$$\max_{\substack{v_l,\boldsymbol{\delta},\boldsymbol{P^g},\boldsymbol{P^f},\boldsymbol{\Phi},z_l^s,z_l^r,r_l^s,r_l^r,\\ \mu_l,t_l,b_l,\lambda_n,\underline{\theta}_j,\bar{\theta}_j,\underline{\omega}_l,\bar{\omega}_l,\underline{\alpha}_n,\bar{\alpha}_n}} \quad \sum_{n\in N} \Phi_n \tag{3.61}$$

$$\text{s.t.} \quad \sum_{l\in L}(1-v_l) \leq K \tag{3.62}$$

$$z_l \leq v_l, \quad \forall\, l \in L \tag{3.63}$$

$$v_l \in \{0,1\}, \quad \forall\, l \in L \tag{3.64}$$

$$P_l^f = B_l(z_l^s - z_l^r), \quad \forall\, l \in L \tag{3.65}$$

$$z_l^s = \delta_{s(l)} - r_l^s, \quad \forall\, l \in L \tag{3.66}$$

$$z_l^r = \delta_{r(l)} - r_l^r, \quad \forall\, l \in L \tag{3.67}$$

$$\underline{\delta} v_l \leq z_l^s \leq \bar{\delta} v_l, \quad \forall\, l \in L \tag{3.68}$$

$$\underline{\delta} v_l \leq z_l^r \leq \bar{\delta} v_l, \quad \forall\, l \in L \tag{3.69}$$

$$\underline{\delta}\,(1-v_l) \leq r_l^s \leq \bar{\delta}\,(1-v_l)\,, \quad \forall\, l \in L \tag{3.70}$$

$$\underline{\delta}\,(1-v_l) \leq r_l^r \leq \bar{\delta}\,(1-v_l)\,, \quad \forall\, l \in L \tag{3.71}$$

$$\sum_{j\in J_n} P_j^g - \sum_{l\in L} A_{nl} P_l^f + \Phi_n = P_n^d, \quad \forall\, n \in N \tag{3.72}$$

$$-\bar{P}_l^f \leq P_l^f \leq \bar{P}_l^f, \quad \forall\, l \in L \tag{3.73}$$

$$\underline{P}_j^g \leq P_j^g \leq \bar{P}_j^g, \quad \forall\, j \in J \tag{3.74}$$

$$0 \leq \Phi_n \leq P_n^d, \quad \forall\, n \in N \tag{3.75}$$

$$\sum_{l\in L} B_l t_l A_{nl} = 0, \quad \forall\, n \in N \tag{3.76}$$

$$t_l = \mu_l - b_l, \quad \forall\, l \in L \tag{3.77}$$

$$\underline{\mu}_l v_l \leq t_l \leq \bar{\mu}_l v_l, \quad \forall\, l \in L \tag{3.78}$$

$$\underline{\mu}_l \left(1 - v_l\right) \leq b_l \leq \bar{\mu}_l \left(1 - v_l\right), \quad \forall\, l \in L \tag{3.79}$$

$$\lambda_{n|j \in J_n} + \underline{\theta}_j - \bar{\theta}_j \geq 0, \quad \forall\, j \in J \tag{3.80}$$

$$\mu_l - \sum_{n \in N} A_{nl} \lambda_n + \underline{\omega}_l - \bar{\omega}_l = 0, \quad \forall\, l \in L \tag{3.81}$$

$$1 - \lambda_n - \underline{\alpha}_n + \bar{\alpha}_n \geq 0, \quad \forall\, n \in N \tag{3.82}$$

$$\sum_{n \in N} \left(\lambda_n - \bar{\alpha}_n\right) P_n^d - \sum_{l \in L} \left(\underline{\omega}_l + \bar{\omega}_l\right) \bar{P}_l^f + \sum_{j \in J} \underline{\theta}_j \underline{P}_j^g - \sum_{j \in J} \bar{\theta}_j \bar{P}_j^g = \sum_{n \in N} \Phi_n \tag{3.83}$$

$$\underline{\theta}_j, \bar{\theta}_j, \underline{\omega}_l, \bar{\omega}_l, \underline{\alpha}_n, \bar{\alpha}_n \geq 0, \; \forall\, j \in J, \; l \in L, \; n \in N \tag{3.84}$$

Equations(3.61) to (3.64) represent the middle-level objective function and constraints. Constraints (3.65) to (3.75) represent the primal feasibility constraints of the lower-level operator model. Specifically, (3.65) to (3.71) represent the power flow constraint in each line, and its linearizing constraints. Equations (3.72) represent the power balance equation in each bus. Equations (3.73) to (3.75) represent the upper and lower bounds for the power flow in each line, power output from the generator and the load shed in each bus, respectively. Starting from (3.76) until the end represent the constraints associated with the dual of the lower-level problem. Specifically, (3.76) represents the dual constraint associated with the power angle at each bus (δ_n), whereas (3.77) to (3.79) represent its linearizing constraints. Furthermore, (3.80) to (3.82) represent the dual constraints corresponding to P_j^g, P_l^f and Φ_n, respectively. The strong duality constraint that equates the primal objective function of the lower-level problem to its dual objective is represented in (3.83). Lastly, constraints(3.84) represent the non-negativity conditions on the dual variables.

3.4.4 Modified Enumeration Algorithm

In order to compare the quality of the results from our proposed solution approaches, an exact method for solving the DAD model is needed. Enumeration algorithms in TLP basically decompose the tri-level problem into bi-level sub-problems arranged in a tree-like structure. Although they are computationally expensive, they can be used to find multiple optimal solutions, if they exist, for small-sized problems. We propose a modified enumeration algorithm (MEA) that is inspired from (Scaparra & Church 2008), where they provide an implicit enumeration algorithm based on an observation made through investigating optimal solutions of a facility interdiction problem with fortification. Simply put, the observation states that the *defender/planner (i.e., first level decision-maker) has to protect at least one of the elements that are going to be attacked to maximize the inflicted damage in case there was no protection at all.*

Scaparra & Church (2008) implemented a binary tree search based on that observation on a BLP. Implementation of their enumeration algorithm on a TLP is basically the same except that each node in the tree represents a BLP instead of a single level problem. Figure 3.3 demonstrates the binary tree search enumeration algorithm for solving a DAD model of an electrical transmission network shown in Figure 3.2.

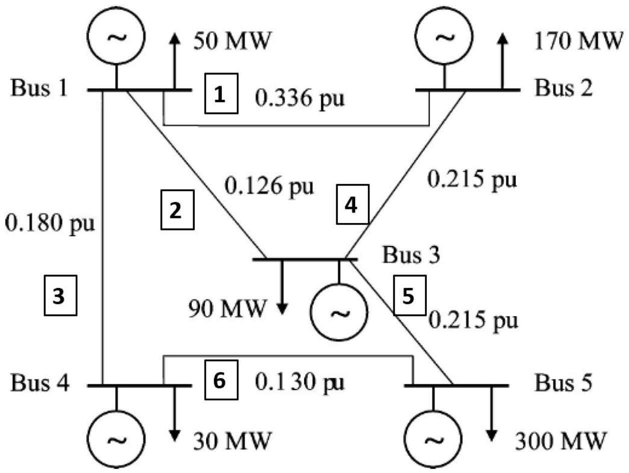


Figure 3.2: Five-Bus electrical transmission network.

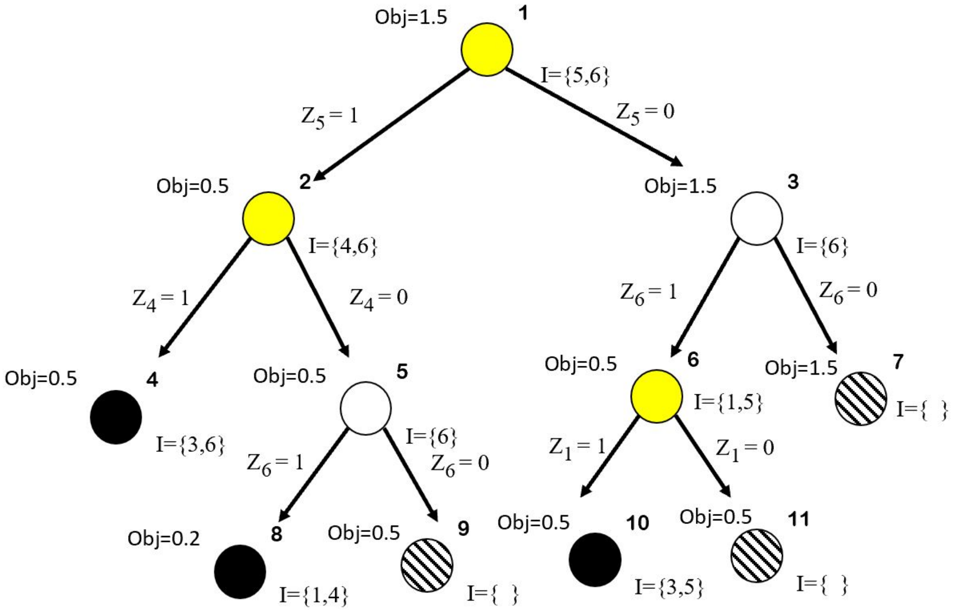


Figure 3.3: Scaparra & Church (2008) enumeration algorithm (D=2, K=2) for network in Figure 3.2.

Consider the instance of having equal defence and attack budgets of 2 transmission lines ($D = 2, K = 2$) in the electrical transmission network 3.2. Nodes are numbered from 1 to 11; each represents solving an instance of the AD model with different fortification strategies. Node 1 represents the parent node, which involves solving with no protection at all. This resulted in attacking Lines 5 and 6, which are the lines that cause maximum damage with a load shed of 1.5×10^2 MW if there was no fortification. We choose one of the lines randomly (Line 5) and branch on two nodes: one representing the AD model for protecting Line 5 ($Z_5 = 1$), and the other proceeds with no protection on the same line ($Z_5 = 0$). Yellow and black nodes represent a solved AD model. The white and hashed nodes are not solved for, as they inherit the same characteristics of the parent node, except for the candidate lines for protection. For instance, Node 3 inherited the same characteristics of Node 1 except that the candidate line for protection is line 6, since $Z_5 = 0$.

There are two fathoming rules. First, if the set of candidate lines for protection is empty such as Nodes 7, 9 and 11 (hashed nodes). Second, if the defence budget for the planner is reached ($D = 2$) which is the case for Nodes 4, 8 and 10 (black nodes). In order to know the defence strategy for a specific node, the path is traced from that node to parent node 1. For example, the defence strategy for Node 8 is $Z_5 = 1$ and $Z_6 = 1$.

Our MEA is based on a *non-binary tree search* implementation of the same problem, where nodes are branched according to the number of attacked lines (K=2) as demonstrated in Figure 3.4. Node 1 is the parent node, where the AD model is solved without any fortification. Nodes 2 and 3 are branched from the candidate set of lines resulting from solving Node 1. Branching is done in the same way until the defence budget (D=2) is exhausted as shown in Nodes 4, 5, 6 and 7. Numbers in the

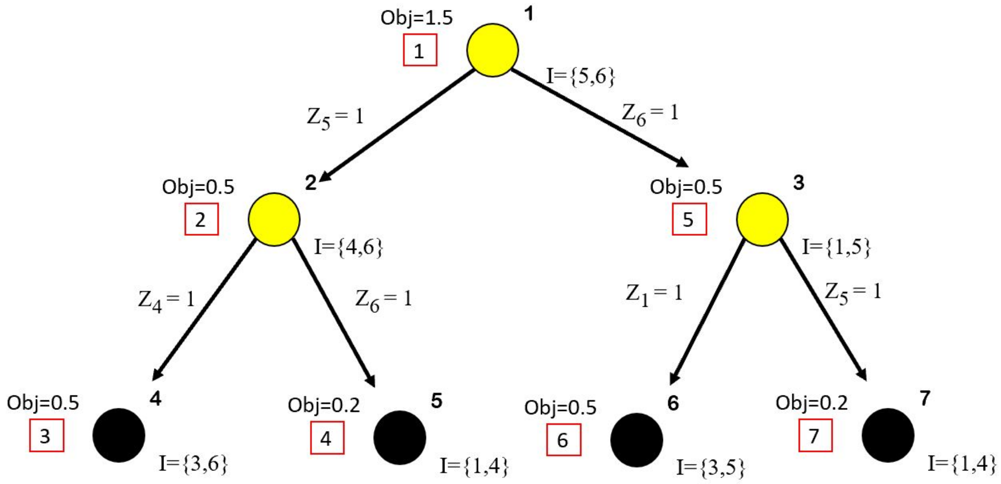


Figure 3.4: Non-binary search tree with minimum upper bound exploration (D=2, K=2).

red boxes represent the sequence by which each node is solved. Since each parent node represents an upper bound for the children nodes, the algorithm is programmed to explore (solve the AD model) nodes with the minimum upper bound first, then continues until the list of waiting nodes (not solved) is empty. For example, after Node 1 is solved, Nodes 2 and 3 have an equal upper bound of 1.5×10^2 MW. Thus, the algorithm solves the first node in the list (Node 2), which has an objective value of 0.5×10^2 MW. Nodes 4 and 5 are branched from Node 2, and both of them have an upper bound of 0.5×10^2 MW. Hence Nodes 4 and 5 are solved before Node 3, which has an upper-bound of 1.5×10^2 MW. However, in this version of the algorithm, protection patterns may be repeated. For instance Nodes 5 and 7 have similar protection strategies $Z_5 = 1$ and $Z_6 = 1$, which means that the same instance of the AD problem is being solved twice as both nodes have same properties.

In order to avoid redundant solutions, a record is kept with the solved nodes and their respective protection strategies. Before the solver attempts to solve any new node, protection strategies of the new node are compared with the records of solved nodes strategies. If they are the same, the node is fathomed and the solver does not attempt to solve that instance. This is demonstrated in Figure 3.5. Since Nodes 5 and 7 have the same protection strategy, and the algorithm explores nodes by the pattern indicated by the red boxes, Node 5 is solved before Node 7. Thus when the algorithm tries to explore Node 7, a comparison of protection strategies of the solved nodes is done before the solver attempts to solve. Hence, a similarity between Node 5 (solved) and Node 7 (still to be solved) in terms of protection strategy is found and Node 7 is fathomed without solving for that instance.

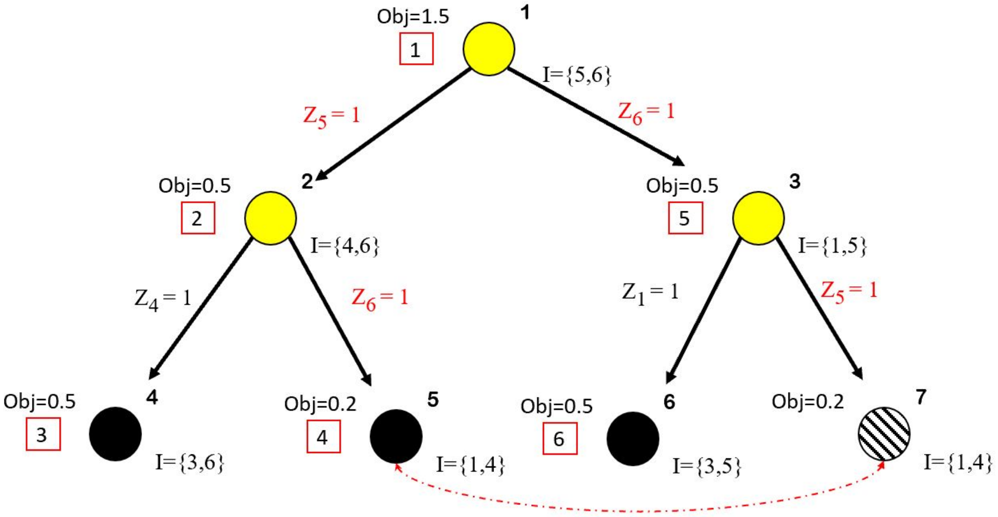


Figure 3.5: MEA Algorithm with (D=2, K=2).

Modified Enumeration Algorithm with Warm-Starting Solutions[1]

In order to accelerate the MEA algorithm, children nodes can be initialized using some information from the parent node, given the similarity between the two problems solved at each node. At first, we have experimented by initializing children nodes with the optimal solution of the parent node; however this yields an infeasible Mixed-binary Program (MBP) start and makes the optimization solver (CPLEX) takes more time trying to fix the infeasible solution. The infeasibility stems from the assumption that a defended transmission line can not be attacked, and children nodes are created by defending an attacked line from parent node, as depicted in Figure 3.5. To avoid an infeasible MIP start, a set is formed with the difference between attacked lines resulting from solving parent node and defended lines of the current node; the cardinality of this set should be equal to ($\textit{Attack Budget} - 1$). Hence, an additional line needs to be added to the initial solution; this line is chosen randomly from the difference of two sets. The first is the set of all lines, and the second is the union of lines that are defended and lines that are already chosen as initial solutions from the parent node. An analysis has been done to compare the running time of MEA algorithm with and without warm-starting solutions. We have noticed that warm-starting solutions save time for large instances, while it is slightly slower in small instances. We have summarized performance measures such as created nodes, solved nodes and average run-time for the enumeration algorithm Scaparra & Church (2008), MEA and MEA with warm-starting solutions in Table 3.3. We

[1] We would like to thank an anonymous reviewer for suggesting the warm-starting idea.

observed that MEA with warm starting solutions can save time up-to 56.25% compared to the classical enumeration algorithm, when the nodes being solved passes 10. For small instances, classical enumeration algorithm is slightly superior than MEA with warm-starting solutions. Moreover, we have provided a comparison between the performance of MEA with and without warm-starting solutions in Table 3.4. We noticed that MEA without warm-starting solutions can save time in small instances with small difference compared to initialized MEA. However, when it comes to large instances, MEA with warm-starting solutions can save time up to 15%.

Instance	Def.	Att.	MEA with Warm-starting Sol.			Classical Enumeration Alg.			Percentage of
Num.	Budget	Budget	Created Nodes	Solved Nodes	Avg. Run-time	Created Nodes	Solved Nodes	Avg. Run-time	Saved Time
1	1	3	4	4	0.64	7	4	0.52	-
2	1	4	5	5	0.71	9	5	0.65	-
3	2	2	7	6	0.90	11	6	0.59	-
4	2	3	13	10	1.03	19	10	1.34	23.14
5	2	4	21	11	1.29	29	15	2.05	36.98
6	2	5	31	23	1.49	45	21	2.06	27.54
7	2	6	43	29	1.54	55	28	2.18	29.17
8	3	2	13	11	1.06	23	12	1.82	41.47
9	3	3	31	20	1.52	45	23	3.58	55.47
10	3	4	61	37	2.77	79	40	4.73	41.32
11	3	5	116	66	3.48	137	69	6.01	42.02
12	4	1	5	5	0.70	8	5	0.89	20.9
13	4	2	23	17	1.34	41	21	2.86	52.9
14	4	3	61	38	2.56	103	52	5.85	56.25
15	4	4	161	87	5.01	199	100	9.73	48.5

Table 3.3: Comparing MEA with Warm-starting Solutions against Classical Enumeration Algorithm

Instance Num.	Def. Budget	Att. Budget	MEA with Warm-starting Sol. Avg. Run-time	MEA Avg. Run-time	Percentage of Saved Time
1	1	3	0.64	0.60	5.7
2	1	4	0.71	0.67	-
3	2	2	0.90	0.83	-
4	2	3	1.03	1.00	-
5	2	4	1.29	1.33	3.2
6	2	5	1.49	1.60	6.75
7	2	6	1.54	1.58	2.4
8	3	2	1.06	1.05	-
9	3	3	3.58	1.71	-
10	3	4	2.77	2.67	-
11	3	5	3.48	3.72	6.4
12	4	1	0.70	0.64	-
13	4	2	1.34	1.44	-
14	4	3	2.56	2.81	8.45
15	4	4	5.01	5.95	15.9

Table 3.4: Comparing MEA with and without Warm-starting Solutions

3.4.5 Numerical Results

Our proposed solution approaches are tested on three different electrical transmission networks and compared to exact solutions obtained with MEA. The heuristic-based algorithms have been programmed in MATLAB R2018b, and optimized by connecting the MATLAB toolbox function to the IBM ILOG CPLEX V 12.7.1 optimization software Manual (1987). Moreover, MEA has been implemented using GAMS Studio V 27.1.0, while using CPLEX as a solver. Numerical results have been carried out on an Intel Core I7 CPU (7th generation) at 2.7 GHZ with 8 GB of RAM and 64-bit operating system.

Five-Bus System

The first electrical transmission network is shown in Figure 3.2 and was used before in Arroyo & Galiana (2005). It consists of 6 transmission lines, 5 generators and 5 buses. The loads are specified on each bus, as well as the per unit reactance of each line. It is worth mentioning that the BMVA (Base-Mega-Volt-Ampere) and BkV (Base-kilo-Volt) are taken as 100 MVA and 138 kV, respectively. The maximum power flow ($\bar{P}^f_l$) in each transmission line has been set to 100 MW, while the maximum and minimum power ($\bar{P}^g_j$, $\underline{P}^g_j$,) that a generator can produce is set to 150 and zero MW, respectively. Moreover, transmission lines are numbered (squared boxes) as per Figure 3.2.

The heuristic-based solution approaches and MEA algorithm have been tested on all possible defence and attack budgets' scenarios for the Five-Bus system. The results are summarized in Table 3.5, where we report the attacked line(s), objective value and defended line(s), if relevant. We note that since we have six transmission lines in total, (i.e. $K + D \leq 6$), there are no feasible combinations in the lower diagonal part for each solution approach.

Table 3.5: Load Shed for Five-Bus System with Six Lines.

			Defence Budget					
			D=0	D=1	D=2	D=3	D=4	D=5
Attack Budget	*LPRank*	K=0	0	$0, L6$	$0, L5, L6$	$0, L3, L5, L6$	$0, L3, L4, L5, L6$	$0, L1, L3, L4, L5, L6$
		K=1	$L6, 50$	$L5, 50, L6$	$L4, 0, L5, L6$	$L4, 0, L3, L5, L6$	$L1, 0, L3, L4, L5, L6$	—"—
		K=2	$L5, L6, 150$	$L4, L5, 50, L6$	$L1, L4, 20, L5, L6$	$L1, L4, 20, L3, L5, L6$	$L1, L2, 10, L3, L4, L5, L6$	
		K=3	$L3, L5, L6, 150$	$L1, L4, L5, 70, L6$	—"—	—"—		
		K=4	$L3, L4, L5, L6, 150$	—"—	—"—			
		K=5	$L1, L3, L4, L5, L6, 170$	—"—				
		K=6	—"—					
	HybridRank	K=0	0	$0, L6$	$0, L5, L6$	$0, L4, L5, L6$	$0, L1, L4, L5, L6$	$0, L1, L3, L4, L5, L6$
		K=1	$L6, 50$	$L5, 50, L6$	$L4, 0, L5, L6$	$L1, 0, L4, L5, L6$	—"—	—"—
		K=2	$L5, L6, 150$	$L4, L5, 50, L6$	$L1, L4, 20, L5, L6$	$L1, L2, 10, L4, L5, L6$	—"—	
		K=3	$L4, L5, L6, 150$	$L1, L4, L5, 70, L6$	—"—	—"—		
		K=4	$L1, L4, L5, L6, 170$	—"—	—"—			
		K=5	—"—	—"—				
		K=6	—"—					
	MBLPRank	K=0	0	$0, L6$	$0, L5, L6$	$0, L4, L5, L6$	$0, L1, L4, L5, L6$	$0, L1, L3, L4, L5, L6$
		K=1	$L6, 50$	$L5, 50, L6$	$L4, 0, L5, L6$	$L1, 0, L4, L5, L6$	—"—	—"—
		K=2	$L5, L6, 150$	$L4, L5, 50, L6$	$L1, L4, 20, L5, L6$	$L1, L2, 10, L4, L5, L6$	—"—	
		K=3	$L4, L5, L6, 150$	$L1, L4, L5, 70, L6$	—"—	—"—		
		K=4	$L1, L4, L5, L6, 170$	—"—	—"—			
		K=5	—"—	—"—				
		K=6	—"—					
	MEA	K=0	0	$0, L6$	$0, L5, L6$	$0, L4, L5, L6$	$0, L1, L3, L5, L6$	$0, L2, L3, L4, L5, L6$
		K=1	$L6, 50$	$L5, 50, L6$	$L3, 0, L5, L6$	$L1, 0, L4, L5, L6$	$L4, 0, L1, L3, L5, L6$	$L1, 0, L2, L3, L4, L5, L6$
		K=2	$L5, L6, 150$	$L1, L5, 50, L6$	$L1, L4, 20, L5, L6$	$L3, L4, 0, L1, L5, L6$	$L3, L4, 0, L1, L5, L6$	
		K=3	$L1, L5, L6, 150$	$L1, L4, L5, 70, L6$	—"—	—"—		
		K=4	$L1, L4, L5, L6, 170$	—"—	—"—			
		K=5	—"—	—"—				
		K=6	—"—					

The results obtained by the MEA algorithm have been validated with those obtained by (Arroyo & Galiana 2005) with an AD model where the defence budget is zero as in the first column ($D = 0$) of Table 3.5. It is worth mentioning that increasing the attack budget does not necessarily increase the load shed. For example, if the number of destroyed lines is increased from 2 to 3, this increase in attack budget will result in the same load shed of 150 MW. This is due to the fact that the system has a high robustness or low vulnerability to attacks due the presence of generators at each bus. We also note that the maximum load shed that can be achieved is 170

MW. This an upper bound that can be calculated without solving a MBLP using the following formula:

$$UB = \sum_{n \in N} P_n^d \; - \; min(P_n^d, \bar{P}^g_{j(n)}) \tag{3.85}$$

The UB is system specific. Looking at the system structure in Figure 3.2 and imagining that each node is isolated, the portion of load that can and will always be satisfied is that which is connected to the node less than or equal to the generator maximum power connected to the same node. We use the UB values as a stopping criteria for the proposed heuristics, as it is more computationally efficient than solving a MBLP.

As for the *LPRank* we notice that it found an optimal solution in about 86% of the instances (24 out of 28 instances). According to *LPRank*, the highest priority lines are ranked as follows: $L6$, $L5$, $L3$, $L4$, $L1$, $L2$, as can be seen from the first column ($D = 0$) of Table 3.5. *HybridRank* and *MBLPRank* both yielded same solutions, however the computational burden of *MBLPRank* is higher than that of *HybridRank* as illustrated later for the larger bus problem in Subsection 3.4.5. Both heuristics have the same solution quality solving about 93% of the instances to optimality (26 out of 28 instances). *LPRank* also failed to obtain optimal solutions for the same two instances. However, the solution quality of *HybridRank* or *MBLPRank* is better than that of *LPRank* (half the value) in those two instances. According to both *HybridRank* and *MBLPRank*, the highest priority lines are ranked as follows: $L6$, $L5$, $L4$, $L1$, $L3$, $L2$.

In Table D.1, in Appendix D, we include further computational results from

LPRank such as average run time and number of LPs solved.

Six-Bus System

The second electrical transmission network has been studied previously by Jiang et al. (2019). The system consists of eight transmission lines, two generators and six buses. The input parameters for the system's generator, load, branch data and line numbering have been taken from (Jiang et al. (2019)) for validation purposes.

Table 3.6: Load Shed for Six-Bus System (8 Lines) Under Different Attack and Defence Budgets

		Defence Budget				
Attack Budget		D=0	D=1	D=2	D=3	D=4
LPRank	K=1	L5,43.6	L2,31.1, L5	L4, 25, L2, L5	L4,25,L2, L3, L5	L7,0, L2, L3, L4, L5
	K=2	L2, L5,130	L2, L3,110.8, L5	L4, L6,70, L2, L5	L4, L6,70, L2, L3, L5	—"—
	K=3	L2, L3, L5,210	L2, L3, L4,170, L5	L3, L4, L6,90, L2, L5	L4, L6, L8,70, L2, L3, L5	—"—
	K=4	L2, L3, L4, L5,290	L2, L3, L4, L7,170, L5	L3, L4, L6, L8,150, L2, L5	—"—	—"—
	K=5	—"—	L2, L3, L4, L6, L7,220, L5	—"—	—"—	
	K=6	—"—	—"—	—"—		
	K=7	—"—	—"—			
	K=8	—"—				
HybridRank	K=1	L5,43.6	L2,31.1, L5	L4,25, L2, L5	L4,25, L2, L3, L5	L7,0, L2, L3, L4, L5
	K=2	L2, L5,130	L2, L3,110.8, L5	L4, L6,70, L2, L5	L4, L6,70, L2, L3, L5	—"—
	K=3	L2, L3, L5,210	L2, L3, L4,170, L5	L3, L4, L6,90, L2, L5	L4, L6, L8,70, L2, L3, L5	—"—
	K=4	L2, L3, L4, L5,290	L2, L3, L4, L7,170, L5	L3, L4, L6, L8,150, L2, L5	—"—	—"—
	K=5	—"—	L2, L3, L4, L6, L7,220, L5	—"—	—"—	
	K=6	—"—	—"—	—"—		
	K=7	—"—	—"—			
	K=8	—"—				
MBLPRank	K=1	L5,43.6	L2,31.1, L5	L4,25, L2, L5	L4,25, L2, L3, L5	L7,0, L2, L3, L4, L5
	K=2	L2, L5, 130	L2, L3, 110.8, L5	L3, L8, 80, L2, L5	L4, L6, 70, L2, L3, L5	—"—
	K=3	L2, L3, L5, 210	L2, L3, L4, 170, L5	L3, L4, L8, 90, L2, L5	L4, L6, L8, 70, L2, L3, L5	—"—
	K=4	L2, L3, L4, L5, 290	L1, L2, L3, L4,170, L5	L3, L4, L6, L8, 150, L2, L5	—"—	—"—
	K=5	—"—	L2, L3, L4, L6, L7, 220, L5	—"—	—"—	
	K=6	—"—	—"—	—"—		
	K=7	—"—	—"—			
	K=8	—"—				
MEA	K=1	L5,43.6	L2,31.1, L5	L4,25, L2, L5	L3,20, L2, L4, L5	L7,0, L2, L3, L4, L5
	K=2	L2, L5,130	L4, L5,95, L2	L3, L8,80, L2, L5	L5, L7,60, L2, L3, L4	—"—
	K=3	L2, L4, L5,210	L2, L3, L4,170, L5	L2, L4,90, L3, L5	L5, L6, L7,70, L2, L3, L4	—"—
	K=4	L2, L3, L4, L5,290	L2, L3, L4, L7,170, L5	L4, L5, L6, L7,140, L2, L3	—"—	—"—
	K=5	—"—	L1, L2, L4, L5, L6,210, L3	—"—	—"—	
	K=6	—"—	—"—	—"—		
	K=7	—"—	—"—			
	K=8	—"—				

Results from the proposed solution algorithms have been grouped and summarized in Table 3.6. MEA algorithm confirmed the results obtained via the protection strategy by Jiang et al. (2019), where two instances are listed: one when the attack and defence budgets are both equal to 2 (i.e $K = 2,\ D = 2$) yielding 80 MW, and the other when $K = 2$ and $D = 3$ resulting in 60 MW total load shed. In total, there are 45 possible instances satisfying the inequality $K + D \leq 6$. Nevertheless, Table 3.6 shows only 30 of them as all proposed algorithms yielded optimal solutions in those 15 instances with zero MW load shed. Regarding the heuristic-based approaches, *LPRank* and *HybridRank* had the same solution quality for all instances. Both of them obtained optimal solutions for 35 instances (about 78% of the instances), while near-optimal results were obtained for the remaining 10 instances Moreover, *MBLPRank* obtained optimal solutions for 36 instances (about 80% of instances). It is worth mentioning that *MBLPRank* failed to obtain optimal results for the same instances as those of *HybridRank* and *LPRank* except for one, when $K = 2,\ D = 2$. All heuristic-based approaches gave the same priority to the transmission lines as: $L5,\ L2,\ L3,\ L4$. The ranking of the rest of the transmission lines does not make a difference as the system becomes robust (with zero MW load shed) after protecting the previously mentioned lines.

In Tables D.2 and D.4, in Appendix D, we include additional computational results. Table D.2 shows several performance measures such as average running time, number of LPs and MBLPs solved. We note that the number of MBLPs solved is minimal and we did not need to solve any MBLPs in several instances. Table D.4 shows average running time for each proposed heuristic; this is compared to the running time taken by MEA with warm-starting solutions to obtain the global optimal solution. As expected, *LPRank* heuristic took the least time compared to *HybridRank* and *MBLPRank*. The difference between the average running time of *HybridRank* and *MBLPRank* is minimal. In some instances *HybridRank* took more time than *MBLPRank*; this is mainly because some instances require invoking both the linear program and mixed-binary linear program solvers. For LPRank, the number of linear programs solved for each instance is reported; while for HybridRank, the number of linear programs and mixed-binary linear programs is recorded in Table D.4. Additionally, the time taken by MBLPRank and number of mixed-binary linear programs to reach the solution is added to Table D.4. Instances that are highlighted in bold are those which achieved global optimality as obtained by MEA.

IEEE 57-BUS System

The third electrical transmission network consists of 57 buses, 80 transmission lines and 7 generators. The single line diagram and dataset used for the 57-Bus system is available in the Appendix of (Jiang et al. 2019).

It should be noted that for this system the DAD problem consists of 160 binary variables. Solving a single node (AD model) for the 57-Bus system using the MEA algorithm takes on average 432 seconds. As an example of the computational burden, consider one instance of the DAD problem, when $K = 5, D = 4$, which results in the creation of 781 nodes (AD problems). In the worst-case scenario, if the solver attempts to solve all nodes with an average of 432 seconds per node, the total computational time will be 93.28 hours. Using the heuristic-based approaches in such relatively large problems becomes very useful, where near-optimal solutions are widely accepted in most practical applications.

In Table 3.7, we list *LPRank* priority for up to 20 lines, as the maximum load shed occurs at $K = 20$, when *LPRank* is used. The computational time that was taken for *LPRank* was exactly 33.67 seconds.

Table 3.7: *LPRank* Priority for 57-Bus System (80 Lines)

	(MW)	**obj. val.**	**Lines**		**obj. val.**	**Lines**
LPRank	K=1	75.6	$L18$	K=11	376.6	$L7$
	K=2	131.2	$L41$	K=12	391.8	$L22$
	K=3	197	$L8$	K=13	391.8	$L2$
	K=4	246.7	$L17$	K=14	391.8	$L65$
	K=5	292	$L16$	K=15	391.8	$L25$
	K=6	304.6	$L29$	K=16	391.8	$L27$
	K=7	376.6	$L15$	K=17	391.8	$L23$
	K=8	376.6	$L3$	K=18	424.8	$L11$
	K=9	376.6	$L1$	K=19	424.8	$L21$
	K=10	376.6	$L26$	K=20	449.8	$L5$

Table 3.8 lists the repository of *HybridRank* algorithm along with the high priority lines resulting from that repository. Line numbers mentioned on the same line have equal priority. The second column of Table 3.8 explains the progression of implementing *HybridRank* algorithm until the maximum load shed is achieved (449.8 MW). As previously mentioned, if the objective value is the same for two successive iterations, *HybridRank* triggers the MBLP solver. In total, there were 7 MBLPs solved until the maximum load shed was achieved. The most critical lines found were $L18$ and $L8$, as they had the highest number of counts. The computational time for applying *HybridRank* was 13.5 minutes.

Table 3.9 lists the repository for *MBLPRank*. It has the same format as Table

Table 3.8: *HybridRank* Repository for 57-Bus System (80 Lines)

	(MW)	**Type**	**obj. val.**	**Lines**	**High Priority Lines**
HybridRank	K=1	*LP*	75.6	$L18$	$L18$
	K=2	*LP*	131.2	$L18$, $L41$	$L8$
	K=3	*LP*	197	$L8$, $L18$, $L41$	$L41$, $L17$
	K=4	*LP*	246.7	$L8$, $L17$, $L18$, $L41$	$L16$
	K=5	*LP*	292	$L8$, $L16$, $L17$, $L18$, $L41$	$L15$
	K=6	*LP*	304.6	$L8$, $L16$, $L17$, $L18$, $L29$, $L41$	$L3$
	K=7	*LP*	376.6	$L8$, $L15$, $L16$, $L17$, $L18$, $L29$, $L41$	$L26$, $L21$, $L5$
	K=8	*LP*	376.6	$L3$, $L8$, $L15$, $L16$, $L17$, $L18$, $L29$, $L41$	$L20$, $L19$
	K=8	*MBLP*	403.8	$L8$, $L15$, $L16$, $L17$, $L18$, $L19$, $L20$, $L41$	$L29$, $L27$, $L25$, $L23$, $L11$, $L1$
	K=9	*LP*	403.8	$L3$, $L8$, $L15$, $L16$, $L17$, $L18$, $L19$, $L20$, $L41$	$L65$, $L30$, $L22$, $L6$, $L7$
	K=9	*MBLP*	416.8	$L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L41$	
	K=10	*LP*	416.8	$L1$, $L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L41$	
	K=10	*MBLP*	416.8	$L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L26$, $L41$	
	K=11	*LP*	416.8	$L1$, $L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L26$, $L41$	
	K=11	*MBLP*	416.8	$L3$, $L5$, $L6$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L22$, $L26$	
	K=12	*LP*	416.8	$L3$, $L5$, $L6$, $L7$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L22$, $L26$	
	K=12	*MBLP*	416.8	$L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L26$, $L30$, $L41$, $L65$	
	K=13	*LP*	416.8	$L1$, $L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L26$, $L30$, $L41$, $L65$	
	K=13	*MBLP*	436.8	$L8$, $L11$, $L15$, $L16$, $L17$, $L18$, $L19$, $L20$, $L23$, $L25$, $L26$, $L27$, $L41$	
	K=14	*LP*	436.8	$L3$, $L8$, $L11$, $L15$, $L16$, $L17$, $L18$, $L19$, $L20$, $L23$, $L25$, $L26$, $L27$, $L41$	
	K=14	*MBLP*	449.8	$L3$, $L5$, $L8$, $L11$, $L15$, $L16$, $L17$, $L18$, $L21$, $L23$, $L25$, $L26$, $L27$, $L41$	

3.8 without the problem type, as *MBLPRank* solves all problems as MBLPs and determines the priority of lines based on the count of lines in the repository. The elapsed time for *MBLPRank* was 37.3 minutes.

Table 3.9: *MBLPRank* Repository for 57-Bus System (80 Lines)

	(MW)	**obj. val.**	**Lines**	**High Priority Lines**
MBLPRank	K=1	75.6	$L18$	$L18$
	K=2	131.2	$L18$, $L41$	$L41$, $L8$
	K=3	197	$L8$, $L18$, $L41$	$L17$
	K=4	246.7	$L8$, $L17$, $L18$, $L41$	$L16$, $L15$
	K=5	297.6	$L8$, $L15$, $L16$, $L17$, $L41$	$L26$, $L21$, $L5$, $L3$
	K=6	367.3	$L8$, $L15$, $L16$, $L17$, $L18$, $L41$	$L27$, $L25$, $L23$, $L20$, $L19$, $L11$
	K=7	376.6	$L8$, $L15$, $L16$, $L17$, $L18$, $L29$, $L41$	$L65$, $L30$, $L29$, $L22$, $L6$
	K=8	403.8	$L8$, $L15$, $L16$, $L17$, $L18$, $L19$, $L20$, $L41$	
	K=9	416.8	$L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L41$	
	K=10	416.8	$L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L26$, $L41$	
	K=11	416.8	$L3$, $L5$, $L6$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L422$, $L26$	
	K=12	416.8	$L3$, $L5$, $L8$, $L15$, $L16$, $L17$, $L18$, $L21$, $L26$, $L41$, $L65$	
	K=13	436.8	$L8$, $L11$, $L15$, $L16$, $L17$, $L18$, $L19$, $L20$, $L23$, $L25$, $L26$, $L27$	
	K=14	449.8	$L3$, $L5$, $L8$, $L11$, $L15$, $L16$, $L17$, $L18$, $L21$, $L23$, $L25$, $L26$	

Table D.3, in Appendix D, shows randomly chosen 37 instances of the 57-Bus system. Running time for each of the proposed heuristics is shown along with the deviation from the global optimal solution. LPRank's performance was the best among the proposed heuristics, as it obtained global optimality in 6 instances. Moreover, the maximum absolute deviation for LPRank was 24% from the global optimal solution with significant time savings compared to the enumeration technique. HybridRank obtained global optimality in 3 instances with maximum deviation of 52.76%, while MBLPRank obtained global optimality in 11 instances with maximum deviation of 247%. It is worth mentioning that the computational time difference between the heuristics and accelerated enumeration algorithm might not be significant for small instances as can be seen in Tables D.1 and D.2. Nevertheless, it becomes much more significant for large instances as evident in Table D.3.

3.5 Conclusions

In this paper we present a new class of tri-level mixed integer linear programming. We discuss both its dual and KKT reformulations and present some structural analysis properties. Given the complexity of the problem, we present three solution approaches as well as an exact enumeration method, for benchmarking purposes. As an illustration, the solutions approaches were applied to improve the resilience of three different electrical transmission networks that varied in size. Our proposed algorithms provided optimal solutions in most of the test instances. Moreover, they proved to offer a good substitute when obtaining exact solutions is more computationally expensive for large problems instances.

The use of multi-level programming is becoming more prominent due to the increase of decentralized decision-making applications, which raises the need for future research works in this area pertaining to modelling and algorithm development. Depending on the application under study, meticulous modelling of the problem can pave the way to solving it efficiently and precisely. Nevertheless, this is not always applicable which raises the need to design different algorithms. Developing algorithms for solving tri-level programming problems is still in its infancy, and it comes with many challenges. One of the possible future extensions is to consider designing decomposition methods. Wu & Conejo (2017) used a variation of Benders decomposition, however it deviates from the classical Benders implementation and does not guarantee optimality. We have performed preliminary tests using that approach on our systems and found that it is not competitive both in terms of solution quality and time. Therefore, there is a potential for developing efficient decomposition approaches that guarantee optimality. The authors are currently pursuing this line of research. Another possible direction for future research is considering mixed-integer decision variables in all levels, where strong duality theory fails and we have to rely on weak duality for providing bounds. Using multi-parametric programming theories can be explored to develop algorithms, either in the general sense or suited to specific multi-level programming applications such as dealing with multi-dimensional knapsack constraints in more than one level. Furthermore, decision diagrams have been used recently to solve optimization problems and can be further investigated to be applied on discrete multi-level programming problems.

Acknowledgement

The authors would like to thank four anonymous reviewers for their excellent comments that have significantly improved the quality of the paper. The authors would also like to acknowledge the support of the Ontario Trillium Scholarship Program and the Natural Sciences and Engineering Research Council of Canada.

Bibliography

Alguacil, N., Delgadillo, A. & Arroyo, J. M. (2014), 'A trilevel programming approach for electric grid defense planning', *Computers & Operations Research* **41**, 282–290.

Alvarez, R. E. (2004), Interdicting electrical power grids, PhD thesis, Monterey, California. Naval Postgraduate School.

Anandalingam, G. (1988), 'A mathematical programming model of decentralized multi-level systems', *Journal of the Operational Research Society* **39**(11), 1021–1033.

Arroyo, J. M. (2010), 'Bilevel programming applied to power system vulnerability analysis under multiple contingencies', *IET generation, transmission & distribution* **4**(2), 178–190.

Arroyo, J. M. & Galiana, F. D. (2005), 'On the solution of the bilevel programming formulation of the terrorist threat problem', *IEEE transactions on Power Systems* **20**(2), 789–797.

Aussel, D. & Svensson, A. (2019), 'Is pessimistic bilevel programming a special case of a mathematical program with complementarity constraints?', *Journal of Optimization Theory and Applications* **181**(2), 504–520.

Babick, J. P. (2009), Tri-level optimization of critical infrastructure resilience, Technical report, Naval Postgraduate School Monterey United States.

Bard, J. F. (1984), 'An investigation of the linear three level programming problem', *IEEE Transactions on Systems, Man, and Cybernetics* (5), 711–717.

Bard, J. F. (1991), 'Some properties of the bilevel programming problem', *Journal of optimization theory and applications* **68**(2), 371–378.

Bard, J. F. (2013), *Practical bilevel optimization: algorithms and applications*, Vol. 30, Springer Science & Business Media.

Bard, J. F. & Moore, J. T. (1992), 'An algorithm for the discrete bilevel programming problem', *Naval Research Logistics (NRL)* **39**(3), 419–435.

Blair, C. (1992), 'The computational complexity of multi-level linear programs', *Annals of Operations Research* **34**(1), 13–19.

Brown, G., Carlyle, M., Salmerón, J. & Wood, K. (2006), 'Defending critical infrastructure', *Interfaces* **36**(6), 530–544.

Candler, W. & Norton, R. (1977), 'Multilevel programming and developing policy', *Technical Report 258, World Bank Development Research Center* .

Colson, B., Marcotte, P. & Savard, G. (2005), 'Bilevel programming: A survey', *4or* **3**(2), 87–107.

Colson, B., Marcotte, P. & Savard, G. (2007), 'An overview of bilevel optimization', *Annals of operations research* **153**(1), 235–256.

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

Dempe, S. & Dutta, J. (2012), 'Is bilevel programming a special case of a mathematical program with complementarity constraints?', *Mathematical programming* **131**(1), 37–48.

Fakhry, R., Hassini, E., Ezzeldin, M. & El-Dakhakhni, W. (2022), 'Tri-level mixed-binary linear programming: Solution approaches and application in defending critical infrastructure', *European Journal of Operational Research* **298**(3), 1114–1131.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. & Mirjalili, S. (2018), 'Hybrid optimizers to solve a tri-level programming model for a tire closed-loop supply chain network design problem', *Applied Soft Computing* **70**, 701–722.

Florensa, C., Garcia-Herreros, P., Misra, P., Arslan, E., Mehta, S. & Grossmann, I. E. (2017), 'Capacity planning with competitive decision-makers: trilevel milp formulation, degeneracy, and solution approaches', *European Journal of Operational Research* **262**(2), 449–463.

Floudas, C. A. (1995), *Nonlinear and mixed-integer optimization: fundamentals and applications*, Oxford University Press.

Fortuny-Amat, J. & McCarl, B. (1981), 'A representation and economic interpretation of a two-level programming problem', *Journal of the operational Research Society* **32**(9), 783–792.

Gu, Y., Cai, X., Han, D. & Wang, D. Z. (2019), 'A tri-level optimization model for a private road competition problem with traffic equilibrium constraints', *European Journal of Operational Research* **273**(1), 190–197.

Han, J., Zhang, G., Hu, Y. & Lu, J. (2016), 'A solution to bi/tri-level programming problems using particle swarm optimization', *Information Sciences* **370**, 519–537.

Hong, S., Cheng, H. & Zeng, P. (2017), 'Nk constrained composite generation and transmission expansion planning with interval load', *IEEE Access* **5**, 2779–2789.

Irohara, T., Kuo, Y.-H. & Leung, J. M. (2013), From preparedness to recovery: a tri-level programming model for disaster relief planning, *in* 'International Conference on Computational Logistics', Springer, pp. 213–228.

Jiang, P., Huang, S. & Zhang, T. (2019), 'Optimal deception strategies in power system fortification against deliberate attacks', *Energies* **12**(3), 342.

Krebs, V., Schewe, L. & Schmidt, M. (2018), 'Uniqueness and multiplicity of market equilibria on dc power flow networks', *European Journal of Operational Research* **271**(1), 165–178.

Lazzaro, G. L. (2016), Tri-level optimization algorithms for solving defender-attacker-defender network models, Technical report, Naval Postgraduate School Monterey United States.

Lin, Y. & Bie, Z. (2018), 'Tri-level optimal hardening plan for a resilient distribution system considering reconfiguration and dg islanding', *Applied Energy* **210**, 1266–1279.

Mahmoodjanloo, M., Parvasi, S. P. & Ramezanian, R. (2016), 'A tri-level covering fortification model for facility protection against disturbance in r-interdiction median problem', *Computers & Industrial Engineering* **102**, 219–232.

Manual, C. U. (1987), 'Ibm ilog cplex optimization studio', *Version* **12**, 1987–2018.

Moore, J. T. & Bard, J. F. (1990), 'The mixed integer linear bilevel programming problem', *Operations research* **38**(5), 911–921.

O'Flaherty, K. (2019), 'U.s. government makes surprise move to secure power grid from cyberattacks', `https://tinyurl.com/2wbv9bce`. Accessed: 2019-11-06.

Parkatti, V.-P., Assmuth, A., Rämö, J. & Tahvonen, O. (2019), 'Economics of boreal conifer species in continuous cover and rotation forestry', *Forest Policy and Economics* **100**, 55–67.

Ríos-Mercado, R. Z., Wu, S., Scott, L. R. & Boyd, E. A. (2002), 'A reduction technique for natural gas transmission network optimization problems', *Annals of Operations Research* **117**(1-4), 217–234.

Sarhadi, H., Tulett, D. M. & Verma, M. (2015), 'A defender-attacker-defender approach to the optimal fortification of a rail intermodal terminal network', *Journal of Transportation Security* **8**(1-2), 17–32.

Sarhadi, H., Tulett, D. M. & Verma, M. (2017), 'An analytical approach to the protection planning of a rail intermodal terminal network', *European Journal of Operational Research* **257**(2), 511–525.

Scaparra, M. P. & Church, R. L. (2008), 'A bilevel mixed-integer program for critical infrastructure protection planning', *Computers & Operations Research* **35**(6), 1905–1923.

Schweitzer, D. & Medal, H. (2019), 'Wireless lan transmitter location under the threat of jamming attacks', *Computers & Operations Research* **106**, 14–27.

Sinha, S. (2001), 'A comment on anandalingam (1988). a mathematical programming model of decentralized multi-level systems. j opl res soc 39: 1021-1033', *Journal of the Operational Research Society* **52**(5), 594–596.

Smith, R. (2014), 'Assault on california power station raises alarm on potential for terrorism', `https://tinyurl.com/5whn9x2m`. Accessed: 2019-04-30.

Sussman, B. (2019), 'Revealed: Details of 'first of its kind' disruptive power grid attack', `https://tinyurl.com/4ntum92s`. Accessed: 2019-11-06.

Vicente, L. N. & Calamai, P. H. (1994), 'Bilevel and multilevel programming: A bibliography review', *Journal of Global optimization* **5**(3), 291–306.

von Stackelberg, H. (2011), *Market Structure and Equilibrium*, Springer Berlin Heidelberg, Berlin, Heidelberg.

White, D. (1997), 'Penalty function approach to linear trilevel programming', *Journal of Optimization Theory and Applications* **93**(1), 183–197.

Wu, X. & Conejo, A. J. (2017), ‘An efficient tri-level optimization model for electric grid defense planning’, *IEEE Transactions on Power Systems* **32**(4), 2984–2994.

Xu, P. & Wang, L. (2014), ‘An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions’, *Computers & operations research* **41**, 309–318.

Yao, Y., Edmunds, T., Papageorgiou, D. & Alvarez, R. (2007), ‘Trilevel optimization in power network defense’, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **37**(4), 712–718.

Chapter 4

A Branch-and-Bound Algorithm for Solving Bi- and Tri-Level programmes

Abstract

Bi-level and tri-level mathematical programmes have been used in a variety of important applications such as critical infrastructure defence, machine learning (e.g., hyper-parameter optimization, and reinforcement learning), pricing and revenue management, and energy. Despite the need for general-purpose solvers for multi-level programmes, there has not been enough effort dedicated for algorithm development of such solvers. This is mainly due to the challenging nature of multi-level programmes that are proven to be $\mathcal{NP}$-hard even in their most simplest case of continuous bi-level linear programmes. Recently, there is an increasing interest in developing general-purpose solvers for bi-level programmes to cope with the diverse practical applications that are changing dynamically. This research work proposes a general-purpose branch-and-bound algorithm capable of handling mixed-integer variables in both levels of a bi-level linear programme. Moreover, it also solves a general class of tri-level mixed-integer programmes with a convex optimization problem being at the most lower-level. The class of tri-level programmes handled by our algorithm has been motivated by the important application of defending critical electrical power transmission networks. Although, this is a special class of tri-level programmes, the proposed algorithm can handle more general versions of tri-level programmes. Furthermore, in an effort to enrich the bi-level mixed-integer library of test problems and encourage knowledge transfer between different fields of research, we provide a Matlab live editor to convert any electrical transmission network into

a bi-level mixed-integer instance for algorithm testing purposes. We validate our algorithm with 100 randomly-generated instances from the literature and use our Matlab-based testing tool to generate instances for two electrical transmission networks.

4.1 Introduction

Multi-level decision-making, first introduced by Candler & Norton (1977), has been regarded as an important planning phase as it organizes multiple decision-makers' conflicting interactions in a hierarchical structure (Bard 2013). There are many applications that deal with multiple decision-makers acting in a competitive environment, and often with conflicting goals. Multi-level programmes allow us to model hierarchical optimization problems where decisions are made in a sequential fashion. Each decision-maker at a specific level affects the decisions made at the subsequent lower-levels in addition to the utility/payoffs at the upper-levels.

In this research work, we deal with bi-level linear optimization problems; which are proven to be $\mathcal{NP}$-hard by Jeroslow (1985), and a special class of tri-level problems that can be reduced to a bi-level mixed-integer linear programme (**BMILP**). Consequently, we mainly deal with two decision-makers: a leader who takes the first decision, and then the follower, affected by the leader's decisions, optimizes its corresponding objective function. Hence, the optimization problem from the leader's perspective can be viewed as a nested optimization programme, having the follower's problem as an inner-problem. As such, these multi-level problems can be viewed as sequential Stackelberg games von Stackelberg (2011) with perfect information, as both objective functions, upper-level, and lower-level constraints are known to both players; thus the leader can anticipate the follower's decisions and vice versa.

Multi-level optimization has been used to model practical applications such as cross-dock truck scheduling (Konur & Golias 2013), facility location (Cao & Chen 2006), bi-level knapsack and capacitated lot-sizing (Lozano & Smith 2017), taxation and highway pricing (Labbé et al. 1998), interdiction games (Fischetti et al. 2019), defending critical infrastructure (Alvarez (2004), Alguacil et al. (2014), Fakhry et al. (2022)), and natural gas planning (Dempe et al. 2011). Nevertheless, despite this increasing interest, and growing number of applications, there is a paucity of algorithms for general-purpose bi-level optimization solvers. Motivated by the work of (Bard & Moore 1990, Wen & Yang 1990, Xu & Wang 2014, Fischetti et al. 2017, Kleinert, Labbé, Plein & Schmidt 2021), we propose a general-purpose branch and bound algorithm for solving BMILPs. The driving application for our algorithm is that of protecting critical infrastructure (Brown et al. 2006, Arroyo & Galiana 2005, Arroyo 2010, Akbari-Jafarabadi et al. 2017, Alvarez 2004) with a focus on defending electrical power grids. Operations of other critical infrastructure, such as water and roads networks, are interdependent on electrical transmission networks, Thus, ensuring a reliable operation of electrical power grids is often given high priority for governments.

The motivation for using a branch-and-bound (**B&B**) approach for defending electrical power grids is three-folds:

- Finding alternative solutions that would enhance the set of options available for the leader (i.e., first decision-maker) which would protect the transmission network against worst-case scenarios in case of operational hidden constraints, that might impede the original fortification plan.

- Most recent research work done on protecting critical infrastructure (e.g., Yuan et al. (2014), Davarikia & Barati (2018), Davarikia et al. (2020)), especially that on electrical transmission networks, used a column-and-constraint (**C&C**)

generation algorithm or a variation of the Benders algorithm (Wu & Conejo 2017) with no guarantee of reaching optimal solutions. These algorithms are known to require fine tuning of some parameters (e.g., gap between lower-bound and upper-bound, and penalty values) to converge. Such tuning might differ from one electrical network to another, making the available algorithms dependent on the specific networks they were developed for. There is a need for a general-purpose algorithm for protecting electrical transmission networks that can determine all optimal strategies.

- The B&B algorithm would pave the way for efficient and exact solutions methods for protecting electrical transmission networks by offering insights on nodes' characteristics that contain optimal solutions.

Furthermore, in addition to the aforementioned contributions from an application perspective, we make the following contributions from a computational and algorithmic perspective:

- We present a branch-and-bound algorithm with a new branching rule which can be used as a general-purpose bi-level mixed-integer linear programme solver.
- We test our algorithm on a test-bed of randomly generated instances from the literature. We report on computational efficiency in addition to the numbers and types of relaxation problems solved to reach the optimal solution(s).
- We test our algorithm on a specific class of tri-level problems which can be reduced to a mixed-integer bi-level programme.

- To enrich the test bed of bi-level mixed-integer linear problems, we provide a Matlab live editor that converts any electrical transmission network to a bi-level mixed-integer programme instance in the context of defending that network against attacks/disruptions.

In the sequel, we provide necessary background and definitions in Section 4.2. Section 4.3 includes a detailed review of the relevant literature. We introduce definitions, assumptions and algorithmic details of the proposed B&B in Section 4.4. Numerical results are presented in Section 4.5. Finally, we draw conclusions and provide directions for future research in Section 4.6.

4.2 Background

First, let us a consider a generic BMILP which can be defined as follows

$$\max_{\boldsymbol{x},\boldsymbol{y}} \quad c_{\boldsymbol{x}}^T \, \boldsymbol{x} + c_{\boldsymbol{y}}^T \, \boldsymbol{y} \tag{4.1}$$

$$\text{s. t.} \quad G_{\boldsymbol{x}}^T \, \boldsymbol{x} + G_{\boldsymbol{y}}^T \, \boldsymbol{y} \leq q, \tag{4.2}$$

$$x_j \text{ integer } \; \forall \;\; j \in J_{\boldsymbol{x}}, \tag{4.3}$$

$$\boldsymbol{y} \in \arg\max_{\boldsymbol{y}^{'} \in \mathbb{R}^{n_2}} \{ d^T \, \boldsymbol{y}^{'} : A \, \boldsymbol{x} + B \, \boldsymbol{y}^{'} \leq b, \; 0 \leq \boldsymbol{y}^{'} \leq U, \; y_j^{'} \text{ integer} \, \forall \;\; j \in J_{\boldsymbol{y}} \}, \tag{4.4}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_1},\ \boldsymbol{y} \in \mathbb{R}^{n_2}$, while $c_{\boldsymbol{x}}, c_{\boldsymbol{y}}, G_{\boldsymbol{x}}, G_{\boldsymbol{y}}, q, d, A, B, b$, and U are given matrices of appropriate sizes. Furthermore, $J_{\boldsymbol{x}}$ and $J_{\boldsymbol{y}}$ contain the indices of the integer variables in $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. In case of degeneracy in the follower's sub-problem (i.e., multiple optimal solutions), we resolve the degeneracy in favour of the leader, i.e., we adopt an optimistic approach, (Loridan & Morgan 1996), rather than adopting a pessimistic approach, where we resolve degeneracy against the leader's payoffs (Liu et al. 2018).

To aid us in the presentation of our algorithm and relate it to the extant literature, we make the following formal definitions. We start by the feasible regions in Definition 4.2.1.

Definition 4.2.1 (Constraint Regions)**.** *A BMILP has three constraint regions:*

- *The* ***first-level constraint region****:*

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^{n_1} : \ \ G_{\boldsymbol{x}}^T\, \boldsymbol{x} + G_{\boldsymbol{y}}^T\, \boldsymbol{y} \leq q, x_j \text{ integer } \forall \ \ j \in J_{\boldsymbol{x}}\}. \tag{4.5}$$

- *The* ***second-level constraint region****:*

$$\mathcal{Y} = \{\boldsymbol{y} \in \mathbb{R}^{n_2} : \ \ A\, \boldsymbol{x} + B\, \boldsymbol{y} \leq b,\ 0 \leq \boldsymbol{y} \leq U,\ y_j \text{ integer} \forall \ \ j \in J_{\boldsymbol{y}}\}. \tag{4.6}$$

- *The* ***bi-level constraint region****:*

$$\mathcal{S} = \{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}\}. \tag{4.7}$$

Given a feasible leader's decision vector $\boldsymbol{x}$, we are interested in the corresponding bi-level constraint region. Hence, we define the projection of the $\mathcal{S}$ space onto that of the leader in Definition 4.2.2.

Definition 4.2.2 (Projection of bi-level constraint region)**.** *The* ***projection of $\mathcal{S}$ onto the leader's decision space*** *is*

$$\mathcal{S}(\mathcal{X}) = \{\boldsymbol{x} \in \mathcal{X} : \exists \;\; \boldsymbol{y} \textit{ such that } (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}\}. \tag{4.8}$$

For a leader decision vector $\boldsymbol{x} \in \mathcal{S}(\mathcal{X})$, we are interested in finding an optimal follower's response vector. We define the set of such responses in Definition 4.2.3.

Definition 4.2.3 (Rational reaction set)**.** *The* ***rational reaction set*** *of the follower for a fixed leader decision vector $\boldsymbol{x} \in \mathcal{S}(\mathcal{X})$ is*

$$\mathcal{L}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{Y} : \;\boldsymbol{y} \in \underset{\boldsymbol{y}' \in \mathbb{R}^{n_2}}{\arg\max}\{d^T \, \boldsymbol{y}^{'}\}. \tag{4.9}$$

Using Definitions 4.2.1–4.2.3, in Definition 4.2.4 we define the set of possibilities for the leader if given control on all decision variables.

Definition 4.2.4 (Inducible region)**.** *The* ***inducible region*** *is*

$$\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}, \;\; \boldsymbol{y} \in \mathcal{L}(\boldsymbol{x})\}. \tag{4.10}$$

Using the terms in Definitions 4.2.1–4.2.4, the generic BMILP 4.1 can be expressed in a compact form as:

$$\max_{\boldsymbol{x},\boldsymbol{y}} \; c_{\boldsymbol{x}}^T \,\boldsymbol{x} + c_{\boldsymbol{y}}^T \,\boldsymbol{y} \text{ s. t. } \boldsymbol{x}, \boldsymbol{y} \in \mathcal{I}(\boldsymbol{x}, \boldsymbol{y}). \tag{4.11}$$

Next, we define the concepts of BMILP feasibility and optimality in Definitions 4.2.5 and 4.2.6.

Definition 4.2.5 (BMILP feasibilities (Moore & Bard 1990))**.** *A pair of decision vectors* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ *is* ***bi-level feasible*** *if* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in \mathcal{I}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$.

Note that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is bi-level feasible implies that $\hat{\boldsymbol{y}} \in \mathcal{L}(\hat{\boldsymbol{x}})$ and $\hat{\boldsymbol{x}} \in \mathcal{S}(\mathcal{X})$.

Definition 4.2.6 (BMILP Optimality (Moore & Bard 1990))**.** *A pair of decision vectors* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *is* ***bi-level optimal*** *if* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *is bi-level feasible and for all feasible pairs* $(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i) \in \mathcal{I}(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i)$,

$$c_{\boldsymbol{x}}^T \,\boldsymbol{x}^* + c_{\boldsymbol{y}}^T \,\boldsymbol{y}^* \; \geq c_{\boldsymbol{x}}^T \,\hat{\boldsymbol{x}}_i + c_{\boldsymbol{y}}^T \,\hat{\boldsymbol{y}}_i$$

.

Finally, we state BMILP 4.1 in its **marginal-function formulation** (Outrata 1990), also known as the **value-function formulation** (**VF-BMILP**) (Dempe 2002, Dempe et al. 2012):

$$\max_{\boldsymbol{x},\boldsymbol{y}} \qquad c_{\boldsymbol{x}}^T \,\boldsymbol{x} + c_{\boldsymbol{y}}^T \,\boldsymbol{y} \tag{4.12}$$

$$\text{s.t.} \qquad G_{\boldsymbol{x}}^T\,\boldsymbol{x} + G_{\boldsymbol{y}}^T\,\boldsymbol{y} \leq q, \tag{4.13}$$

$$A\,\boldsymbol{x} + B\,\boldsymbol{y} \leq b, \tag{4.14}$$

$$0 \leq \boldsymbol{y} \leq U \tag{4.15}$$

$$x_j \text{ integer } \forall\ j \in J_{\boldsymbol{x}}, \tag{4.16}$$

$$y_j \text{ integer } \forall\ j \in J_{\boldsymbol{y}}, \tag{4.17}$$

$$d^T\,\boldsymbol{y} \geq \phi(\boldsymbol{x}), \tag{4.18}$$

where for a given feasible leader's decision vector, $\boldsymbol{x} \in \mathcal{S}(\mathcal{X})$, the value-function for the follower's problem can be calculated as $\phi(\boldsymbol{x}) = \max\limits_{\boldsymbol{y}\in\mathcal{L}(x)} d^T\,\boldsymbol{y}$. Dropping the constraint 4.18 in VF-BMILP will lead to the high-point problem (**HPP**) relaxation as first defined by Bialas & Karwan (1984), and then used by Moore & Bard (1990). There has been different versions of the HPP which might differ depending on the algorithm developed. For instance, (Bialas & Karwan (1984), Moore & Bard (1990), Fischetti et al. (2017)) used the same version of the HPP, while (Xu & Wang (2014), Liu et al. (2021)) used a different version. In the next section we will clearly state our version for the HPP, and its relaxations.

4.3 Literature Review

We briefly review the relevant literature to bi-level programming, and the special class of tri-level programming that is under consideration.

4.3.1 Bi-Level programmes

Motivated by the need for modelling decentralized planning, the work done by Bialas & Karwan (1984) motivated the need for efficient and tractable algorithms for solving the bi-level problem. Moore & Bard (1990) established that it is not possible to obtain tight upper-bounds from the natural relaxation. By providing examples and toy problems, they established that two of the three well known fathoming rules used in B&B in single-level mixed-integer programming cannot be used in BMILP. In their influential work, Moore & Bard (1990) provided an implicit enumeration technique for finding bi-level feasible solutions, and a series of heuristics that offer a trade-off between quality and efficiency. In (Bard & Moore 1990), a B&B approach was suggested that makes use of exploiting the follower's Karush-Kuhn-Tucker (**KKT**) conditions; the algorithm enforces the underlying complementary slackness conditions suggested by Fortuny-Amat & McCarl (1981). A special algorithm developed for binary decision variables for the leader that affects real-valued decision variables for the follower has been proposed by Wen & Yang (1990) where exact and heuristic solutions were provided for a special class of linear bi-level programmes. Moreover, Saharidis & Ierapetritou (2009) presented a decomposition approach for BMILP in which they used a variation of Benders decomposition approach. The master- and sub-problems are designed as a relaxation and restriction of BMILP, where they interact by adding cuts using Lagrangian information from the current sub-problem. Zeng & An (2014) also presented a computing scheme based on a decomposition strategy by converting BMILP into a single-level reformulation and using a column-and-constraint generation algorithm. Xu & Wang (2014) presented an exact B&B

algorithm with three simplifying assumptions for tractability: 1) all variables in the decision vector are required to be integral, 2) variables are bounded from below and above, and 3) the coefficient matrix A in BMILP 4.4 is integral. Kleniati & Adjiman (2014) presented an algorithm called branch-and-sandwich, in which two solution spaces corresponding to the first- and second-levels, are explored using a single B&B tree. In particular, two pairs of upper- and lower- bounds are computed: one for the objective function of the leader, and the other pair is for the follower's objective value. Motivated by recent efforts at that time, Fischetti et al. (2017) suggested a new branch-and-cut (**B&C**) algorithm for BMILP, in which they provided specific pre-processing strategies, valid linear inequalities, along with separation procedures. Recently, Tahernejad et al. (2020) presented a generalized B&C algorithmic framework for solving BMILPs; in which features from single-level and bi-level algorithms are combined. The aim was to produce a flexible and robust framework for solving a variety of different BMILPs. Furthermore, based on the fact that B&C has proven to be more powerful than B&B in single-level mixed-integer optimization problems, Kleinert, Labbé, Plein & Schmidt (2021) were motivated to review existing cuts for linear bi-level problems and introduced a new valid inequality that examines the strong duality constraint of the follower's level, and strengthened variants of the inequality derived from McCormick envelopes. Most recently, Liu et al. (2021) presented an enhanced branching rule based on the algorithm developed by Xu & Wang (2014). However, the new branching rule might discard bi-level feasible solutions if the lower-level problem possesses alternative optima, which may in-turn lead to bi-level feasibility (i.e., sub-optimality in BMILP).

From the perspective of bi-level integer linear programmes (**BILP**s), Bard (2013) presented an algorithm for the binary case for both leader and follower decision variables. This was achieved by converting the leader's objective function to a parametrized constraint and solving the re-formulated problem to produce a bi-level feasible solution. After which, improvements are gradually sought leading to the global optimum. DeNegre & Ralphs (2009) proposed a B&C approach for BILP, which improves on the B&B approach proposed by Bard & Moore (1990) by adding cutting planes to provide tighter bounds. It is worth mentioning that this approach does not require special branching strategies and was implemented through publicly available linear solvers. Furthermore, using almost the same branching rules stated in (Xu & Wang 2014), but taking advantage of the integer requirements in BILP, Wang & Xu (2017) proposed the watermelon algorithm, in which a polyhedron is formed to encapsulate bi-level infeasible solutions. The complement of this polyhedron is then taken as disjunction hyperplanes in a B&B framework. Indeed, the area of including cuts or valid inequalities to bi-level programmes is a fertile area for research. Influential papers that discuss the application of including valid inequalities and cuts include (Fischetti et al. 2016) and (Fischetti et al. 2018) that are based on the idea of convexity cuts (Balas (1971), Glover (1973, 1974)).

Other approaches relying on parametric programming algorithms to solve bi-level quadratic and BMILPs have been proposed in the literature. Faísca et al. (2007) substituted the rational reaction sets, $\mathcal{L}(\boldsymbol{x})$, in the leader's problem and transformed the bi-level problem into a set of independent quadratic, linear or mixed-integer linear problems that can be solved to optimality. Mitsos (2010) proposed an algorithm for the global optimization of non-linear bi-level mixed-integer programmes where it relies on a lower-bound obtained by solving mixed-integer non-linear programmes and a parametric upper-bound to the optimal solution function of the lower-level programme.

The majority of the research done on bi-level programming deals with the optimistic case, i.e., in case of a non-unique rational response (e.g., maximizes/minimizes payoffs) for the follower, the strategy that is in favour of the leader would be chosen. From a game-theory perspective, this is known as a strong Stackelberg game (Breton et al. 1988). On the other hand, if the follower picks a strategy that is against the leader's payoffs, this is considered a weak Stackelberg game (Loridan & Morgan 1996) which corresponds to a pessimistic two-level optimization problem (Dempe 2002, Liu et al. 2018). It should be noted that Leitmann (1978) introduced the concept of a generalized Stackelberg game, accounting for non-unique followers' responses, after which Breton et al. (1988) introduced a formal definition for the strong-weak Stackelberg games. Furthermore, obtaining the optimality conditions for bi-level linear programming problem has been discussed in the literature under the assumption of uniqueness (Bard 1984), optimistic (Dempe et al. 2006, Gadhi & Dempe 2012), and pessimistic (Dempe et al. 2014) approaches. Bi-level programmes are often re-formulated using KKT conditions for the lower-level problem, if it is a parametric convex optimization problem, resulting in a single-level mathematical programme with complementary slackness conditions. The question of equivalence of both programmes has been discussed in (Dempe & Dutta 2012) and showed that it depends on the existence of Slater's constraint qualification for the lower-level problem for the optimistic approach. The work done by Aussel & Svensson (2019) discusses the equivalence in the pessimistic approach.

For a comprehensive review of bi-level programming, solution approaches and practical applications, the interested reader may refer to the following reviews:

- Wen & Hsu (1991) recaps basic models, applications, solution approaches for the linear bi-level programming problems.
- Ben-Ayed (1993) gives a review of the features of linear bi-level programmes, applications, algorithms and clarifies some confusing representations in the literature.
- Dempe (2003) provides some main direction of research highlighting re-formulated bi-level programmes with complementary slackness conditions, difficulties arising from non-uniqueness of followers' optimal solutions, and on optimality conditions.
- The work done in (Colson et al. 2005, 2007) gives an introductory survey of bi-level programmes motivated by simple applications, main properties of different cases (e.g., linear-quadratic), and an overview of solution approaches.
- Lu et al. (2016) review multi-level decision-making with a focus on bi-level programming.
- Liu et al. (2018) review the definitions, properties of the pessimistic bi-level optimization approach and follows up with a discussion on solution approaches

and some practical applications.

- Kleinert, Labbé, Ljubić & Schmidt (2021) review bi-level algorithmic approaches that make use of mixed-integer programming techniques with a focus on linear lower-level problems. They provide a review on solution approaches for mixed-integer bi-level problems with integer constraints in the follower's level.

4.3.2 A Class of Tri-Level programmes

Our proposed B&B approach can solve a class of tri-level linear programmes (**TLP**s); those that can be reduced to an equivalent BMILPs. Such TLPs have a convex linear problem in their third-level that satisfies Slater's constraint qualification (Dempe & Dutta 2012). The re-formulation of TLPs into an equivalent BMILP can be done using two approaches: 1) the duality approach that adds primal and dual constraints, and a strong duality constraint of the third-level problem to the second-level; and 2) the KKT approach that involves adding the KKT conditions as constraints in the second-level problem. It is worth mentioning that the former approach might be computationally superior over the latter; due to the systematically introduced bilinear terms from the complementary slackness conditions. A proof of the equivalence of both approaches for TLPs can be found in the Appendix of (Fakhry et al. 2022).

4.3.3 Contributions

The class of tri-level programmes discussed in (Fakhry et al. 2022) is only a subclass of the TLPs that our proposed B&B approach can handle. In particular, our algorithm can handle mixed-integer variables with linear constraints in both first- and second- levels, as long as the third-level has continuous decision variables with linear constraints. Furthermore, each decision-level can have its own objective function (i.e., different agendas), unlike (Fakhry et al. 2022) where the proposed heuristics can handle only shared objectives that are a function of the third-level decision variables. Nevertheless, since the application of defending electrical transmission networks (i.e., the defender-attacker-defender model) is a sub-class of TLPs that our proposed approach can tackle, we validate the results obtained in (Fakhry et al. 2022). Moreover, with a focus on the defender-attacker-defender (**DAD**) model, we outline how our proposed approach differs from other recently developed algorithms in Table 4.1. In particular, we capitalize on the nature of the B&B structure in terms of providing a guarantee of bi-level feasibility, and optimality according to Definitions 4.2.5, and 4.2.6. It is worth mentioning that all research works mentioned in the first column of Table 4.1 handle a TLP that can be reduced to BMILP through the aforementioned methods, thus we call the reduced TLP as a bi-level programme. All research works in Table 4.1 provide optimal/sub-optimal (i.e., bi-level feasible) solutions very efficiently with no guarantee or proof of bi-level optimality. This is due to the fact that they rely on efficient decomposition approaches that resembles Benders decomposition (Wu & Conejo 2017), or C&C generation (Davarikia et al. 2020, Xiang et al. 2020, Davarikia & Barati 2018, Yuan et al. 2014). These approaches are known to require fine tuning that might differ from an electric transmission network to another.

Not only, does our proposed approach provide a guarantee of bi-level optimality,

Research Work	Bi-level Feasibility	Bi-level Optimality	Alternative Optima	General Solver
Yuan et al. (2014)	✓	✗	✗	✗
Wu & Conejo (2017)	✓	✗	✗	✗
Davarikia & Barati (2018)	✓	✗	✗	✗
Xiang et al. (2020)	✓	✗	✗	✗
Davarikia et al. (2020)	✓	✗	✗	✗
Fakhry et al. (2022)	✓	✗	✗	✗
Proposed Approach	✓	✓	✓	✓

Table 4.1: Contributions from an Application Perspective.

it outlines alternative optimal strategies (if they exist) that the defender can utilize in case there are hidden operational constraints impeding the fortification of a specific hardening strategy. Moreover, our proposed approach is generalized to solve TLPs that are reducible to BMILPs. Thus, it can be easily tuned to accommodate different objective functions for the leader and follower in addition to constraints having follower decision variables in the first-level problem. Furthermore, we develop a Matlab live editor in which the user can simply input a comma-separated values (**CSV**) file, with the attributes of an electrical transmission network, to obtain as output the DAD model in a BMILP format ready to be fed to any general-purpose BMILP solver. More details on the publicly available Matlab live editor are mentioned in Section 4.5.

4.4 Algorithm

In this section we provide details on the algorithm including assumptions, definitions and sub-procedures.

4.4.1 Assumptions

We make the following assumptions about the generic BMILP 4.1:

Assumption 4.4.1 (Boundedness)**.** *The leader decision vector, $\boldsymbol{x}$, is bounded and can have continuous variables as long as they do not have a direct effect on the follower's optimal reaction set, $\mathcal{L}(\boldsymbol{x})$.*

The boundedness assumption 4.4.1 ensures finite termination of the algorithm. The existence of the leader's continuous variables would interfere with the branching rule, which potentially might lead to discarding bi-level feasible solutions. For a similar reason, in Assumption 4.4.2, we enforce the integrality of the followers constraint matrix coefficients.

Assumption 4.4.2 (Integrality)**.** *Matrix A in the follower's problem* 4.4 *consists of integer inputs.*

Having non-integer inputs for matrix A will affect the branching rule, potentially leading to having open sets, missing bi-level feasible, and possibly optimal solutions.

4.4.2 Definitions

We introduce definitions for the B&B algorithm to facilitate the presentation of the main procedures in the algorithm.

Definition 4.4.1 (B&B sub-problem, $\mathcal{B}(l, u, w)$)**.** *Each node in the B&B algorithm represents a sub-problem, $\mathcal{B}(l, u, w)$, that is a parametric BMILP with specific bounds*

(l, u, w) *and can be defined as follows:*

$$\begin{aligned}
\max_{\boldsymbol{x},\boldsymbol{y}} \quad & c_{\boldsymbol{x}}^T\, \boldsymbol{x} + c_{\boldsymbol{y}}^T\, \boldsymbol{y} && (4.19)\\
\text{s. t.} \quad & G_{\boldsymbol{x}}^T\, \boldsymbol{x} + G_{\boldsymbol{y}}^T\, \boldsymbol{y} \le q, && (4.20)\\
& A\, \boldsymbol{x} + B\, \boldsymbol{y} \le b, && (4.21)\\
& l \le A\, \boldsymbol{x} \le u, && (4.22)\\
& d^T\, \boldsymbol{y} \ge w, && (4.23)\\
& x_j \textit{ integer } \forall\ \ j \in J_{\boldsymbol{x}}, && (4.24)\\
& \boldsymbol{y} \in \mathcal{L}(\boldsymbol{x}). && (4.25)
\end{aligned}$$

It is worth mentioning that $\mathcal{B}(l, u, w)$ is equivalent to the generic BMILP 4.1 for $(l = -\infty, u = \infty, w = -\infty)$. Moreover, note that problem $\mathcal{B}(l, u, w)$ resembles VF-BMILP 4.12 except for constraints 4.22, and 4.23. These two constraints, that have the parameters (l, u, w), will be later used for branching children nodes from the parent node.

Next, we introduce our definition for the HPP, which is a combination of the VF-BMILP 4.12 and the HPP version used by (Moore & Bard 1990).

Definition 4.4.2 (High-point problem, HPP)**.** *The HPP, $\mathcal{H}(l, u, w)$, is a relaxation of the parametric BMILP node problem $\mathcal{B}(l, u, w)$. In particular, the follower's decision vector, $\boldsymbol{y}$), belongs to the second-level constraint region, $\mathcal{Y}$, instead of the optimal reaction set, $\mathcal{L}(\boldsymbol{x})$, as follows:*

$$\max_{\boldsymbol{x},\boldsymbol{y}} \qquad c_{\boldsymbol{x}}^T\, \boldsymbol{x} + c_{\boldsymbol{y}}^T\, \boldsymbol{y} \qquad (4.26)$$

$$\text{s.t.} \quad G_{\boldsymbol{x}}^T \, \boldsymbol{x} + G_{\boldsymbol{y}}^T \, \boldsymbol{y} \leq q, \tag{4.27}$$

$$A \, \boldsymbol{x} + B \, \boldsymbol{y} \leq b, \tag{4.28}$$

$$l \leq A \, \boldsymbol{x} \leq u, \tag{4.29}$$

$$d^T \, \boldsymbol{y} \geq w, \tag{4.30}$$

$$x_j \; integer \; \forall \; j \in J_{\boldsymbol{x}}, \tag{4.31}$$

$$\boldsymbol{y} \in \mathcal{Y}. \tag{4.32}$$

It should be noted that our Definitions 4.4.1 and 4.4.2 were inspired by the work done by Xu & Wang (2014). However, we allow for the presence of continuous variables in the leader's decision vector $\boldsymbol{x}$, as long as it satisfies Assumption 4.4.1. According to Lemmas 5–8 in Xu & Wang (2014), if the parametric HPP relaxation $\mathcal{H}(l, u, w)$ is unbounded, then the corresponding parametric node problem $\mathcal{B}(l, u, w)$ can be infeasible, unbounded, or have a finite optimal solution based on the solution of a mixed-integer linear programme (**MILP**) to find the cause of the unboundedness, as defined in 4.4.3.

Definition 4.4.3 (Unboundedness MILP-$\mathcal{U}$)**.** *The unboundedness MILP-$\mathcal{U}$ must have a feasible solution, if the parametric HPP $\mathcal{H}(l, u, w)$ is unbounded and can be defined as follows:*

$$\max_{\Delta \boldsymbol{y}} \quad d^T \, \Delta \boldsymbol{y} \tag{4.33}$$

$$\text{s.t.} \quad G_{\boldsymbol{y}}^T \, \Delta \boldsymbol{y} \leq 0, \tag{4.34}$$

$$B \, \Delta \boldsymbol{y} \leq 0, \tag{4.35}$$

$$\Delta \boldsymbol{y} \geq 0, \tag{4.36}$$

$$\Delta y_j \textit{ integer } \forall \; j \in J_{\boldsymbol{y}}. \tag{4.37}$$

It is clear from the definition of MILP-$\mathcal{U}$ that it searches to find the optimum value of an extreme ray $\Delta \boldsymbol{y}$, hence MILP-$\mathcal{U}$ would have a feasible solution if and only if the parametric HPP $\mathcal{H}(l, u, w)$ is unbounded. Moreover, since MILP-$\mathcal{U}$ is independent of the node parameters (l, u, w) and is a function of the generic BMILP instance parameters, we do not need to do this check more than once for each instance. If MILP-$\mathcal{U}$ has a feasible solution, then the resolution of the unboundedness of the parametric HPP, $\mathcal{H}(l, u, w)$, stems from the lower-level problem (**LLP**) 4.4, due to Assumption 4.4.1. In particular, depending on the objective value direction of MILP-$\mathcal{U}$, the generic BMILP 4.1 can be unbounded if $d^T \ \Delta \boldsymbol{y} = 0$, have a finite optimal solution if $d^T \ \Delta \boldsymbol{y} < 0$, or infeasible if $d^T \ \Delta \boldsymbol{y} > 0$ because LLP is unbounded.

4.4.3 Algorithm Main Procedures

We provide a general overview of the B&B algorithm by introducing its four main procedures:

- **Initialization procedure**: we start by loading the parameters of the BMILP instances (i.e., matrices for the generic BMILP 4.1), followed by creating the root node and initializing bounds, (l, u, w). Furthermore, we initialize repositories for reporting on the types and numbers of nodes explored, number and type of relaxation problems solved, and finally the overall bi-level instance type,i.e., optimal, infeasible, or unbounded.

- **Checking nodes procedure**: this function evaluates waiting nodes to be explored, eliminates nodes if certain conditions are met, and reports the final bi-level instance type.

- **HPP procedure**: this procedure invokes a mixed-integer linear programme solver for solving a parametric HPP, $\mathcal{H}(l, u, w)$, and then evaluates the output to determine the next steps depending on the solution status of the solver.

- **LLP procedure**: this function invokes a mixed-integer linear programme solver for solving LLP 4.4, the follower's problem. Consequently, determining an optimal reaction response for the follower, $\boldsymbol{y} \in \mathcal{L}(\boldsymbol{x})$, and proceed to the branching rule if specific conditions are met.

The algorithm overview is presented in the flowchart in Figure 4.1. In the remainder of this section, we provide more details on each of the procedures and the flow between them.

For each BMILP instance, we have the input matrices in 4.1- 4.4. We denote ψ^* as the bi-level optimal objective value 4.1 and $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ as the corresponding bi-level optimal pair. For each BMILP instance, we report the triplet $(\boldsymbol{x}^*, \boldsymbol{y}^*, \psi^*)$. We denote the number of nodes waiting in the repository N and the flag for unboundedness F. This flag is set to 1 if the unboundedness MILP-$\mathcal{U}$ is solved, and zero otherwise. For each BMILP instance, the B&B algorithm creates nodes through the branching rule to search for feasible/optimal solutions (if they exist) resulting in (l^k, u^k, w^k) for each node problem $k \in \{1, \ldots, N\}$. Additionally, for each parametric node problem $\mathcal{B}(l, u, w)$, we store the following data: parent node number, node ID,

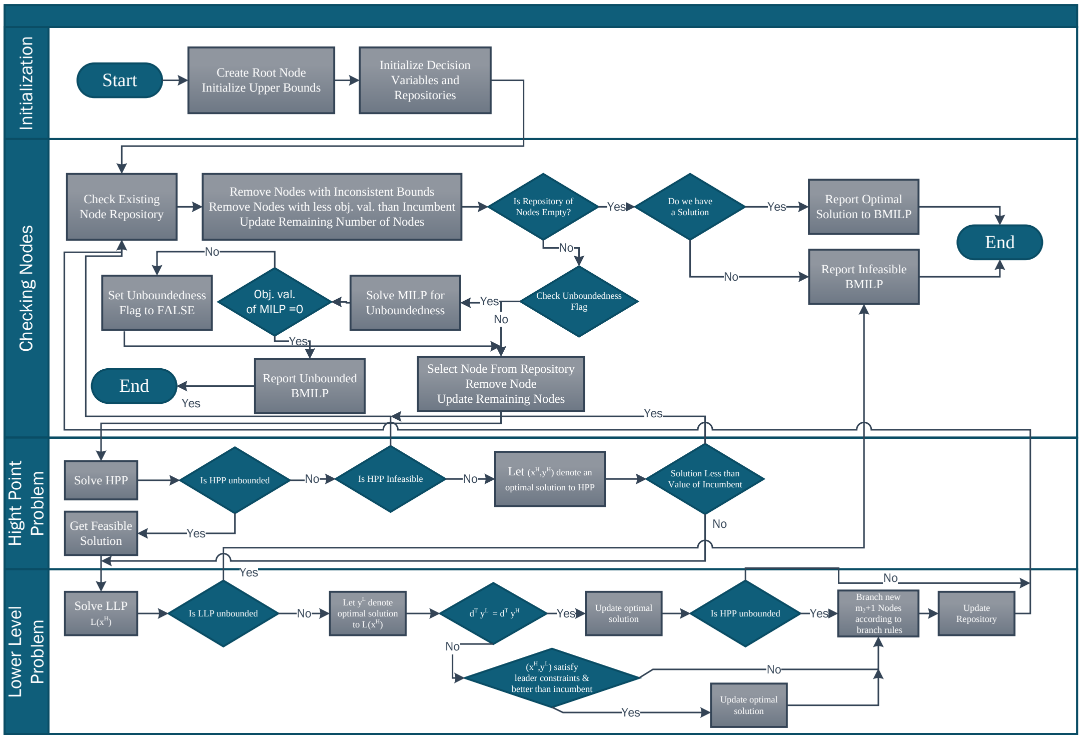

Initialization
Start
Create Root Node
Initialize Upper Bounds
Initialize Decision Variables and Repositories
Checking Nodes
Check Existing Node Repository
Remove Nodes with Inconsistent Bounds
Remove Nodes with less obj. val. than Incumbent
Update Remaining Number of Nodes
Is Repository of Nodes Empty?
Yes
Do we have a Solution
Yes
Report Optimal Solution to BMILP
End
No
Report Infeasible BMILP
No
Check Unboundedness Flag
Yes
No
Solve MILP for Unboundedness
Obj. val. of MILP =0
No
Set Unboundedness Flag to FALSE
Yes
Report Unbounded BMILP
End
Yes
Select Node From Repository
Remove Node
Update Remaining Nodes
Yes
Hight Point Problem
Solve HPP
Is HPP unbounded
No
Is HPP Infeasible
No
Let (x^H,y^H) denote an optimal solution to HPP
Solution Less than Value of Incumbent
No
Yes
Get Feasible Solution
Yes
Lower Level Problem
Solve LLP $L(x^H)$
Is LLP unbounded
No
Let y^L denote optimal solution to $L(x^H)$
$d^T y^L = d^T y^H$
Yes
Update optimal solution
Is HPP unbounded
Yes
No
Branch new m_2+1 Nodes according to branch rules
Update Repository
No
(x^H,y^L) satisfy leader constraints & better than incumbent
No
Yes
Update optimal solution

Figure 4.1: B&B Algorithm Chart.

node status, parent node objective value p^{obj}, the bounds for each node (l^k, u^k, w^k), output from the parametric HPP problem $(\boldsymbol{x}^H, \boldsymbol{y}^H, z^H)$ (if it exists), output from solving LLP$(\boldsymbol{x}^H)$ 4.4:$(\boldsymbol{y}^L, z^L)$ and LLP problem status (if applicable).

4.4.4 Checking Nodes Procedure

The flow chart for this procedure is depicted in Figure 4.2. It is a critical part of the algorithm where we assign parametric node problem $\mathcal{B}(l^k, u^k, w^k)$ to the HPP procedure. Depending on the output of the HPP procedure, the node problem might be discarded or sent to the LLP procedure for further branching. This can be seen

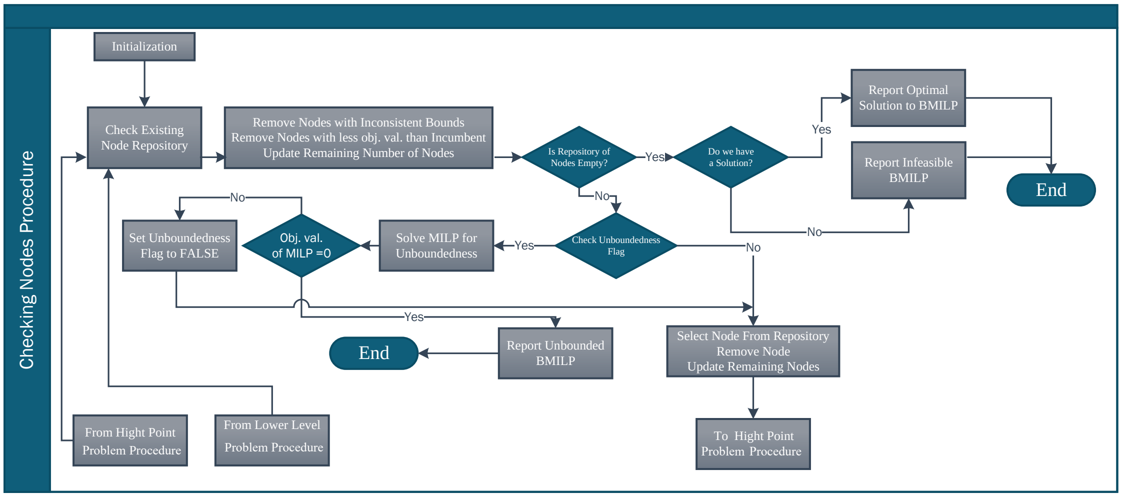

Checking Nodes Procedure
Initialization
Check Existing Node Repository
Remove Nodes with Inconsistent Bounds
Remove Nodes with less obj. val. than Incumbent
Update Remaining Number of Nodes
Is Repository of Nodes Empty?
Yes
Do we have a Solution?
Yes
Report Optimal Solution to BMILP
Report Infeasible BMILP
End
No
No
Check Unboundedness Flag
Yes
No
Solve MILP for Unboundedness
Obj. val. of MILP =0
No
Set Unboundedness Flag to FALSE
Yes
Report Unbounded BMILP
End
Select Node From Repository
Remove Node
Update Remaining Nodes
To Hight Point Problem Procedure
From Hight Point Problem Procedure
From Lower Level Problem Procedure

Figure 4.2: Checking and Updating Nodes in Repository.

at the bottom-left of Figure 4.2. For any BMILP instance, the process starts with root node $\mathcal{B}(l^1, u^1, w^1)$ and initialized bounds, $l^1 = -\infty$ and $u^1 = w^1 = \infty$, that are then fed into the HPP and LLP procedures for branching. The process continues with checking nodes in the repository. Each node k is characterized with three parameters (l^k, u^k, w^k) obtained from branching. Checking nodes procedure removes nodes with inconsistent bounds (i.e., $l_j^k > u_j^k$) for any constraint j in the constraint set 4.22. Another check occurs on the parent node objective value p^{obj} to see if any nodes can be discarded depending on the incumbent solution (i.e., best solution found so far). This is followed up with updating the number of remaining nodes in the repository. If the number of remaining nodes $N = 0$ and we have a solution (i.e., $(\boldsymbol{x}^*, \boldsymbol{y}^*, \psi^*)$), we report the optimal solution to the current BMILP instance. Otherwise, we report infeasibility. However, if the repository of nodes is not empty, we check the unboundedness of the BMILP instance by solving the unboundedness

MILP-$\mathcal{U}$ and set the unboundedness flag to 1 for that BMILP instance. Thus, no computational time is wasted on resolving problems. If the objective value of MILP-$\mathcal{U}$ is equal to zero, i.e., $d^T \ \Delta \boldsymbol{y} = 0$, then we report the unboundedness of the BMILP instance and set the unboundedness flag to true. Otherwise, we set the unboundedness flag to false and proceed to select a node from the repository, remove the node from the waiting list, and update the number of remaining nodes N. The selected node with its corresponding parameters (l^k, u^k, w^k) is passed to the HPP procedure.

4.4.5 High Point Problem Procedure

We proceed by presenting a crucial proposition for the HPP procedure.

Proposition 4.4.1. *Assuming an optimal solution* $(\boldsymbol{x}^H, \boldsymbol{y}^H, z^H)$ *has been found for the parametric HPP* $\mathcal{H}(l, u, w)$*, then* $\mathcal{B}(l, u, w)$ *will have an optimal solution* $(\boldsymbol{x}^B, \boldsymbol{y}^B, z^B)$ *such that* $z^H \geq z^B$*, i.e.,* z^H *is an upper-bound for* $\mathcal{B}(l, u, w)$*.*

Proof. The definition of parametric node problem $\mathcal{B}(l, u, w)$ includes constraint 4.25 which necessitates that $\boldsymbol{y}$ has to be in the rational reaction set, i.e., $\boldsymbol{y} \in \mathcal{L}(\boldsymbol{x})$), while the corresponding parametric HPP $\mathcal{H}(l, u, w)$ has exactly the same definition except for constraint 4.32, which is a relaxation of 4.25 since $\boldsymbol{y} \in \mathcal{Y}$, the second-level constraint region. Since both are maximization problems, $\mathcal{H}(l, u, w)$ has an optimal solution, call it $(\boldsymbol{x}^B, \boldsymbol{y}^B, z^B)$, and since it is a relaxation of $\mathcal{B}(l, u, w)$ it follows that $z^H \geq z^B$. □

Figure 4.3 shows the steps in the HPP procedure. It is invoked from the checking nodes procedure, where one of the waiting nodes in the repository is picked. Since each node is represented by parametric node problem $\mathcal{B}(l, u, w)$, the parameters are passed to $\mathcal{H}(l, u, w))$ where a mixed-integer linear programme is solved.

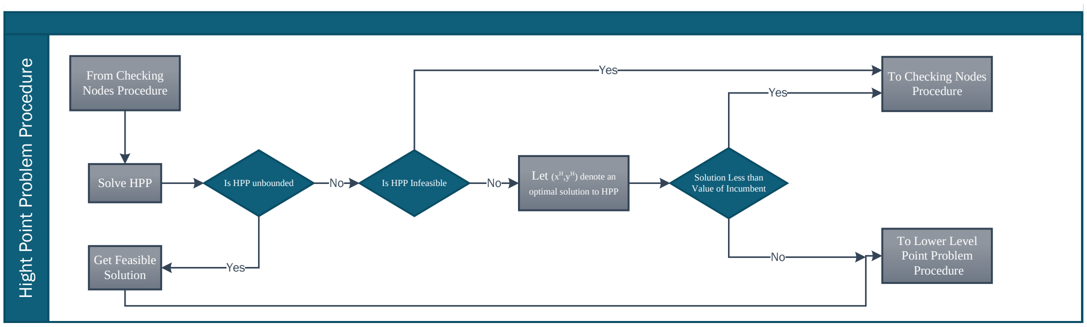

Hight Point Problem Procedure
From Checking Nodes Procedure
Solve HPP
Is HPP unbounded
No
Is HPP Infeasible
No
Let (x^H,y^H) denote an optimal solution to HPP
Solution Less than Value of Incumbent
Yes
Yes
To Checking Nodes Procedure
Yes
Get Feasible Solution
No
To Lower Level Point Problem Procedure

Figure 4.3: High Point Problem Procedure in B&B Algorithm.

Depending on the outcome from the solver's solution status, the next steps are determined:

- If $\mathcal{H}(l, u, w)$ is unbounded, then further examination is required by passing $\boldsymbol{x}^H$ to the LLP procedure. This particular node problem is marked as "Explored."
- If $\mathcal{H}(l, u, w)$ is infeasible, then this particular node problem is marked as "Explored" and "Infeasible."
- Otherwise, an optimal solution is found for HPP. The output is stored and compared to the best solution found so far (if applicable). If $z^H \geq \psi^*$, then there is a room for improvement or finding alternative optima and we proceed with branching by passing $\boldsymbol{x}^H$ to the LLP procedure. This node is marked as "Explored" and "Integer Optimal." Otherwise, if $z^H < \psi^*$, then there is no

room for improvement from this node and it is marked as "Better obj. Value Found."

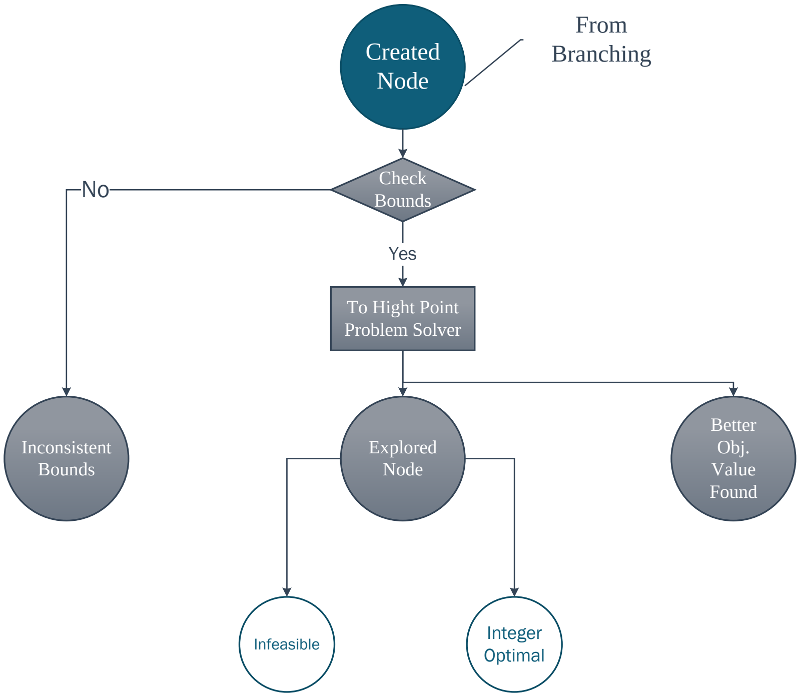


Figure 4.4: Node Labelling in B&B Algorithm.

Figure 4.4 shows the different labelling for nodes in the B&B algorithm. First the node is created from branching, unless it is the root node, in the LLP procedure and assigned a triplet of bounds (l^k, u^k, w^k), depending on the branching rule. The checking nodes procedure verifies the consistency of the bounds. In particular, if $l^k > u^k$ then the node is marked as "Inconsistent Bounds," and the solver is not invoked, which means that the node is removed from repository and not passed to the HPP procedure. Otherwise, if the bounds are consistent, the node problem is passed to the HPP procedure where the solver is invoked. The node is then marked as either “Explored,” with an additional tag of being “Infeasible,” “Integer Optimal”, or the incumbent solution was better marking it as “Better Obj. Value Found.”

4.4.6 LLP Procedure

Figure 4.5 depicts the LLP procedure. It is invoked from the HPP procedure block in two cases: 1) if the HPP problem is found unbounded, there might be a possibility that the corresponding node problem might have a finite optimal solution, and 2) if an integer optimal solution was found from HPP and there is a room for improvement and so we proceed to branching Balas (1971) in search for a better objective value.

In Proposition 4.4.2, we present an important result for finding feasible and optimal bi-level solutions.

Proposition 4.4.2. *Given an optimal solution, $(\boldsymbol{x}^H, \boldsymbol{y}^H, z^H)$, from the HPP procedure $\mathcal{H}(l, u, w)$, the LLP procedure admits bi-level feasible solutions.*

Proof. From the definition of HPP, the leader’s objective value and constraints are met optimally. We solve LLP$(\boldsymbol{x}^H)$ 4.4 to get the follower’s decision vector $\boldsymbol{y}^L$ which belongs to the optimal reaction set for the follower, $\boldsymbol{\mathcal{L}}(\boldsymbol{x}^H)$. We end up with three options depending on $\boldsymbol{y}^L$:

- if $d^T\ \boldsymbol{y}^L = d^T\ \boldsymbol{y}^H$, we have a bi-level feasible solution and possibly optimal depending on the value of the incumbent.
- Otherwise, we check if $\boldsymbol{y}^L$ satisfies the leader constraints, and if so, we have a bi-level feasible solution and possibly optimal depending on the value of the incumbent.

- If $\boldsymbol{y}^L$ does not satisfy the leader constraints, then we proceed with branching in search of a better objective value.

□

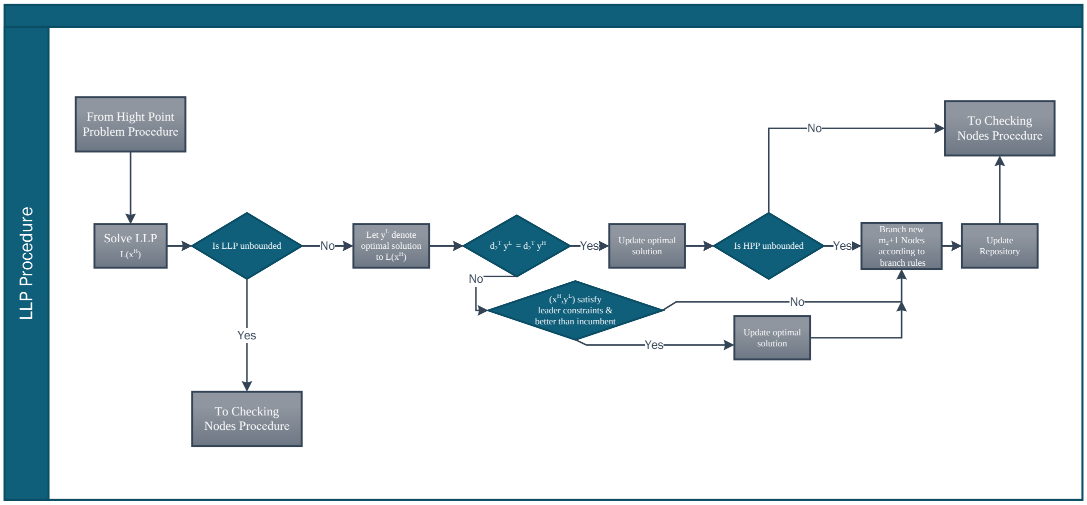

LLP Procedure
From Hight Point Problem Procedure
Solve LLP $L(x^H)$
Is LLP unbounded
No
Yes
To Checking Nodes Procedure
Let y^L denote optimal solution to $L(x^H)$
$d_2^T y^L = d_2^T y^H$
Yes
No
Update optimal solution
Is HPP unbounded
Yes
No
To Checking Nodes Procedure
Branch new m_2+1 Nodes according to branch rules
Update Repository
(x^H,y^L) satisfy leader constraints & better than incumbent
No
Yes
Update optimal solution

Figure 4.5: Lower Level Problem Procedure in B&B Algorithm.

4.4.7 Branching Rules

Branching is included in the LLP procedure. When the LLP procedure is invoked after solving the HPP we obtain a feasible solution $(\boldsymbol{x}^H, \boldsymbol{y}^H)$, if the HPP was found to be unbounded, or an optimal solution to HPP $(\boldsymbol{x}^H, \boldsymbol{y}^H)$ that has a value better than the incumbent.

Figure 4.6 depicts the interactions between HPP, LLP, and Branching rules. First, a solution $(\boldsymbol{x}^H, \boldsymbol{y}^H)$ is obtained from the HPP procedure. The vector $\boldsymbol{x}^H$ is then passed to the LLP to obtain $\boldsymbol{y}^L \in \mathcal{L}(\boldsymbol{x^H})$. It is worth mentioning that solving the

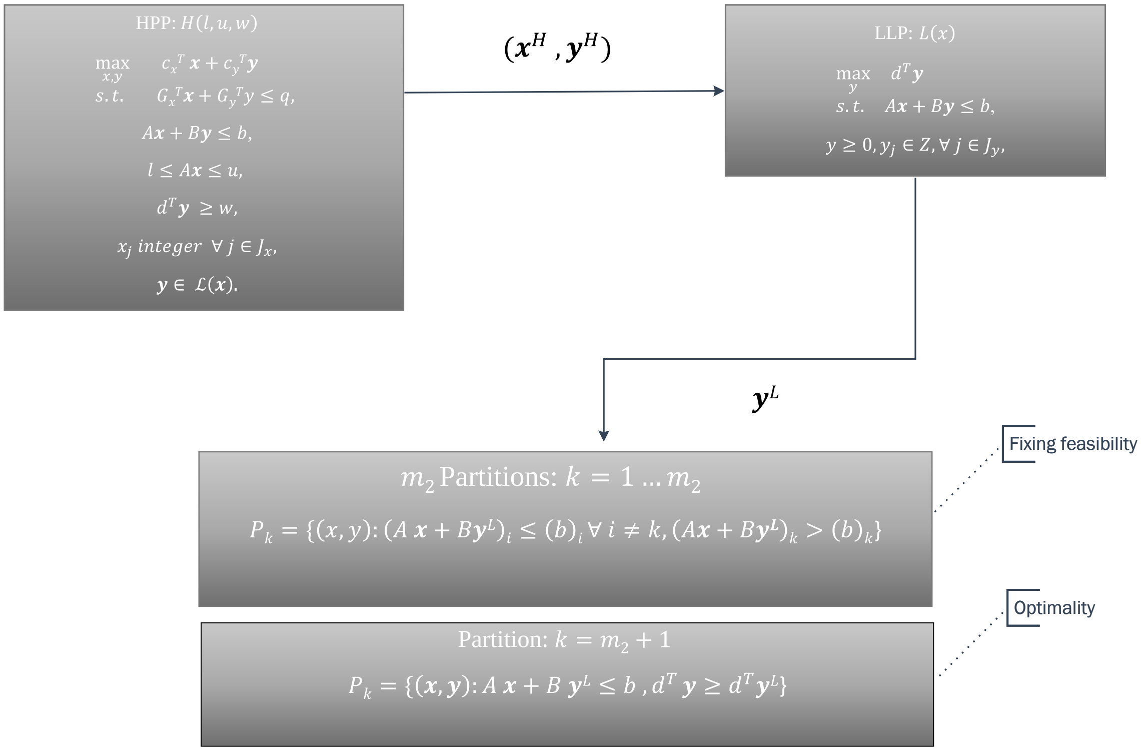

HPP: $H(l,u,w)$
$\max_{x,y}$ $c_x^T \boldsymbol{x} + c_y^T \boldsymbol{y}$
$s.t.$ $G_x^T \boldsymbol{x} + G_y^T y \leq q,$
$A\boldsymbol{x} + B\boldsymbol{y} \leq b,$
$l \leq A\boldsymbol{x} \leq u,$
$d^T \boldsymbol{y} \geq w,$
x_j $integer$ $\forall j \in J_x,$
$\boldsymbol{y} \in \mathcal{L}(\boldsymbol{x}).$
$(\boldsymbol{x}^H, \boldsymbol{y}^H)$
LLP: $L(x)$
$\max_{y}$ $d^T \boldsymbol{y}$
$s.t.$ $A\boldsymbol{x} + B\boldsymbol{y} \leq b,$
$y \geq 0, y_j \in Z, \forall j \in J_y,$
$\boldsymbol{y}^L$
m_2 Partitions: $k = 1 \ldots m_2$
$P_k = \{(x,y): (A\,\boldsymbol{x} + B\boldsymbol{y}^L)_i \leq (b)_i \forall\, i \neq k, (A\boldsymbol{x} + B\boldsymbol{y}^{\boldsymbol{L}})_k > (b)_k\}$
Fixing feasibility
Partition: $k = m_2 + 1$
$P_k = \{(\boldsymbol{x},\boldsymbol{y}): A\,\boldsymbol{x} + B\,\boldsymbol{y}^L \leq b\,, d^T\,\boldsymbol{y} \geq d^T \boldsymbol{y}^L\}$
Optimality

Figure 4.6: Branching Rules in B&B Algorithm.

LLP can only bear two results: either the LLP is unbounded yielding the BMILP instance as infeasible or has an optimal solution $(\boldsymbol{x}^H, \boldsymbol{y}^L)$. The LLP cannot be infeasible because this would have resulted in an infeasible HPP and the node would have been discarded at the HPP procedure stage. The branching rules implemented here are needed to fix either feasibility or optimality of the bi-level solution by branching in the follower's feasible space (Balas (1971), Glover (1973)). Infeasibility is fixed by reversing the inequalities of hyperplanes creating a node for each reversed hyperplane. In other words, if we have m_2 constraints for the follower, the feasible region is divided in m_2 partitions (i.e., $k = 1, \ldots, m_2$) to fix infeasibility as follows:

$$P_k = \{(\boldsymbol{x}, \boldsymbol{y}) : (A\boldsymbol{x} + B\boldsymbol{y}^L)_i \leq b_i,\ \forall i \neq k, (A\boldsymbol{x} + B\boldsymbol{y}^L)_k > b_k\}.$$

Optimality from the follower's perspective is handled by creating one last partition (i.e., $k = m_2 + 1$) as follows:

$$P_k = \{(\boldsymbol{x}, \boldsymbol{y}) : A\boldsymbol{x} + B\boldsymbol{y}^L \leq b, d^T\boldsymbol{y} \geq d^T\boldsymbol{y}^L\}.$$

Hence each parametric node problem $\mathcal{B}(l, u, w)$ is divided into $m_2 + 1$ nodes. Most importantly, the branching is applied in a way guaranteeing the enumeration of all possible feasible bi-level solutions and consequently the optimality of the incumbent.

4.5 Numerical Results

4.5.1 Randomly Generated Instances

Our proposed B&B algorithm was tested on randomly generated instances that were first published in Xu & Wang (2014). The algorithm was able to solve all instances to optimality. These randomly generated instances had the number of leader's, n_1, and follower's, n_2, decision variables equal and incremented by a value of 50 from 10 to 460 variables. The constraint dimensions m_1 and m_2 were set to be 40% of n_1. Since the leader's decision vector $\boldsymbol{x}$ is bounded, an upper-bound was chosen to be 10. Elements of all matrices and vectors have been chosen to follow a uniform distribution where $G_{\boldsymbol{x}}, G_{\boldsymbol{y}}, A,$ and B are within $[0, 10]$; $c_{\boldsymbol{x}}, c_{\boldsymbol{y}},$ and d are within $[-50, 50]$; q is within

$[30, 130]$; and b is within $[10, 110]$. Each element in the set $J_{\boldsymbol{y}}$, which contains the indices of integer decision variables for the follower, has been chosen independently and randomly with a 50% chance. Furthermore, for each level of n_1, ten random instances were generated.

4.5.2 Matlab Live Editor

The Matlab Live Editor is a tool that can be used online and in Matlab for combining code, output, and formatted text in an executable code notebook. In order to enrich the BMILP instance library, we are making this notebook publicly available, which can be used in Matlab and Matlab online. This notebook simply takes a CSV file that contains the attributes of the electrical network needed for building the mathematical programme for the defender-attacker-defender model. The output is a MAT file that contains the matrices for a generic BMILP 4.1.

Figure 4.7 include the format of the CSV file, together with Matlab live editor file should have sufficient information for users to generate BMILP instances for different electrical networks. Next, we generate BMILP instances for two electrical networks that have been used in the literature, and we validate our B&B algorithm results with other methodologies used in the literature.

4.5.3 Five-Bus System

The first electrical transmission network is shown in Figure 4.8 and was used before in Arroyo & Galiana (2005). It consists of 6 transmission lines, 5 generators and 5 buses. The loads are specified on each bus as well as the per unit reactance

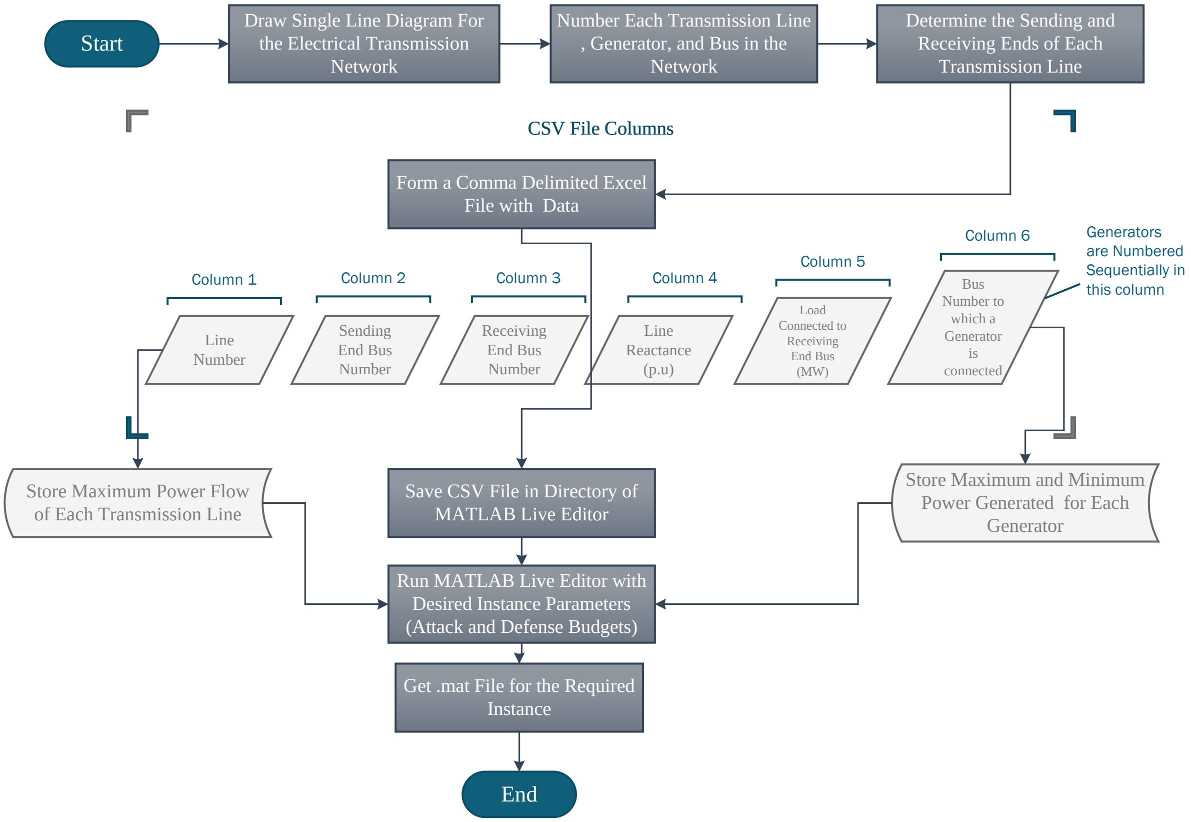

Start
Draw Single Line Diagram For the Electrical Transmission Network
Number Each Transmission Line , Generator, and Bus in the Network
Determine the Sending and Receiving Ends of Each Transmission Line
CSV File Columns
Form a Comma Delimited Excel File with Data
Column 1
Line Number
Column 2
Sending End Bus Number
Column 3
Receiving End Bus Number
Column 4
Line Reactance (p.u)
Column 5
Load Connected to Receiving End Bus (MW)
Column 6
Bus Number to which a Generator is connected
Generators are Numbered Sequentially in this column
Store Maximum Power Flow of Each Transmission Line
Save CSV File in Directory of MATLAB Live Editor
Store Maximum and Minimum Power Generated for Each Generator
Run MATLAB Live Editor with Desired Instance Parameters (Attack and Defense Budgets)
Get .mat File for the Required Instance
End

Figure 4.7: Flowchart for Formation of CSV File as Input for MATLAB Live Editor.

of each line. The BMVA (Base-Mega-Volt-Ampere) and BkV (Base-kilo-Volt) are taken as 100 MVA and 138 kV, respectively. The maximum power flow ($\bar{P}_l^f$) in each transmission line has been set to 100 MW, while the maximum and minimum power ($\bar{P}_j^g$, $\underline{P}_j^g$) that a generator can produce is set to 150 and zero MW, respectively. Moreover, transmission lines are numbered (squared boxes) as per Figure 4.8. The number of Leader decision variables are 6, and the number of leader constraints is 1. The number of follower decision variables is 174, while the number of follower constraints is 209. Table 4.2 summarizes the results for the B&B in comparison with

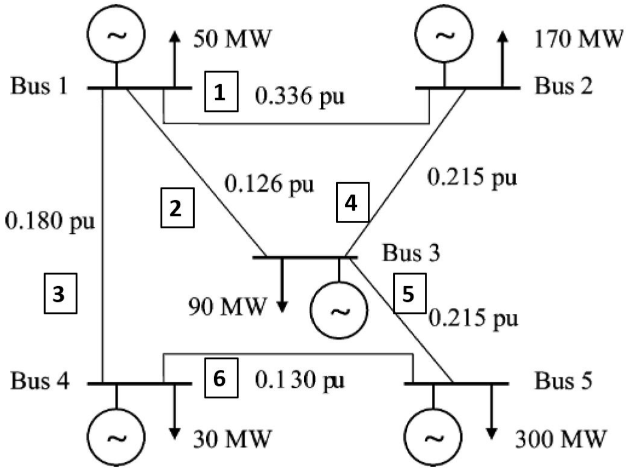


Figure 4.8: Five-Bus electrical transmission network.

the modified enumeration algorithm (i.e., MEA) proposed in Fakhry et al. (2022). The running time for MEA is significantly better than the B&B as seen in columns 4 and 5. This is due to three main reasons:

- The MEA algorithm is designed based on the structure of the attacker-defender models with a warm-start solution technique that significantly reduces the non-binary tree nodes visited, and as such it cannot be generalized beyond the scope of those models.
- The B&B algorithm is able to find alternative optimal solutions (if they exist), and consequently spending more time finishing the search tree.
- The branching rule used in the B&B algorithm grows exponentially with the size of the follower's constraints. In case of defending electrical power grids, all constraints belong to the follower's level making a huge search tree even in the most simplest cases.

Finding alternative optimal solutions can be useful in protecting critical infrastructure. It gives the planner the edge of having possibilities/alternatives of reducing the worst case scenario. In the context of protecting electrical power grids, there be might some hidden operational constraints that might impede implementing the optimal defence strategy. Deeply buried underground cables are an example of such difficulties, or cables that span long distances making it costly to defend or patrol.

Instance Num.	Defense Budget	Attack Budget	Avg. Run-time Node (sec)	Avg. Run-time Node Enum. Alg. (sec)	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. Created Nodes	Num. HPP Solved	Num. LLP Solved	Num. Alt. Optima
1	0	1	0.04	0.1506	50	-	6	211	212	2	0
2	0	2	0.09	0.0488	150	-	5, 6	211	212	2	0
3	0	3	0.13	0.0454	150	-	4, 5, 6	211	212	2	0
4	0	4	0.17	0.0390	170	-	3, 4, 5, 6	211	212	1	0
5	0	5	0.21	0.0372	170	-	1, 3, 4, 5, 6	211	212	1	0
6	0	6	0.26	0.0350	170	-	1, 2, 3, 4, 5, 6	211	212	2	0
7	1	1	0.17	0.1614	50	6	5	421	417	3	2
8	1	2	0.16	0.0523	50	5	2, 6	631	623	4	2
9	1	3	0.16	0.0454	70	6	1, 4, 5	841	831	4	0
10	1	4	0.18	0.0481	70	6	1, 2, 4, 5	1051	1041	5	0
11	1	5	0.22	0.0355	70	6	1, 2, 3, 4, 5	1051	1046	6	0
12	2	1	0.6	0.0638	0	5, 6	-	421	417	14	0
13	2	2	0.24	0.0693	20	5, 6	1, 4	1261	1240	7	0
14	2	3	0.18	0.0415	20	5, 6	1, 2, 4	2311	2270	12	0
15	2	4	0.17	0.0279	20	5, 6	1, 2, 3, 4	3571	3509	18	0
16	3	1	1.48	0.0616	0	5, 6	-	421	417	4	0
17	3	2	0.41	0.0357	0	1, 5, 6	3, 4	1681	1651	9	0
18	3	3	0.22	0.0282	0	1, 5, 6	2, 3, 4	3991	3913	20	0
19	4	1	2.1	0.0724	0	5, 6	-	421	417	4	0
20	4	2	0.57	0.0227	0	1, 4, 5, 6	2, 3	1681	1651	9	3
21	5	1	2.31	0.0546	0	5, 6	-	421	417	4	0

Table 4.2: Five-Bus System Instances using Branch and Bound Algorithm.

In addition to comparing run-times, Table 4.2 lists the objective value, defence and attack strategies for each allocated defense and attack budgets. Moreover, we list the number of created nodes, HPPs, LLPs traversed/solved by the B&B algorithm in each instance as shown in columns 8, 9 and 10 in Table 4.2. The number of alternative optimal solutions is listed in the lasted column. For example, in instance number 8 (i.e., defence budget=1, attack budget=2), the defender can choose line number 5 to defend, and in retaliation the attacker would choose line numbers 2 and 6 causing a load shed of 50 MW. As an alternative defence strategy, the planner/defender can defend line number 6, and then attacker would chose line numbers 4 and 5 causing the same load shed of 50 MW. In case of any hidden operational constraints, the planner can benefit from the knowledge that defending either line 5 or line 6 would prevent the worst damage to the electrical system.

4.5.4 Six-Bus System

The second electrical transmission network has been studied previously by Jiang et al. (2019). The system consists of eight transmission lines, two generators and six buses. The input parameters for the system's generator, load, branch data and line numbering have been taken from (Jiang et al. (2019)) for validation purposes. Results from the proposed solution algorithm have been grouped and summarized in Table 4.3.

Instance Num.	Defense Budget	Attack Budget	Avg. Run-time	Avg. Run-time	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. Created Nodes	Num. HPP Solved	Num. LLP Solved	Num. Alt. Optima
			Node (sec)	Node Enum. Alg. (sec)							
1	0	1	0.08	0.4785	43.63	-	5	265	266	1	0
2	0	2	0.15	0.5624	130	-	2, 5	265	266	1	0
3	0	3	0.24	0.1075	210	-	2, 3, 5	265	266	1	0
4	0	4	0.29	0.1906	290	-	2, 3, 4, 5	265	266	2	0
5	0	5	0.38	0.0549	290	-	2, 3, 4, 5	265	266	2	0
6	0	6	0.46	0.0917	290	-	2, 3, 4, 5	265	266	2	0
7	0	7	0.54	0.0455	290	-	2, 3, 4, 5, 6, 7, 8	265	266	2	0
8	0	8	0.63	0.0511	290	-	1, 2, 3, 4, 5, 6, 7, 8	265	266	2	0
9	1	1	0.44	0.153	31.08	5	2	529	526	2	0
10	1	2	0.39	0.1938	95	2	4, 5	793	788	3	0
11	1	3	0.38	0.1592	170	2	3, 4, 5	1057	1049	4	2
12	1	4	0.40	0.11092	170	2	3, 4, 5	1321	1310	5	2
13	1	5	0.39	0.0627	210	3	2, 4, 5, 8	1585	1571	6	1
14	1	6	0.39	0.0419	210	3	2, 4, 5, 6, 7, 8	1849	1829	7	1
15	1	7	0.40	0.0393	210	3	1, 2, 4, 5, 6, 7, 8	2113	2086	8	1
16	2	1	1.12	0.0578	25	2, 5	4	793	786	3	0
17	2	2	0.61	0.048	80	2, 5	3, 8	1585	1569	6	0
18	2	3	0.41	0.058	90	3, 5	2, 4, 7	2641	2607	10	2
19	2	4	0.37	0.0413	140	3, 4	2, 5, 6, 8	3961	3908	15	4
20	2	5	0.35	0.0315	140	3, 4	2, 5, 6, 7, 8	6073	5974	23	3
21	2	6	0.37	0.0278	140	2, 3	1, 4, 5, 6, 7, 8	8449	8284	32	3
22	3	1	3.06	0.053175	20	2, 4, 5	3	1057	1046	4	0
23	3	2	1.21	0.0614	60	2, 3, 4	5, 7	2905	2863	11	0
24	3	3	0.81	0.0454	70	3, 4, 5	2, 7, 8	5017	4933	19	3
25	3	4	0.52	0.0282	70	2, 3, 4	5, 6, 7, 8	10033	9849	38	3
26	3	5	0.46	0.0173	70	2, 3, 4	1, 5, 6, 7, 8	18217	17818	69	3
27	4	1	7.9	0.0017	0	2, 3, 4, 5	-	1057	1046	5	0
28	4	2	2.68	0.034	0	2, 3, 4, 5	-	3169	3121	14	0
29	4	3	1.95	0.0163	0	2, 3, 4, 5	-	4489	4417	18	0
30	4	4	0.77	0.0121	0	2, 3, 4, 5	1, 6, 7, 8	12409	12171	48	0
31	5	1	9.05	0.0553	0	2, 3, 4, 5	-	1057	1046	5	0
32	5	2	2.83	0.0251	0	2, 3, 4, 5	-	3433	3378	14	0
33	5	3	2.35	0.01	0	2, 3, 4, 5, 8	1, 6, 7	4489	4149	18	0
34	6	1	9.43	0.0623	0	2, 3, 4, 5	-	1057	1046	5	0
35	6	2	0.29	0.0143	0	2, 3, 4, 5, 6, 8	1, 7	3433	3378	14	0
36	7	1	0.96	0.046	0	2, 3, 4, 5	-	1057	1046	5	0

Table 4.3: Six-Bus System Instances using Branch and Bound Algorithm.

Following the same presentation of the previous electrical transmission network, Table 4.3 lists all possible instances for the Six-Bus system, in addition to run-time comparison between the B&B and MEA algorithm. For each instance, we list objective value (i.e., load-shed), defence and attack strategies. It is worth mentioning that the B&B algorithm produces heavy analytics regarding each created node, as depicted in Figure 4.4, in addition to the total number of nodes created, HPPs and LLPs solved, and number of alternative optimal solution available to the planner. The alternative solutions can be very critical in case of a limited defence budget such as instance 19 (i.e., defence budget=2, attack budget=4) in Table 4.3, where the planner can have four alternative strategies; all of which prevent the worst damage that can occur in the network.

4.6 Conclusions and Future Work

This paper has proposed a B&B approach for solving bi-level mixed-integer linear programmes and a general class of tri-level mixed-integer programmes with a convex optimization problem in their most lower-level. Furthermore, we provided a detailed literature review on the most recent efforts on developing general-purpose bi-level mixed-integer programmes. We have tested our algorithm on randomly generated instances from the literature. We report on computational efficiency in addition to rich data analytics on the solution of any BMILP instance. The reporting on the solution is done on the instance level and within the instance. The instance level reports on the numbers and types of relaxation problems solved to reach the optimal solution(s), and we report the number of alternative optima if applicable. Within the instance level reports on data specific to how the solution(s) was reached; these data include but not limited to the branching tree, number of nodes created, explored, and how it was fathomed. Furthermore, we test our algorithm on a specific class of tri-level problems which can be reduced to a mixed-integer bi-level programme; where we focus on the application of defending electrical transmission networks. Our noteworthy contributions include guaranteeing bi-level optimality, providing alternative optimal solution, and creating a general-purpose tool that can be tuned to account for different constraint or objectives. Additionally, in order to enrich the test bed of bi-level mixed-integer linear problems, we provide a Matlab live editor that converts any electrical transmission network to bi-level mixed-integer programme instance, in the context of enhancing the resilience of the electrical network under consideration.

For future research, we are planning to capitalize on the rich data that can be produced from the B&B algorithm, assess existing machine learning strategies that can aid the process of reaching the optimal solution, and extract insights that can help in developing theoretical results. Another future area of research is using the B&B algorithm in addition to specialized cuts tailored for specific practical applications.

Bibliography

Akbari-Jafarabadi, M., Tavakkoli-Moghaddam, R., Mahmoodjanloo, M. & Rahimi, Y. (2017), 'A tri-level r-interdiction median model for a facility location problem under imminent attack', *Computers & Industrial Engineering* **114**, 151–165.

Alguacil, N., Delgadillo, A. & Arroyo, J. M. (2014), 'A trilevel programming approach for electric grid defense planning', *Computers & Operations Research* **41**, 282–290.

Alvarez, R. E. (2004), Interdicting electrical power grids, PhD thesis, Monterey, California. Naval Postgraduate School.

Arroyo, J. M. (2010), 'Bilevel programming applied to power system vulnerability analysis under multiple contingencies', *IET generation, transmission & distribution* **4**(2), 178–190.

Arroyo, J. M. & Galiana, F. D. (2005), 'On the solution of the bilevel programming formulation of the terrorist threat problem', *IEEE transactions on Power Systems* **20**(2), 789–797.

Aussel, D. & Svensson, A. (2019), 'Is pessimistic bilevel programming a special case of a mathematical program with complementarity constraints?', *Journal of Optimization Theory and Applications* **181**(2), 504–520.

Balas, E. (1971), 'Intersection cuts—a new type of cutting planes for integer programming', *Operations Research* **19**(1), 19–39.

Bard, J. F. (1984), 'Optimality conditions for the bilevel programming problem', *Naval research logistics quarterly* **31**(1), 13–26.

Bard, J. F. (2013), *Practical bilevel optimization: algorithms and applications*, Vol. 30, Springer Science & Business Media.

Bard, J. F. & Moore, J. T. (1990), 'A branch and bound algorithm for the bilevel programming problem', *SIAM Journal on Scientific and Statistical Computing* **11**(2), 281–292.

Ben-Ayed, O. (1993), 'Bilevel linear programming', *Computers & operations research* **20**(5), 485–501.

Bialas, W. F. & Karwan, M. H. (1984), 'Two-level linear programming', *Management science* **30**(8), 1004–1020.

Breton, M., Alj, A. & Haurie, A. (1988), 'Sequential stackelberg equilibria in two-person games', *Journal of Optimization Theory and Applications* **59**(1), 71–97.

Brown, G., Carlyle, M., Salmerón, J. & Wood, K. (2006), 'Defending critical infrastructure', *Interfaces* **36**(6), 530–544.

Candler, W. & Norton, R. (1977), 'Multilevel programming and developing policy', *Technical Report 258, World Bank Development Research Center* .

Cao, D. & Chen, M. (2006), 'Capacitated plant selection in a decentralized manufacturing environment: a bilevel optimization approach', *European journal of operational research* **169**(1), 97–110.

Colson, B., Marcotte, P. & Savard, G. (2005), 'Bilevel programming: A survey', *4or* **3**(2), 87–107.

Colson, B., Marcotte, P. & Savard, G. (2007), 'An overview of bilevel optimization', *Annals of operations research* **153**(1), 235–256.

Davarikia, H. & Barati, M. (2018), 'A tri-level programming model for attack-resilient control of power grids', *Journal of Modern Power Systems and Clean Energy* **6**(5), 918–929.

Davarikia, H., Barati, M., Al-Assad, M. & Chan, Y. (2020), 'A novel approach in strategic planning of power networks against physical attacks', *Electric Power Systems Research* **180**, 106140.

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

Dempe, S. (2003), 'Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints'.

Dempe, S. & Dutta, J. (2012), 'Is bilevel programming a special case of a mathematical program with complementarity constraints?', *Mathematical programming* **131**(1), 37–48.

Dempe, S., Kalashnikov, V. V. & Kalashnykova, N. (2006), Optimality conditions for bilevel programming problems, *in* 'Optimization with Multivalued Mappings', Springer, pp. 3–28.

Dempe, S., Kalashnikov, V. V., Pérez-Valdés, G. A. & Kalashnykova, N. I. (2011), 'Natural gas bilevel cash-out problem: convergence of a penalty function method', *European Journal of Operational Research* **215**(3), 532–538.

Dempe, S., Mordukhovich, B. S. & Zemkoho, A. B. (2012), 'Sensitivity analysis for two-level value functions with applications to bilevel programming', *SIAM Journal on Optimization* **22**(4), 1309–1343.

Dempe, S., Mordukhovich, B. S. & Zemkoho, A. B. (2014), 'Necessary optimality conditions in pessimistic bilevel programming', *Optimization* **63**(4), 505–533.

DeNegre, S. T. & Ralphs, T. K. (2009), A branch-and-cut algorithm for integer bilevel linear programs, *in* 'Operations research and cyber-infrastructure', Springer, pp. 65–78.

Faísca, N. P., Dua, V., Rustem, B., Saraiva, P. M. & Pistikopoulos, E. N. (2007), 'Parametric global optimisation for bilevel programming', *Journal of Global Optimization* **38**(4), 609–623.

Fakhry, R., Hassini, E., Ezzeldin, M. & El-Dakhakhni, W. (2022), 'Tri-level mixed-binary linear programming: Solution approaches and application in defending critical infrastructure', *European Journal of Operational Research* **298**(3), 1114–1131.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2016), Intersection cuts for bilevel optimization, *in* 'International Conference on Integer Programming and Combinatorial Optimization', Springer, pp. 77–88.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2017), 'A new general-purpose algorithm for mixed-integer bilevel linear programs', *Operations Research* **65**(6), 1615–1637.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2018), 'On the use of intersection cuts for bilevel optimization', *Mathematical Programming* **172**(1), 77–103.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2019), 'Interdiction games and monotonicity, with application to knapsack problems', *INFORMS Journal on Computing* **31**(2), 390–410.

Fortuny-Amat, J. & McCarl, B. (1981), 'A representation and economic interpretation of a two-level programming problem', *Journal of the operational Research Society* **32**(9), 783–792.

Gadhi, N. & Dempe, S. (2012), 'Necessary optimality conditions and a new approach to multiobjective bilevel optimization problems', *Journal of Optimization Theory and Applications* **155**(1), 100–114.

Glover, F. (1973), ‘Convexity cuts and cut search’, *Operations Research* **21**(1), 123–134.

Glover, F. (1974), ‘Polyhedral convexity cuts and negative edge extensions’, *Zeitschrift für Operations Research* **18**(5), 181–186.

Jeroslow, R. G. (1985), ‘The polynomial hierarchy and a simple model for competitive analysis’, *Mathematical programming* **32**(2), 146–164.

Jiang, P., Huang, S. & Zhang, T. (2019), ‘Optimal deception strategies in power system fortification against deliberate attacks’, *Energies* **12**(3), 342.

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**, 100007.

Kleinert, T., Labbé, M., Plein, F. & Schmidt, M. (2021), ‘Closing the gap in linear bilevel optimization: a new valid primal-dual inequality’, *Optimization Letters* **15**(4), 1027–1040.

Kleniati, P.-M. & Adjiman, C. S. (2014), ‘Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. part i: Theoretical development’, *Journal of Global Optimization* **60**(3), 425–458.

Konur, D. & Golias, M. M. (2013), ‘Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty’, *Computers & Industrial Engineering* **65**(4), 663–672.

Labbé, M., Marcotte, P. & Savard, G. (1998), ‘A bilevel model of taxation and its application to optimal highway pricing’, *Management science* **44**(12-part-1), 1608–1622.

Leitmann, G. (1978), ‘On generalized stackelberg strategies’, *Journal of optimization theory and applications* **26**(4), 637–643.

Liu, J., Fan, Y., Chen, Z. & Zheng, Y. (2018), ‘Pessimistic bilevel optimization: a survey’, *International Journal of Computational Intelligence Systems* **11**(1), 725.

Liu, S., Wang, M., Kong, N. & Hu, X. (2021), ‘An enhanced branch-and-bound algorithm for bilevel integer linear programming’, *European Journal of Operational Research* **291**(2), 661–679.

Loridan, P. & Morgan, J. (1996), ‘Weak via strong stackelberg problem: new results’, *Journal of global Optimization* **8**(3), 263–287.

Lozano, L. & Smith, J. C. (2017), ‘A backward sampling framework for interdiction problems with fortification’, *INFORMS Journal on Computing* **29**(1), 123–139.

Lu, J., Han, J., Hu, Y. & Zhang, G. (2016), ‘Multilevel decision-making: A survey’, *Information Sciences* **346**, 463–487.

Mitsos, A. (2010), ‘Global solution of nonlinear mixed-integer bilevel programs’, *Journal of Global Optimization* **47**(4), 557–582.

Moore, J. T. & Bard, J. F. (1990), ‘The mixed integer linear bilevel programming problem’, *Operations research* **38**(5), 911–921.

Outrata, J. V. (1990), ‘On the numerical solution of a class of stackelberg problems’, *Zeitschrift für Operations Research* **34**(4), 255–277.

Saharidis, G. K. & Ierapetritou, M. G. (2009), ‘Resolution method for mixed integer bi-level linear problems based on decomposition technique’, *Journal of Global Optimization* **44**(1), 29–51.

Tahernejad, S., Ralphs, T. K. & DeNegre, S. T. (2020), ‘A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation’, *Mathematical Programming Computation* **12**(4), 529–568.

von Stackelberg, H. (2011), *Market Structure and Equilibrium*, Springer Berlin Heidelberg, Berlin, Heidelberg.

Wang, L. & Xu, P. (2017), ‘The watermelon algorithm for the bilevel integer linear programming problem’, *SIAM Journal on Optimization* **27**(3), 1403–1430.

Wen, U.-P. & Hsu, S.-T. (1991), ‘Linear bi-level programming problems—a review’, *Journal of the Operational Research Society* **42**(2), 125–133.

Wen, U.-P. & Yang, Y. (1990), ‘Algorithms for solving the mixed integer two-level linear programming problem’, *Computers & Operations Research* **17**(2), 133–142.

Wu, X. & Conejo, A. J. (2017), ‘An efficient tri-level optimization model for electric grid defense planning’, *IEEE Transactions on Power Systems* **32**(4), 2984–2994.

Xiang, Y., Zhang, X., Shi, D., Diao, R. & Wang, Z. (2020), ‘Robust optimization for transmission defense against multi-period attacks with uncertainties’, *International Journal of Electrical Power & Energy Systems* **121**, 106154.

Xu, P. & Wang, L. (2014), ‘An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions’, *Computers & operations research* **41**, 309–318.

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

Zeng, B. & An, Y. (2014), ‘Solving bilevel mixed integer program by reformulations and decomposition’, *Optimization online* pp. 1–34.

Chapter 5

On Solving Fortification Games: A Heuristically-Enhanced Decomposition-Based Exact Algorithm

Abstract

This chapter examines a special class of tri-level optimization problems known as Stackelberg sequential games. In general terms, the three-level non-cooperative game, with perfect information, can either have min-max-min or max-min-max structure, where each level represents a player sharing a set of items with the next player, and optimizing a common objective function in opposite direction. These problems are notoriously difficult to optimize, because of the inherent tri-level structure which is crucial for modelling the players' interactions. The three-stage problem structure cannot be evaded, if the most lower-level problem is $\mathcal{NP}$-hard. Nevertheless, even for the simplest case where the lower-level problem is a convex problem, and the tri-level problem can be reduced to a bi-level structure using Karush-Khun-Tucker (KKT) conditions or duality theory, the mathematical programme is known to be strongly $\mathcal{NP}$-hard. In this chapter, we propose a heuristically-enhanced exact algorithm for solving the aforementioned class of tri-level problems, where the most lower-level problem can be $\mathcal{NP}$-hard. The main idea of the algorithm relies on forming a single-level equivalent of the tri-level problem, where the feasible region is constructed incrementally in each iteration of the algorithm. Moreover, we rely on heuristics gained from structural domain-knowledge of the application to enhance the formation of the feasible region. This idea can be implemented on various applications, and we demonstrate the effectiveness of our proposed solution on two

applications. The first is the widely-studied application of defending critical infrastructure to improve its resilience against intentional attacks. In this context, we use a defender-attacker-operator model and apply it to electrical transmission networks, where the most lower-level is a convex optimization problem. The second application is the capacitated lot-sizing problem with the capability of interdiction and fortification. This modified version of the ubiquitous lot-sizing problem is characterized by having its most lower-level problem a mixed-binary programme rendering the overall tri-level problem inherently very difficult to solve. We test our solution approaches on three electrical networks that vary in size, and randomly generated instances of lot-sizing problems.We present the results of our numerical computations as well as some insights.

5.1 Introduction

Decentralized planning was the main drive-force behind multi-level programmes (**MLPs**). Since the late 20th century, and first introduced by Candler & Norton (1977), MLPs have been used to model decentralized problems that require planning and involve several decision-makers. In many practical optimization scenarios, a decision-maker has to factor other parties into account resulting in nested mathematical programmes, where the lower-levels have a subset of their decision variables affecting the upper-level programmes.

5.1.1 Bi-Level Programmes

In the most elementary and simplest form of MLPs, a bi-level programme (**BLP**) is considered a special case, where there are only two decision-makers, and is shown to be $\mathcal{NP}$-hard (Bard 1991). Such bi-level structures were introduced in the game theory field by von Stackelberg (2011) where two non-cooperative players interact in a sequential manner, also denoted as *leader-follower game.* In that particular situation, the leader gets to take the first move, and then the follower reacts to it. Hence, the follower's reaction is influenced by the first decision-maker, the leader, who has full knowledge of the follower's objective function and constraints, i.e., a game with perfect information. A generic BLP can be formulated as follows:

$$\begin{aligned} &\text{`}\max_{\boldsymbol{x}}\text{'} && f_1(\boldsymbol{x},\boldsymbol{y}) \\ &\text{s.t.} && g_1(\boldsymbol{x},\boldsymbol{y}) \geq 0 \\ &&& \boldsymbol{z} \in \arg\min_{\boldsymbol{y}} \quad f_2(\boldsymbol{x},\boldsymbol{y}) \\ &&& \qquad\quad \text{s.t.} \qquad g_2(\boldsymbol{x},\boldsymbol{y}) \geq 0 \end{aligned} \tag{5.1}$$

Problem 5.1 can be seen as a leader-follower game, where the leader controls decision variables $\boldsymbol{x}$ and the follower reacts to the leader's move through its own decision variables $\boldsymbol{y}$. The quotation marks in Problem 5.1 are used to indicate the ill-positioning of that particular BLP, as no solution approach, or strategy, has been selected. In particular, the ill-positioning stems from the possibility of the follower's problem being non-strictly convex and possibly resulting in multiple alternative global optima. Thus, an approach needs to be selected for the aforementioned problem to be well-posed: an *optimistic approach* (Dempe 2002) where one of the alternative optima is selected in favour of the leader's objective function, $f_1(\boldsymbol{x},\boldsymbol{y})$, or a *pessimistic approach* (Aussel & Svensson 2019) where a solution is picked to be against the leader's objective function.

BLPs have gained an increasing attention from researchers over the past years as discussed in Chapter 4. With that came the need for developing efficient general-purpose solvers. For instance, Bard & Moore (1990) proposed a branch-and-bound method to solve mixed-integer linear BLPs. Their ideas and definitions paved the way for subsequent refinement trials proposed by Xu & Wang (2014), Fischetti et al.

(2017), and most recently Liu et al. (2021).

Interdiction Games

The closest related class of BLPs that relates to the tri-level programme that we examine in this chapter is known in the literature as *interdiction games* (Fischetti et al. 2019). In this class of problems, the leader and the follower share a set of decision variables and an objective function. The leader gets to make the first move by *interdicting* (i.e., choosing a subset of the shared variables), then with an adversarial counter-act manner the follower tries to push the objective function in an opposite direction subject to their own set of constraints. This is why interdiction games (**IGs**) either have a min-max or a max-min structure. In mathematical terms, an IG problem $\mathcal{I}$ can be structured as follows:

$$\mathcal{I}: \quad D^{\mathcal{I}} = \max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} f(\boldsymbol{y}) \tag{5.2}$$

Generally, IGs can be divided into two problems: the upper-level, or the outer-level, in which we would formally call an *attack* problem $\mathcal{A}$ and the lower-level, or the inner-level, denoted as a *recourse* problem $\mathcal{R}$. IGs, at an abstract level, can be modeled over problems with a well-studied network structure such as shortest path problems (Cappanera & Scaparra (2011), Israeli & Wood (2002), Held & Woodruff (2005)), optimal load flow problems in electrical power grids (Salmeron et al. (2004), Delgadillo et al. (2009)), and maximum flow networks (Akgün et al. 2011). Most often, the link between the attack and recourse problems is through binary decision variables that when selected, or *interdicted*, by the first decision-maker implies that a set of actions that can be taken by the second decision-maker are inhibited. This implies a 1-1 mapping between the *interdicted*, a subset of the follower's decision variables, and *interdiction* variables. It is worth mentioning that Problem 5.2 is not ill-posed. This is so because of the facts that both players share the same objective function $f(\boldsymbol{z})$ and the adversarial nature of IGs. For a comprehensive review on interdiction games, the reader can refer to Smith & Lim (2008) and Smith (2010).

5.1.2 Tri-Level Programmes

A tri-level programme (**TLP**) acquires the complexity of a BLP and appends an extra layer to the hierarchical structure. For example, having two inner-problems raises the need to resolve degenerate solutions at two levels- instead of just the lower-level in case of BLP. However, each of the alternatives might have a different impact on the overall problem. For a brief discussion on resolving degenerate solutions for TLPs, the interested reader might refer to the work done by Florensa et al. (2017), and Fakhry et al. (2022).

Fortification Games

The class of tri-level programmes that we focus on in this chapter is known in the literature as *interdiction games with fortifications* (Lozano & Smith 2017) or in-short *fortification games* (**FGs**). Formally, let $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ denote the vectors of decision variables for the first-, second-, and third-levels in a FG, respectively. We can formulate a general FG problem $\mathcal{F}$ as follows:

$$\mathcal{F}: \quad D^* = \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \min_{\boldsymbol{z}\in\mathcal{Z}(\boldsymbol{y})} f(\boldsymbol{z}). \tag{5.3}$$

Where $\mathcal{X} = \left\{\boldsymbol{x} \in \{0,1\}^{n_x} : A_f^T\boldsymbol{x} \leq B_f\right\}$ is the set of constraints for the leader, i.e., defender in FG, n_x is the number of assets that can be protected, $A_f^T \in \mathbb{R}_+^{1\times n_x}$ denotes the cost of fortifying each asset, and $B_f \in \mathbb{R}_+$ is the fortification budget. It is clear that a FG can be reduced to an IG, i.e., the two inner-problems, if a feasible defence vector $\hat{\boldsymbol{x}} \in \mathcal{X}$ is passed as a parameter to the interdiction game. On that regard, let the IG problem for a given defence strategy $\hat{\boldsymbol{x}}$ be defined as follows:

$$\mathcal{I}(\hat{\boldsymbol{x}}): \quad D^{\mathcal{I}}(\hat{\boldsymbol{x}}) = \max_{\boldsymbol{y}\in\mathcal{Y}(\hat{\boldsymbol{x}})} \min_{\boldsymbol{z}\in\mathcal{Z}(\boldsymbol{y})} f(\boldsymbol{z}). \tag{5.4}$$

Where $\mathcal{Y}(\hat{\boldsymbol{x}}) = \left\{\boldsymbol{y} \in \{0,1\}^{n_y} : A_d^T\boldsymbol{y} \leq B_d,\ y_i \leq 1 - \hat{x}_i \ \ \forall\, i \in n_{xy}\right\}$ is the set of constraints for the follower (i.e., attacker in FG), n_y is the number of assets that can be disrupted, $A_d^T \in \mathbb{R}_+^{1\times n_y}$ is the cost damage vector incurred by the attacker for choosing to attack assets, and $B_d \in \mathbb{R}_+$ is the damage budget. Lastly, the linking constraint in the feasible region $\mathcal{Y}$, implies that if the defender chose to fortify a certain asset (i.e., $\hat{x}_i = 1$) in the shared set of variables n_{xy} between the two players, it cannot be interdicted by the attacker (i.e., $y_i = 0$). This singles out the last inner-problem, which can be seen as another sequential move by the defender rendering the whole FG problem as *defender-attacker-defender* (**DAD**), or a move by an operator trying to minimize the damage caused by the attacker through taking recourse decisions. Let the recourse problem $\mathcal{R}$ for a given defence and attack strategies $\hat{\boldsymbol{x}}$,

$\hat{\boldsymbol{y}}$, respectively, be denoted $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ and defined as follows:

$$\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}): \quad D^{\mathcal{R}}(\hat{\boldsymbol{y}}) = \min_{\boldsymbol{z} \in \mathcal{Z}(\hat{\boldsymbol{y}})} f(\boldsymbol{z}) \tag{5.5}$$

As is the case of Problem 5.2, the FG Problem 5.3 is not ill-posed, as each sequential player is trying to optimize the same objective function $f(\boldsymbol{z})$ in an opposite direction. Hence, FGs can either have a min-max-min or max-min-max structure, with the latter being less prevalent. Furthermore, as an extension of IGs, FGs can arise in models whose network models are well studied such as shorted-path FG (Sadeghi et al. (2017), Lozano & Smith (2017)), protecting rail-road intermodal networks (Sarhadi et al. 2017), facility location problems (Church & Scaparra (2007), Akbari-Jafarabadi et al. (2017)) and protecting electrical power grids (Wu & Conejo (2017), Alguacil et al. (2014), Fakhry et al. (2022)).

The solution approaches for FGs can be divided into three main categories: enumeration techniques, heuristic approaches, and lastly decomposition methodologies. However, more than one methodology can be combined to form a hybrid solution approach.

Explicit (Mahmoodjanloo et al. 2016) or implicit (Church & Scaparra (2007), Scaparra & Church (2012)) enumeration are being used as solution techniques, especially when the total number of possible attack plans is not too large. Explicit enumeration starts by enumerating feasible solutions at the most two outer-problems, the fortification and attack problems, then uses an exact approach for the recourse problem. Implicit enumeration starts by feeding the worst-case scenario to the recourse problem, where it starts the root of the search tree, and then advances forward by enumerating the possible combinations. Enumeration techniques have been refined, where a heuristic is used to aid the search tree, and provide *warm-starting* solutions from the parent node to child nodes. This method was particularly rewarding for mid- to large-sized networks (Fakhry et al. 2022).

Heuristic techniques, including meta-heuristics such as genetic algorithms (Mahmoodjanloo et al. 2016), tabu search, rain-fall optimization, and random greedy search (Akbari-Jafarabadi et al. 2017), are used either in combination with enumeration techniques or to guide the search. However, most of the meta-heuristics are tailored to the structure of the recourse problem. Recently, Fakhry et al. (2022) proposed three generic heuristics, that differ in terms of quality and time needed to reach to optimal/near-optimal solutions, that can be tailored according to the structure of FG.

Decomposition or reformulation techniques in FGs have started with the work of Brown et al. (2006), where the attack and recourse problems, the most two lower-levels, are decomposed into one-level problem, converting a tri-level FG having, for e.g., a min-max-min structure, into a min-max BLP. This is done by taking the dual of the most lower-level, or the recourse problem, and appending dual, primal, and strong duality constraints into the second-level, the attack problem, resulting into a min-max BLP Alguacil et al. (2014). Branch-and-bound general-purpose algorithms, heuristics, and enumeration techniques can be used to tackle the reformulated problem. An equivalent approach for dualizing a TLP FG, and converting it into a BLP is using the KKT conditions of the recourse problem and appending it to the second-level. A proof of this equivalence is provided in (Fakhry et al. 2022) for FGs.

However depending on how the constraints are formulated and the nature of the objective function, one approach might be computationally superior over the other. It is worth mentioning that both transformation techniques require the recourse problem to be convex. In addition, replacing the lower-level by its KKT optimality conditions does not necessarily yield a solution for the initial problem. Recent research on BLPs, discuss these cases and the conditions for equivalence for the pessimistic (Aussel & Svensson 2019), and optimistic Dempe & Dutta (2012) approaches.

There are relatively few studies of FGs that consider $\mathcal{NP}$-hard recourse problems in IGs (Tang et al. 2016), and expectedly, even fewer in FGs (Prince et al. (2013), Sarhadi et al. (2017)). In Prince et al. (2013), the authors transformed the non-convexity of the recourse problem into an equivalent linear programme using a shortest-path formulation that is pseudo-polynomial in size, while in Sarhadi et al. (2017) a decomposition-based heuristic solution was implemented that does not guarantee an optimal solution. A similar study by Wu & Conejo (2017) in the context of protecting electric grids FG, used a decomposition-based approach, where the recourse problem was convex. Information is passed from a sub-problem to a master-problem. However, this method provides only approximate solutions in some cases based on their numerical results.

Our work is inspired by two studies. First is the research done by Fischetti et al. (2019) in IGs, where a decomposition-based solution scheme that introduces cutting planes is applied. Furthermore, the authors used some heuristic procedures to lift and tighten those cuts. Our solution approach extends those cutting planes to FGs. Second, the framework by Lozano & Smith (2017) for solving FGs where the recourse problem is non-convex, by restricting the most lower-level to a set of feasible solutions. Those solutions are called “samples,” and they are collected through a perturbation/sampling procedure of the recourse problem depending on the FG under consideration. Moreover, if the sample size is too large, the IG may be potentially too difficult to solve. On the other hand, if it is too small, it will lead to poor bounds. Our proposed solution approach does not require a set of feasible solutions for the recourse problem.

The contributions of this research work can be summarized as follows:

1. We provide a decomposition-based approach for solving FGs with convex and non-convex recourse problems.

2. Our proposed approach terminates finitely with the exact optimal solution.

3. We provide and apply ideas on enhancing and accelerating the algorithm based on structural knowledge of the FG.

4. We provide numerical results on two types of FGs: protecting electrical power grids characterized with convex recourse problem, and capacitated lot-sizing problem with fortification associated with non-convex recourse problems.

The rest of this chapter is outlined as follows: Section 5.2 provides the algorithmic details and the theoretical argument behind our proposed approach. Section 5.3 provides the background and results for implementing our algorithm on two FGs with different contexts. Lastly Section 5.4 concludes this research and provides directions for possible extensions.

5.2 Algorithm Details

In this section we provide the background as well as the procedures used in our algorithm.

5.2.1 Background

Our proposed approach can be divided into two main procedures:

1. A procedure for getting the optimal attack vector $\boldsymbol{y}^* \in \mathcal{Y}(\hat{\boldsymbol{x}})$ and inflicted damage $D^{\mathcal{I}}(\hat{\boldsymbol{x}})$ solution for the IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$ for a given defence vector $\hat{\boldsymbol{x}}$, which we previously defined as follows:

$$\mathcal{I}(\hat{\boldsymbol{x}}): \quad D^{\mathcal{I}}(\hat{\boldsymbol{x}}) = \max_{\boldsymbol{y} \in \mathcal{Y}(\hat{\boldsymbol{x}})} \min_{\boldsymbol{z} \in \mathcal{Z}(\boldsymbol{y})} f(\boldsymbol{z}), \tag{5.6}$$

where $\mathcal{Y}(\hat{\boldsymbol{x}}) = \Big\{\boldsymbol{y} \in \{0,1\}^{n_y} : A_d^T \boldsymbol{y} \leq B_d,\ y_i \leq 1 - \hat{x}_i \ \ \forall\, i \in n_{xy}\Big\}$ is the set of constraints for the attacker in IG.

2. A procedure for obtaining the optimal defence vector $\boldsymbol{x}^*$ solution for the overall FG $\mathcal{F}$ previously defined in 5.3 as follows:

$$\mathcal{F}: \quad D^* = \min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} \min_{\boldsymbol{z} \in \mathcal{Z}(\boldsymbol{y})} f(\boldsymbol{z}), \tag{5.7}$$

where $\boldsymbol{x}^* \in \mathcal{X} = \Big\{\boldsymbol{x} \in \{0,1\}^{n_x} : A_f^T \boldsymbol{x} \leq B_f\Big\}$.

The general idea of the algorithm relies on decomposing the IG problem into two parts: the first part is the recourse problem $\mathcal{R}$ and the second, a restricted master-problem, for a given defence and attack vectors $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, respectively, let $D^{\mathcal{R}}(\hat{\boldsymbol{y}})$ be the value of recourse inflicted damage for a given attack vector $\hat{\boldsymbol{y}}$ and defined as follows:

$$\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}): \quad D^{\mathcal{R}}(\hat{\boldsymbol{y}}) = \min_{\boldsymbol{z} \in \mathcal{Z}(\hat{\boldsymbol{y}})} f(\boldsymbol{z}) \tag{5.8}$$

In Proposition 5.2.1 we show that a solution to 5.8 yields a lower bound for IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$.

Proposition 5.2.1. *For any feasible attack vector $\hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}})$, the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ yields a lower-bound $LB =\ D^{\mathcal{R}}(\hat{\boldsymbol{y}})$ for the overall IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$.*

Proof. Let $\hat{\boldsymbol{z}}^*$ be the optimal solution to problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ and yields an objective value $D^{\mathcal{R}}(\hat{\boldsymbol{y}})$, thereby providing a solution to the inner problem of the IG, $\mathcal{I}(\hat{\boldsymbol{x}})$. From an attacker's perspective, the value $D^{\mathcal{R}}(\hat{\boldsymbol{y}})$ is just a feasible solution, as $\hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}})$. The proof is completed by the fact that any feasible solution to a maximization problem (i.e., $\mathcal{I}(\hat{\boldsymbol{x}})$) provides a lower bound. □

Proposition 5.2.1 will aid in defining feasible region by providing a lower-bound in each iteration of our proposed algorithm and obtaining the optimal solution, $\hat{\boldsymbol{z}}^*$ to problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$.

5.2.2 Optimal Attack-Recourse Procedure

To get the optimal attack response for an IG, whether the recourse problem is convex or non-convex, a restricted master-problem ($\mathbf{RMP}^{\mathcal{I}}$) is built based on the response (i.e., solution) from the recourse problem. In other words, the RMP$^{\mathcal{I}}$ is built from the attacker's perspective and based on the knowledge of a defence vector $\hat{\boldsymbol{x}}$ (i.e., RMP$^{\mathcal{I}}(\hat{\boldsymbol{x}})$). It can be defined as follows:

RMP$^{\mathcal{I}}(\hat{x})$

$$\max_{\boldsymbol{y}, D^{\mathcal{I}}} \quad D^{\mathcal{I}} \tag{5.9}$$

$$\text{s.t.} \quad D^{\mathcal{I}} \leq f(\boldsymbol{z}_k) + \sum_{i \,\in\, n_y} M_i^{\mathcal{I}}(\boldsymbol{z}_k)\, y_i \quad \forall\, \boldsymbol{z}_k \in \mathcal{Z}(\hat{\boldsymbol{y}}), \hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}}) \tag{5.10}$$

$$\boldsymbol{y} \in \mathcal{Y}(\hat{\boldsymbol{x}}) \tag{5.11}$$

The objective of the restricted master-problem (5.9) is coupled with inequality constraints defined in (5.10). The attacker's objective is to raise the inflicted damage $D^{\mathcal{I}}$. For a given defence vector $\hat{\boldsymbol{x}}$, $D^{\mathcal{I}}$ would be the objective value from the attacker's perspective, which is the maximum damage that could be inflected. However, this is linked with the recourse solution $\boldsymbol{z}_k$, and the shared objective value $f(\boldsymbol{z}_k)$ for each response attack vector $\hat{\boldsymbol{y}}$. It is worth mentioning that each $\hat{\boldsymbol{y}}$ will be feasible for the attack problem because of constraint (5.11). The choice of the $M_i^{\mathcal{I}}(\boldsymbol{z}_k)$ values that are defined in constraint (5.10) is pivotal for the convergence and effectiveness of the proposed algorithm. It depends on the structure of the recourse problem $\mathcal{R}$ 5.5. In particular, the linking constraint(s) between the attack vector $\hat{\boldsymbol{y}}$ and the recourse decision variables $\boldsymbol{z}$ determine how the $M_i^{\mathcal{I}}(\boldsymbol{z}_k)$ are set. This will be explained further in the coming sections, when we address two different applications with relatively complicated recourse problems.

Using penalty terms in IGs has been explored before by Wood (2010) and it was found to be problem-dependent. Moreover, Caprara et al. (2016) found that for the knapsack interdiction problem, the $M_i^{\mathcal{I}}$ values should be set to d_i, where d_i is the benefit of having item i in the follower's knapsack. A few years later, Fischetti et al. (2019) provided a proof for the validity of those cuts under the condition that the IG satisfies the downward monotonicity property. The two applications we will study in this chapter, protecting electrical power grids and capacitated lot-sizing with fortification, do not satisfy the property mentioned in (Fischetti et al. 2019). Furthermore, the structure provided in (Caprara et al. 2016, Fischetti et al. 2019) implies the existence of every element of the recourse decision vector $\boldsymbol{z}$ in the shared objective function (i.e., $d_i > 0$), which is not always applicable in IGs. We provide a practical and straight-forward way for assigning the M_i values based on the problem structure that is a function of the optimal recourse vector $\hat{\boldsymbol{z}}^*$ and we extend those cuts to the overall FG. In Proposition 5.2.4 we provide conditions when the restricted mater-problem leads to an upper bound for the overall IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$.

Proposition 5.2.2. *For a sufficiently large and non-negative $M_i^{\mathcal{I}}(\boldsymbol{z}_k)$ values and a given defence vector $\hat{\boldsymbol{x}}$, the restricted master-problem (RMP$^{\mathcal{I}}$)* (5.9) *provides an upper-bound, UB, for the overall IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$.*

Proof. Let $D^{\mathcal{I}}$ be the optimal value from the restricted master-problem (RMP$^{\mathcal{I}}$) (5.9). We note that the set of constraints (5.10) is applied for optimal recourse decision vector(s) $\boldsymbol{z}_k$, which consequently set the right hand side of constraint (5.10) to the common objective function of value $f(\boldsymbol{z}_k)$. Since we have non-negative $M_i^{\mathcal{I}}(\boldsymbol{z}_k)$ values, this implies $D^{\mathcal{I}} \geq f(\boldsymbol{z}_k)$ providing an UB for the IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$. □

We now have the necessary ingredients to present the optimal attack-recourse algorithm, which we will denote the (**OAR**) algorithm. The pseudocode is presented in Algorithm 5.1. The main idea of the algorithm starts with breaking down the structure of IG $\mathcal{I}(\hat{\boldsymbol{x}})$ into two problems, namely: the restricted master-problem from the attacker's perspective (i.e., $\text{RMP}^{\mathcal{I}}(\hat{x})$) and the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, which is used to iteratively supply cuts to the restricted master-problem. These cuts are forming the feasible region from the attacker's perspective. From a game-theory perspective, the attacker is learning how the defender/operator would behave for a given attack vector $\hat{\boldsymbol{y}}$. This learning behaviour is done through two elements that change with the recourse response vector $\hat{\boldsymbol{z}}^*$ forming the right hand side of constraint (5.10). The first element is the shared objective function $f(\hat{\boldsymbol{z}}^*)$ and most importantly the $M^{\mathcal{I}}(\boldsymbol{z}^*)$ values; those two terms change each iteration. The iterative structure of the OAR algorithm works as long as the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ can be solved to optimality given an attack vector $\hat{\boldsymbol{y}}$. Thereby, if we have a convex recourse such as the shortest-path problem, we can solve it using Dijkstra's algorithm (Dijkstra et al. 1959). Moreover, if we have a non-convex recourse such as the capacitated lot-sizing problem, which is known to be $\mathcal{NP}$-hard (Bitran & Yanasse 1982), we can attempt to solve it using dynamic programming or as a mixed-integer programme using state-of-the-art branch-and-bound commercial solvers.

Lines 1-4 in Algorithm 5.1, initialize the iteration counter, upper-bound (UB), lower-bound (LB), a feasible initial attack vector $\hat{\boldsymbol{y}}_0$, and a corresponding recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}_0)$. Line 5 initiates the repository for storing the cuts' parameters; which are obtained in each iteration by solving the recourse problem. Lines 6 to 19

Algorithm 5.1 Optimal Attack-Recourse Framework

```
Input: Problem $\mathcal{I}(\hat{\boldsymbol{x}})$
Output: An optimal solution to $\mathcal{I}(\hat{\boldsymbol{x}})$
 1: Set iteration count $i = 0$
 2: Set $\text{UB}_i = \infty$ and $\text{LB}_i = -\infty$
 3: Initialize a feasible attack vector $\hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}})$
 4: Populate recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}_0)$
 5: Initialize Cuts (5.10) repository
 6: while $\text{UB}_i - \text{LB}_i > \epsilon$ do
 7:     Set $i = i + 1$
 8:     Solve $\text{LB}_i = \min_{\boldsymbol{z} \in \mathcal{Z}(\hat{\boldsymbol{y}})} f(\boldsymbol{z})$ and obtain an optimal solution $\hat{\boldsymbol{z}}^*$
 9:     Update $M_i^{\mathcal{I}}(\hat{\boldsymbol{z}}^*)$ values for $\text{cut}_i$
10:     Store $\text{cut}_i$(5.10) in Cuts repository
11:     for each $\text{cut}_i$ in Cuts repository do
12:         Add $\text{cut}_i$(5.10) to $\text{RMP}_i^{\mathcal{I}}(\hat{x})$ (5.9)
13:     end for
14:     Solve $\text{UB}_i = \text{RMP}_i^{\mathcal{I}}(\hat{x})$ and obtain an attack vector $\hat{\boldsymbol{y}}_i$
15:     Re-populate linking constraints in recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}_i)$
16:     if $\text{UB}_i = \text{LB}_i$ then
17:         Terminate with solution $(\hat{\boldsymbol{y}}^*, \hat{\boldsymbol{z}}^*)$
18:     end if
19: end while
```

form a while loop that terminates when the UB and LB converge. In each iteration, line 7 updates the iteration count i. Line 8 obtains a LB_i by solving the recourse problem; this is considered the main learning tool for constructing the feasible region of the whole IG $\mathcal{I}(\hat{\boldsymbol{x}})$. Thereby, after solving the recourse problem, and obtaining the optimal solution $\hat{\boldsymbol{z}}^*$, Lines 9 and 10 use this vector to calculate the penalty values $M_i^{\mathcal{I}}(\hat{\boldsymbol{z}}^*)$ from the attacker's perspective, and form the parameters needed to construct the cut for this iteration. As the iterations advance, the attacker uses the collective knowledge from trying different attack scenarios and observing how the defender/operator reacts to each attack. This collective knowledge is stored in the cuts repository. Lines 11 to 13 add those cuts to the $\text{RMP}_i^{\mathcal{I}}(\hat{x})$. At any given iteration, solving the latter problem reflects the attacker's learned knowledge so far. Analogous to realistic learning, not all knowledge content is useful; this can be reflected in redundant cuts which do not improve the UB. It is worth mentioning that choosing a suitable $M_i^{\mathcal{I}}(\hat{\boldsymbol{z}}^*)$ values will lead to tighter bounds in each iteration. Line 14 updates the UB in each iteration i by solving $\text{RMP}_i^{\mathcal{I}}(\hat{x})$ that includes all the cuts stored in the repository and obtains an optimal attack vector $\hat{\boldsymbol{y}}_i$, which is used afterwards to update linking constraints (i.e., recourse decisions affected by $\hat{\boldsymbol{y}}_i$) in the recourse problem. Finally, lines 16 to 18 compare the UB and LB in each iteration and terminate if the equality condition, which is rounded to an appropriate precision, is true. In Proposition 5.2.3, we establish the finiteness of Algorithm 5.1.

Proposition 5.2.3. *For a given defence vector $\hat{\boldsymbol{x}}$, solving problem $RMP_i^{\mathcal{I}}(\hat{x})$ 5.9 yields a non-increasing UB.*

Proof. Consider a feasible defence vector $\hat{\boldsymbol{x}}$ upon which an initialized attack decision vector $\hat{\boldsymbol{y}}_0$ is chosen. According to the OAR procedure outlined in Algorithm 5.1, an optimal recourse solution $\hat{\boldsymbol{z}}^*$ is used to evaluate the penalty values $M_i^{\mathcal{I}}(\hat{\boldsymbol{z}}^*)$ to be implemented in the constraint set (5.10). Using the result from Proposition5.2.2, associated with the fact that in each iteration in Algorithm 5.1, a cut in the form of constraint (5.10) is added to the previous set of existing cuts in the repository (i.e., lines 11 to 13), and the non-negative penalty values $M_i^{\mathcal{I}}(\hat{\boldsymbol{z}}^*)$, then $\text{RMP}_i^{\mathcal{I}}(\hat{x})$ is bounded to two options based on the added cut: (1) If the cut yields a tighter bound than previously existing cuts, then $D^{\mathcal{I}} = \text{UB}_i < \text{UB}_{i-1}$; and (2) the added cut did not improve the UB (i.e., redundant constraint), then $D^{\mathcal{I}} = \text{UB}_i = \text{UB}_{i-1}$

because of the existing cuts in the repository (i.e., collective knowledge). Hence, $D^{\mathcal{I}} = \mathrm{UB}_i \leq \mathrm{UB}_{i-1}$. □

5.2.3 Optimal defence-Attack-Recourse Procedure

This subsection proposes an overall solution approach for FGs 5.3. The main idea is very similar to the attack-recourse procedure detailed in Algorithm 5.1.The main difference stems from the complexity resulting from adding the fortification layer as an extra hierarchical decision layer on top of the IG. Conceptually, the decomposition approach is intrinsically the same. In particular, the FG is decomposed into a relaxed master-problem RMP$^{\mathcal{F}}$ and a sub-problem comprised of an IG defined on a feasible defence strategy $\hat{\boldsymbol{x}}$. The relaxed master-problem RMP$^{\mathcal{F}}$ can be defined as follows:

RMP$^{\mathcal{F}}$

$$\min_{\boldsymbol{x}, D^{\mathcal{F}}} \quad D^{\mathcal{F}} \tag{5.12}$$

$$\text{s.t.} \quad D^{\mathcal{F}} \geq f(\hat{\boldsymbol{z}}_k^*) - \sum_{i \,\in\, n_{xy}} M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}_k^*)\hat{y}_i^* x_i, \ \forall\, (\hat{\boldsymbol{y}}_k^*, \hat{\boldsymbol{z}}_k^*) \in \mathcal{I}(\hat{\boldsymbol{x}}_k), \hat{\boldsymbol{x}}_k \in \mathcal{X} \tag{5.13}$$

$$\boldsymbol{x} \in \mathcal{X} \tag{5.14}$$

The direction of the objective function 5.12 for problem RMP$^{\mathcal{F}}$ reflects the defender's perspective to minimize the inflicted damage $D^{\mathcal{F}}$ through fortification. Specifically, the defender learns about the inflicted damage induced by the attacker for each feasible defence strategy through constraint 5.13. These constraints are enriched in each iteration, as the defender learns the optimal attacker observed moves, $\hat{\boldsymbol{y}}_k^* \in \mathcal{Y}(\hat{\boldsymbol{x}})$, for each feasible defence strategy enforced by constraint 5.14 and the corresponding damage on the system through non-negative penalty terms $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}_k^*)$. As it was the case with Algorithm 5.1, the convergence of the optimal defence- attack- recourse procedure outlined in Algorithm 5.2 is dependent on defining the penalty terms $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}_k^*)$ in a way that reflects the defender's perspective and depends on the recourse problem structure. This will be thoroughly explained in the numerical results in Section 5.3. In Algorithm 5.2, Lines 1-5 initialize iteration count i, UB_i, LB_i, repository of cuts in form of constraint 5.13 and a feasible fortification vector $\hat{\boldsymbol{x}}_1$, a defenceless strategy. Consequently, $\hat{\boldsymbol{x}}_1$ will be used to populate IG problem $\mathcal{I}(\hat{\boldsymbol{x}}_1)$. Moreover, Lines 6-19 repeat till UB_i and LB_i converge to pre-specified limit ϵ. Incrementing the iteration counter i is done in line 7, followed by solving IG problem $\mathcal{I}(\hat{\boldsymbol{x}}_i)$, updating UB_i which need not to be monotone (i.e., fluctuating depending on defence vector $\hat{\boldsymbol{x}}_i$)), and storing the corresponding optimal attack and recourse response $(\hat{\boldsymbol{y}}_i^*, \hat{\boldsymbol{z}}_i^*)$ in line 8. Penalty values $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}^*)$ are then updated to store cut_i 5.13 in the repository. Lines 11-13 prepare problem $\text{RMP}_i^{\mathcal{F}}(\hat{x})$ by adding all the Cuts in the repository from previous iterations. Line 14 solves problem $\text{RMP}_i^{\mathcal{F}}(\hat{x}) = \text{LB}_i$ and obtains the corresponding defence vector $\hat{\boldsymbol{x}}_i$ which subsequently updates the IG problem in line 15. The procedure repeats till the LB_i and UB_i converge, or if the equality condition, which is rounded to an appropriate precision, is true.

Remark 5.2.1. *IG problem $\mathcal{I}(\hat{\boldsymbol{x}}_i)$ can be solved using Algorithm 5.1 regardless of the convexity of problem $\mathcal{R}(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i)$. Nevertheless, if problem $\mathcal{R}(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i)$ is convex and satisfies the complementarity constraints qualifications (Dempe & Dutta 2012), the IG problem $\mathcal{I}(\hat{\boldsymbol{x}}_i)$ can be reduced to a single-level mixed-integer problem using either*

Algorithm 5.2 Optimal defence- Attack- Recourse Framework

```
Input: Problem $\mathcal{F}$
Output: An optimal solution to $\mathcal{F}$
Set iteration count $i = 0$
Set $\text{UB}_i = \infty$ and $\text{LB}_i = -\infty$
Initialize a feasible defence vector $\hat{\boldsymbol{x}}_1 \in \mathcal{X}$
Populate IG problem $\mathcal{I}(\hat{\boldsymbol{x}}_1)$
Initialize Cuts (5.13) repository
while $\text{UB}_i - \text{LB}_i > \epsilon$ do
    Set $i = i + 1$
    Solve $\mathcal{I}(\hat{\boldsymbol{x}}_i)$ (use OAR Algorithm 5.1 for non-convex recourse problems), obtain $(\hat{\boldsymbol{y}}_i^*, \hat{\boldsymbol{z}}_i^*)$ and store $\text{UB}_i = f(\hat{\boldsymbol{z}}^*)$,
    Update $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}^*)$ values for $\text{cut}_i$
    Store $\text{cut}_i$(5.13) in Cuts repository, and corresponding defence vector $\hat{\boldsymbol{x}}_i$
    for each $\text{cut}_i$ in Cuts repository do
        Add $\text{cut}_i$(5.13) to $\text{RMP}_i^{\mathcal{F}}(\hat{x})$ (5.12)
    end for
    Solve $\text{LB}_i = \text{RMP}_i^{\mathcal{F}}(\hat{x})$ and obtain a defence vector $\hat{\boldsymbol{x}}_i$
    Re-populate linking constraints in IG problem $\mathcal{I}(\hat{\boldsymbol{x}}_i)$
    if $\text{UB}_i = \text{LB}_i$ then
        Terminate with solution $(\hat{\boldsymbol{x}}_i^*, \hat{\boldsymbol{y}}_i^*, \hat{\boldsymbol{z}}_i^*)$
    end if
end while
```

KKT or duality approaches (Fakhry et al. 2022) and solved using state-of-the-art commercial solvers.

We now state and proof Propositions 5.2.4 and 5.2.5to establish the theoretical rationale of Algorithm 5.2.

Proposition 5.2.4. *For a sufficiently large and non-negative $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}_k^*)$ values, the restricted master-problem ($RMP^{\mathcal{F}}$) (5.12) provides a LB for the overall FG problem $\mathcal{F}$.*

Proof. Similar to Proposition 5.2.2, let $D^{\mathcal{F}}$ be the optimal value from solving the restricted master-problem (RMP$^{\mathcal{F}}$) (5.12). Moreover, notice that the set of constraints (5.13) is applied for optimal recourse and attack decision vectors $(\hat{\boldsymbol{z}}_k^*, \hat{\boldsymbol{y}}_k^*)$ evaluated at each feasible defence vector $\hat{\boldsymbol{x}}$. This consequently sets the right hand side of constraint (5.13) to the common objective function of value $f(\hat{\boldsymbol{z}}_k^*)$. Since we have non-negative $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}_k^*)$ values, this implies $D^{\mathcal{F}} \leq f(\hat{\boldsymbol{z}}_k^*)$ providing a LB to the FG problem $\mathcal{F}$. □

Proposition 5.2.5. *Each iteration of Algorithm 5.2 yields a non-decreasing LB for problem RMP$_i^{\mathcal{F}}$ 5.12.*

Proof. Consider a feasible defence vector $\hat{\boldsymbol{x}}$ (e.g., a defenceless strategy), upon which an IG $\mathcal{I}(\hat{\boldsymbol{x}})$ is solved to optimality using OAR procedure outlined in Algorithm 5.1, an optimal recourse solution $\hat{\boldsymbol{z}}_k^*$ is used to evaluate the penalty values $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}_k^*)$ to be implemented in the constraint set (5.13). Similar to Proposition5.2.2, associated with the fact that each iteration in Algorithm 5.2, a cut in the form of constraint (5.13) is added to the previous set of existing cuts in the repository, and the non-negative penalty values $M_i^{\mathcal{F}}(\hat{\boldsymbol{z}}_k^*)$. RMP$_i^{\mathcal{F}}$ is bounded to two options based on the added cut: (1) If the cut yields a tighter bound than previously existing cuts, then $D^{\mathcal{F}} = \text{LB}_i > \text{LB}_{i-1}$; and (2) the added cut did not improve the LB (i.e., redundant constraint), then $D^{\mathcal{F}} = \text{LB}_i = \text{LB}_{i-1}$ because of the existing cuts in the repository (i.e., collective knowledge). Hence, $D^{\mathcal{F}} = \text{LB}_i \geq \text{LB}_{i-1}$. □

Next we present three accelerating procedures that can enhance the performance of Algorithm 5.2:

- **Adding Worst Case Cut**: Upon initializing Algorithm 5.2 with a defenceless strategy, a cut is added to prevent the worst case scenario. In other words, at least one of the attacked assets must be defended. This cut will be added to RMP$^{\mathcal{F}}$ in the form of:

$$\boldsymbol{w}^T\boldsymbol{x} \geq 1. \tag{5.15}$$

- **Adding Worst Case Load Shed Cut**: In order to avoid the worst load shed, in case there are multiple attack scenarios causing the worst load shed, this cut will enable the leader to make defence strategies avoiding the worst case load shed. This cut will be added to problem RMP$^{\mathcal{F}}$ in the form of:

  $$D^{\mathcal{F}} \leq f(\boldsymbol{z}^*_{wrst}) - \epsilon, \tag{5.16}$$

  where $\boldsymbol{z}^*_{wrst}$ is the worst case recourse decision vector.

- **Heuristics** outlined in (Fakhry et al. 2022) which rank the critical assets in the network can be used to add cuts depending on the rank of the binary variables. These cuts should fasten the convergence of Algorithm 5.2.

Next, we present two applications on fortification games in which we explain how Algorithms 5.1 and 5.2 can be applied to solve FGs with convex and non-convex recourse problems.

5.3 Applications and Numerical Results

Bi-level programming (**BLP**) and tri-level programming (**TLP**) have been used extensively in determining critical infrastructure links/element for various types of networks such as electrical transmission, transportation and supply chain networks Brown et al. (2006), Babick (2009), Arroyo (2010). Specifically, BLP and TLP are used for modelling attacker-defender problems, i.e., IGs, and defender-attacker-defender models (**DAD**), i.e., FGs. Network flow problems are a typical application of DAD models such as minimum cost flow (Babick 2009), constrained shortest path problems (Lazzaro 2016) or minimizing the cost of load shedding in an electric power transmission network (Fakhry et al. 2022).

5.3.1 Protecting Critical Infrastructure

Electric grid security has been recently becoming a major concern for governments due to the interdependency of other critical infrastructures such as communications, transportation, water systems, and healthcare. A failure of the electric transmission network could cause cascading effects on other sectors leading to disruption that could spiral out of control. Attacking the critical components of the power grid may cause cascading outages and possibly a complete blackout. Consequently, identifying the critical components of the grid which may represent a high potential target for attacks, be it human-made or natural, is crucial for its safe operation and of equal importance to following adequate protection plans.

Defending electrical power grids has received significant attention from researchers in the past few years. Alguacil et al. (2014) implemented the implicit enumeration algorithm developed by Scaparra & Church (2008) in solving the DAD model for defence planning of electrical grids. First, they decomposed the TLP into BLP by taking advantage of the continuous decision variables in the lower-level using duality theory. The resultant bi-level programme was then solved by the enumeration algorithm. Wu & Conejo (2017) decomposed the DAD model into a master-problem and a sub-problem. The authors used a column and constraint generation method (**C&CG**) to iterate between master problem, providing a lower-bound, and a sub-problem, providing an upper-bound, till a predetermined convergence limit is reached. It is important to note that the algorithm they provided does not guarantee optimality. In their test cases, they found near optimal solutions to less than 3% of of the tested problems. The authors also made a comparison in terms of computational time and quality of solution between C&CG and implicit enumeration algorithms. Xiang & Wang (2018) introduced uncertainties in terms of the attacker budget to the traditional DAD model, in which the defender minimizes the expected load shed considering multiple attack scenarios. They have also implemented a C&CG algorithm to solve the DAD model. Recently, Fakhry et al. (2022) proposed three generic heuristic approaches that vary in terms of time and solution quality. These heuristics can be used to tackle a general class of tri-level programming problems that include the DAD model. The general idea behind these heuristics is to rank the impact of decision variables through successively solving a combination of relaxed linear and mixed-integer linear programmes.

Game theoretic approaches have also been used to tackle the DAD model. Holmgren et al. (2007) have paved the way in terms of how to model and conceptualize the IG models from a game theoretic perspective through a simple game model. Chen et al. (2011) have provided a generalized game framework that allows for several interactions between defenders and attackers under static and dynamic environments. They implemented two different algorithms for the defence budget. The first one deals with the allocation of a limited defence budget to critical infrastructure, while the second deals with the amount of budget needed to achieve a certain limit of expected loss. All of the previous literature had the implicit assumption of perfect information between the attacker and the defender. Ma et al. (2013) used Markovian game analysis to model the interactions between the defender and the attacker. The authors solved for the mixed strategies of both players in a state of equilibrium with perfect information and information asymmetry. Information asymmetry in this context means that the attacker is being fed false information about the cost of load shed in different areas. This might lead to targeting less-critical assets in the network. Nemati et al. (2018) succeeded in decomposing the DAD model into a single-level model. First, they used duality theory to combine the middle and lower-level resulting in a bi-level problem. Second, they visualized the attack on transmission lines as if there were virtual attackers, one for each transmission line. Each virtual attacker is trying to maximize the total inflicted damage by interdicting the corresponding transmission line. The model also incorporates Nash and Pareto equilibria conditions as linear constraints. In the end, the DAD model is solved as a single-level mixed-integer linear program.

The DAD model within the context of defending electrical power grids has the following tri-level structure: 1) the first-level represents the defender/planner who is trying to minimize load shedding in the network, and the decision variables are binary subject to a linear budgetary constraint representing the defence resources; 2) the second-level is the attacker who is in turn trying to inflict the maximum possible damage to the network by attacking the most critical lines. The decision variables in this level are also binary, while the constraints are linear and involve the decision variables of both the first- and second-levels; and 3) the third-level represents the operator model, who is trying to maintain the operational constraints of the network and to minimize the inflicted damage. The decision variables in this level are continuous, while the objective function and constraints are linear. However, the constraints are a function of the second level and third level decision variables. This renders the recourse problem in the context of protecting electrical transmission networks a linear convex problem. We will use the notation in Table 5.1 to model the DAD problem for electric grid security. We have opted for using the common notation in this field so that it will be easier for the reader to compare our model with existing studies in this area.

Hence the FG can be modelled as follows:

$$\mathcal{F}: \quad D^* = \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \min_{\boldsymbol{z}\in\mathcal{Z}(\boldsymbol{y})} f(\boldsymbol{z}) = \sum_{n\in N} \phi_n, \tag{5.17}$$

where $\boldsymbol{x} \in \mathcal{X} = \left\{\boldsymbol{x} \in \{0,1\}^L, \sum_{l\in L} x_l \leq B_f\right\}$ is the set of constraints for the leader reflecting the binary requirement, and the allowed fortification budget. The set

$\boldsymbol{y} \in \mathcal{Y}(\hat{\boldsymbol{x}}) = \left\{\boldsymbol{y} \in \{0,1\}^L : \sum_{l \in L} y_l \leq B_A,\ y_l \leq 1 - \hat{x}_l \ \ \forall\, l \in L\right\}$ includes constraints for the attacker for a feasible defence vector $\hat{\boldsymbol{x}}$ reflecting the binary requirement for the attacker's decision variables, allowed attack budget, and a constraint reflecting the assumption that the attacker cannot interdict an already defended asset. Lastly, for a corresponding attack vector $\hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}})$, the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ for the defender, i.e., operator, can be defined as follows:

$$\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}): \quad D^{\mathcal{R}}(\hat{\boldsymbol{y}}) = \min_{\boldsymbol{\delta}, \mathbf{P^g}, \mathbf{P^f}, \boldsymbol{\phi}} \quad \sum_{n \in N} \phi_n \tag{5.18}$$

$$\text{s.t.} \quad P_l^f - B_l \hat{y}_l \sum_{n \in N} A_{nl} \delta_n = 0, \qquad \forall\, l \in L \tag{5.19}$$

$$\sum_{j \in J_n} P_j^g - \sum_{l \in L} A_{nl} P_l^f + \phi_n = P_n^d, \qquad \forall\, n \in N \tag{5.20}$$

$$-\bar{P}_l^f \leq P_l^f \leq P_l^f, \qquad \forall\, l \in L \tag{5.21}$$

$$-\bar{\delta} \leq \delta_n \leq \bar{\delta}, \qquad \forall\, n \in N \tag{5.22}$$

$$0 \leq P_j^g \leq \bar{P}_j^g, \qquad \forall\, l \in L \tag{5.23}$$

$$0 \leq \phi_n \leq P_n^d, \qquad \forall\, n \in N \tag{5.24}$$

The recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ models the network operator's reaction to minimize the load shed based on the attack scenario represented in (5.18). The DC-power flow for each line is modeled in (5.19), whereas the power balance equations in each bus is represented by (5.20) (or node balance equations in other contexts). Constraints (5.21)-(5.24) are the upper and lower bounds for the lower-level decision variables. Constraints (5.24) ensure that the load shed in each consumer sector does not exceed the load at that electric bus. Given the convexity of problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, we have two methods for solving problem $\mathcal{F}$:

- **Method 1**: applying Algorithm 5.1 in solving IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$ iteratively within the context of Algorithm 5.2, hence getting the optimal solution for the overall FG Problem $\mathcal{F}$ 5.17.
- **Method 2**: applying Algorithm 5.2 directly on FG Problem $\mathcal{F}$ 5.17 after reducing IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$ into a single-level problem using either the duality or KKT approach. In particular, the recourse problem 5.18 satisfies the complementarity slackness conditions (Dempe et al. 2014) which makes the single-level reduction of IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$ equivalent to the original problem.

Both proposed methods yield an optimal solution to the FG problem $\mathcal{F}$. However, Method 2 tends to be computationally superior over Method 1. This is mainly due to the fact that Method 2 takes advantage of state-of-the-art branch-and-bound techniques existing in modern commercial solvers for solving the reduced single-level equivalent of the IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$, while Method 1 might produce redundant cuts delaying the convergence of the lower- and upper- bounds when applying Algorithm 5.1 to solve the IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$ in each iteration of Algorithm 5.2.

Method 1

We outline how Algorithm 5.1 can be used to tackle the IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$ that is needed for implementing Step 8 of Algorithm 5.2. In particular, we will explain how the penalty terms $M_i^{\mathcal{I}}(\boldsymbol{z}_k)$ are designed for fast convergence of the LB and UB in IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$.

In the context of defending electrical transmission networks, the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ 5.18 will provide a LB$_i$ in each iteration i. This will act as a learning tool till the convergence of LB$_i$ and UB$_i$. It is important to note that in order to achieve the tightest bounds each iteration, problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ should be tackled from an attacker's perspective and thus maximizing the inflicted damage on the transmission network. In each iteration i, the attacker obtains two important pieces of information: 1) the objective function value for a corresponding attack vector, $D^{\mathcal{R}}(\hat{\boldsymbol{y}})$; and 2) The vector of power flows $\boldsymbol{P^f}$ indicating the power transmitted through the electrical transmission lines into the buses. Hence, the restricted master-problem, which reflects the attacker's perspective on maximizing the inflicted damage on the electrical network, can be presented as follows:

$$\text{RMP}^{\mathcal{I}}(\hat{x})$$

$$\max_{\boldsymbol{y}, D^{\mathcal{I}}} \quad D^{\mathcal{I}} \tag{5.25}$$

$$\begin{aligned} \text{s. t.} \quad & D^{\mathcal{I}} \leq (\sum_{n \in N} \hat{\phi}^*_n)_i + \sum_{l \,\in\, L} \text{abs}((\hat{P_l^{f}}^*)_i)\ (y_l)_i \\ & \qquad \forall\ (\hat{\boldsymbol{\phi}}^*_i, \hat{\boldsymbol{P^f}}^*_i) \in \mathcal{Z}(\hat{\boldsymbol{y}}), \hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}}) \end{aligned} \tag{5.26}$$

$$\boldsymbol{y} \in \mathcal{Y}(\hat{\boldsymbol{x}}) \tag{5.27}$$

The restricted master-problem $\text{RMP}^{\mathcal{I}}(\hat{x})$ will give an UB$_i$ in each iteration i. The attacker will learn the recourse moves to lessen the inflicted damage through the common objective value $(\hat{\phi}^*_n)_i$ while penalizing the interdiction decision variables through the absolute value of the power flowing in each line, $\text{abs}((\hat{P_l^{f}}^*)_i)$, through the added cut 5.26 in each iteration i. Lastly, constraint 5.27 reflects the feasibility of the interdiction decision vector with respect to the defence strategy $\hat{\boldsymbol{x}}$. By knowing the structure of recourse problem 5.18 and examining constraints 5.19–5.20, it can be seen that the attacker's decision vector $\boldsymbol{y}$ is linked to the power flow in the electrical transmission lines $\boldsymbol{P^f}$. Thus, by associating the interdiction decision variable $(y_l)_i$ with the absolute value of the power flow in each line, abs$((\hat{P_l^{f^*}})_i)$, the attacker would be able to learn how to inflict the maximum possible damage by taking into account all recourse cuts from the previous iterations.

Method 2

In a manner similar to Method 1, RMP$^{\mathcal{F}}$ can be formed to provide an optimal solution to the FG problem $\mathcal{F}$. Nevertheless, the penalizing terms has to be adjusted to match the defender's perspective. The defender will now learn from the optimal solution of the IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$. This can be obtained by applying Algorithm 5.1 or by using the duality theory approach to reduce the bi-level structure of the IG problem into a single-level that can be solved by a commercial solver. The main difference arises from constraint (5.29) since the defender learns from the overall solution of the IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$. In particular, the shared objective function's value $(\sum_{n \in N} \hat{\phi}_n^*)_i$, in addition to penalizing the defence vector using the power flow in the electrical transmission lines, abs$(\bar{P}_l^f - (\hat{P_l^{f^*}})_i)$, and the corresponding attack vector $\hat{y}_i^*$ for each iteration i.

RMP$^{\mathcal{F}}$

$$\min_{\boldsymbol{x}, D^{\mathcal{F}}} \quad D^{\mathcal{F}} \tag{5.28}$$

$$\text{s.t.} \quad D^{\mathcal{F}} \geq (\sum_{n \in N} \hat{\phi}_n^*)_i - \sum_{l \,\in\, L} \text{abs}(\bar{P}_l^f - (\hat{P_l^{f*}})_i)\hat{y}_i^* x_i,$$

$$\forall\ (\hat{\boldsymbol{y}}_i^*, \hat{\boldsymbol{z}}_i^*) \in \mathcal{I}(\hat{\boldsymbol{x}}_i), \hat{\boldsymbol{x}}_i \in \mathcal{X} \tag{5.29}$$

$$\boldsymbol{x} \in \mathcal{X} \tag{5.30}$$

Numerical Results

Our proposed solution approaches are tested on three electrical transmission networks and compared to exact solutions obtained with an enhanced enumeration algorithm previously proposed in (Fakhry et al. 2022). Algorithms 5.1 and 5.2 have been programmed in C++ and connected with the IBM ILOG CPLEX 12.10 optimization software (Prasad et al. n.d.). All test results have been verified using an enhanced enumeration algorithm previously proposed in (Fakhry et al. 2022) and implemented using CPLEX with MATLAB R2018b. Numerical results have been carried out on an Intel Core I7 CPU (10$^{\text{th}}$ generation) at 2.6 GHZ with 16 GB of RAM and 64-bit operating system.

Five-Bus System

The first electrical transmission network is shown in Figure 5.1 and was used before in Arroyo & Galiana (2005). It consists of 6 transmission lines, 5 generators and 5 buses. The loads are specified on each bus, as well as the per unit reactance of each line. It is worth mentioning that the BMVA (Base-Mega-Volt-Ampere) and BkV (Base-kilo-Volt) are taken as 100 MVA and 138 kV, respectively. The maximum power

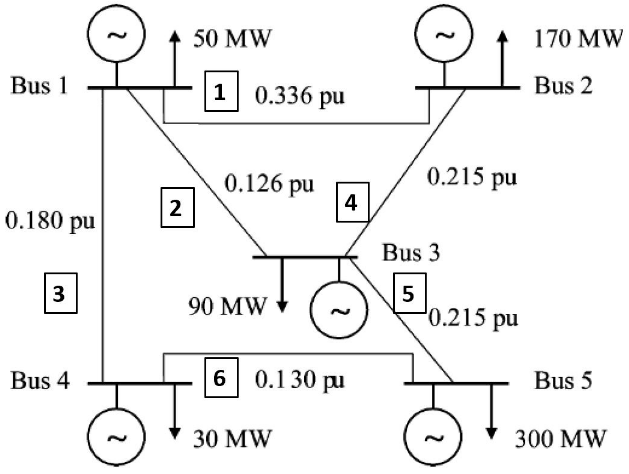


Figure 5.1: Five-Bus System Structure.

flow ($\bar{P}_l^f$) in each transmission line has been set to 100 MW, while the maximum and minimum power ($\bar{P}_j^g$, $\underline{P}_j^g$,) that a generator can produce is set to 150 and zero MW, respectively. Moreover, transmission lines are numbered (squared boxes) as per Figure 5.1.

The results summarized in Table 5.2 show the attacked line(s), objective value, and defended line(s), if relevant. We note that since we have six transmission lines in total, implying that $B_A + B_D \leq 6$, there are no feasible combinations if the aforementioned inequality is not satisfied. Furthermore, we compare the average running time of Algorithm 5.2 with the modified enumeration algorithm previously proposed in (Fakhry et al. 2022). It is worth mentioning that the enumeration algorithm has an accelerating warm-start solution technique to reach the optimal solution efficiently. In most instances, Algorithm 5.2 was able to reach to the optimal solution faster than the enumeration algorithm, where all instances took less than 1 second. We include the defended and attacked line numbers for each instance, as per figure 5.1, for validation and reproducibility. The last column in Table 5.2 indicates the number of iterations taken by Algorithm 5.2 till the lower- and upper-bounds converge.

In Table 5.3, include the *LPRank* heuristic (Fakhry et al. 2022) results for comparison purposes. It should be noted that Algorithm 5.2 performs better in terms of solution quality and run-time.

Six-Bus System

The second electrical transmission network consists of eight transmission lines, two generators and six buses (Jiang et al. 2019). The input parameters for the system's generator, load, branch data and line numbering have been taken from (Jiang et al. (2019)) for validation purposes. Table 5.4 outlines all available instances for the Six-bus system. Each instance is described in terms of the defence budget B_D, attack budget B_A, average run-time in seconds taken by Algorithm 5.2, and the average run-time taken by the enhanced enumeration algorithm. The performance of Algorithm 5.2 outperforms that of the enumeration algorithm in all instances with substantial savings in instances number 25 and 33. The last three columns in Table 5.4 indicate the defended, and attacked line numbers in the electrical transmission network for each instance, in addition to the number of iterations taken by Algorithm 5.2 to converge to the optimal solution.

Regarding the heuristic-based approaches previously proposed in Fakhry et al. (2022), *LPRank* and *HybridRank* had the same solution quality for all instances.

Both of them obtained optimal solutions for 35 instances (about 78% of the instances), while near-optimal results were obtained for the remaining 10 instances. Table 5.5 summarizes the results obtained by the *HybridRank* approach including the average run-time, which is also outperformed by Algorithm 5.2 in terms of solution quality and efficiency in all instances. Furthermore, the superiority of Algorithm 5.2 over the *HybridRank* approach in terms of run-time goes back to the number of linear and mixed-binary programmes solved by the heuristic approach. A breakdown of the run-time, objective value, defended line numbers, attacked line numbers, number of linear programmes and mixed-binary linear programmes, and solution quality for each instance are indicated in Table 5.5.

Table 5.1: Mathematical Notations for Electric Transmission Networks Fortification.

Symbol	**Description**
Indices and Sets	
J	Set of generators.
J_n	Set of generators connected to bus n.
L	Set of transmission lines.
N	Set of buses.
j	Generator index.
l	Transmission line index.
n	Bus index.
Paramters	
A_{nl}	Element of the incidence matrix equals 1 if bus n is the sending end of line l, -1 if bus n is the receiving end of line l, and 0 otherwise.
B_l	Imaginary part of admittance of line l.
B_A	Attack Budget (Number of lines to be attacked).
B_f	Fortification Budget (Number of lines to be defended).
P_n^d	Demand at bus n.
$\bar{P}_l^f$	Maximum power flow in line l.
$\bar{P}_j^g$	Maximum power a generator can produce.
$\underline{P}_j^g$	Minimum power a generator can produce.
$\bar{\delta}$	Maximum power angle for a bus.
$\underline{\delta}$	Minimum power angle for a bus.
Decision Variables	
y_l	Binary variable set to 0 if line l is attacked and 1 otherwise.
x_l	Binary variable set to 1 if line l is defended and 0 otherwise.
P_j^g	Output power from generator j.
P_l^f	Power flow in line l.
δ_n	Power angle for bus n.
Φ_n	Load shed at bus n.

Instance Num.	Def. Budget	Att. Budget	Avg. Run-time (sec)	Avg. Run-time Enum. Alg.	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. Iter.
1	0	1	0.075	0.1506	50	-	6	1
2	0	2	0.110	0.0488	150	-	5, 6	1
3	0	3	0.070	0.0454	150	-	4, 5, 6	1
4	0	4	0.066	0.0390	170	-	3, 4, 5, 6	1
5	0	5	0.136	0.0372	170	-	1, 3, 4, 5, 6	1
6	0	6	0.030	0.0350	170	-	1, 2, 3, 4, 5, 6	1
7	1	1	0.072	0.3228	50	6	5	2
8	1	2	0.131	0.1577	50	5	2, 6	2
9	1	3	0.273	0.1816	70	6	1, 4, 5	4
10	1	4	0.215	0.2406	70	6	1, 2, 4, 5	5
11	1	5	0.312	0.2131	70	6	1, 2, 3, 4, 5	6
12	2	1	0.137	0.1913	0	5, 6	-	3
13	2	2	0.147	0.3841	20	5, 6	1, 4	2
14	2	3	0.288	0.5400	20	5, 6	1, 2, 4	4
15	2	4	0.130	0.5850	20	5, 6	1, 2, 3, 4	3
16	3	1	0.197	0.2465	0	5, 6	-	3
17	3	2	0.104	0.4997	0	1, 5, 6	3, 4	3
18	3	3	0.571	1.098	0	1, 5, 6	2, 3, 4	9
19	4	1	0.140	0.3619	0	5, 6	-	3
20	4	2	0.095	0.6819	0	1, 4, 5, 6	2, 3	3
21	5	1	0.149	0.3277	10	5, 6	-	3

Table 5.2: Five-Bus System Instances using Algorithm 5.2.

Instance Num.	Def. Budget	Att. Budget	Avg. Run-time (sec)	Avg. Run-time Enum. Alg.	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. LP Solved	Optimal or Near-Optimal	Diff. to Optimal (MW)
1	0	1	0.2654	0.1506	50	-	6	8	Optimal	-
2	0	2	0.324	0.0488	150	-	5, 6	15	Optimal	-
3	0	3	0.3757	0.0454	150	-	3, 5, 6	21	Optimal	-
4	0	4	0.4261	0.0390	150	-	3, 4, 5, 6	26	Near-Optimal	20
5	0	5	0.4720	0.0372	170	-	1, 3, 4, 5, 6	30	Optimal	-
6	0	6	0.5027	0.0350	170	-	1, 2, 3, 4, 5, 6	33	Optimal	-
7	1	1	0.3371	0.3228	50	6	5	15	Optimal	-
8	1	2	0.3898	0.1577	50	6	4, 5	21	Optimal	-
9	1	3	0.4490	0.1816	70	6	1, 4, 5	26	Optimal	-
10	1	4	0.5180	0.2406	70	6	1, 2, 4, 5	30	Optimal	-
11	1	5	0.5019	0.2131	70	6	1, 2, 3, 4, 5	33	Optimal	-
12	2	1	0.3715	0.1913	0	5, 6	4	21	Optimal	-
13	2	2	0.4191	0.3841	20	5, 6	1, 4	36	Optimal	-
14	2	3	0.4526	0.5400	20	5, 6	1, 3, 4	30	Optimal	-
15	2	4	0.4798	0.5850	20	5, 6	1, 2, 3, 4	33	Optimal	-
16	3	1	0.4334	0.2465	0	3, 5, 6	4	26	Optimal	-
17	3	2	0.4586	0.4997	20	3, 5, 6	1, 4	30	Near-Optimal	20
18	3	3	0.4740	1.098	20	3, 5, 6	1, 2, 4	33	Near-Optimal	20
19	4	1	0.4584	0.3619	0	3, 4, 5, 6	1	30	Optimal	-
20	4	2	0.4971	0.6819	10	3, 4, 5, 6	1, 2	33	Near-Optimal	10
21	5	1	0.5290	0.3277	10	1, 3, 4, 5, 6	2	33	Optimal	-

Table 5.3: Five-Bus System Instances using *LPRank* Approach proposed in (Fakhry et al. 2022).

Instance Num.	Def. Budget	Att. Budget	Avg. Run-time (sec)	Avg. Run-time Enum. Alg.	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. Iter.
1	0	1	0.04	0.4785	43.63	-	5	1
2	0	2	0.146	0.5624	130	-	2, 5	1
3	0	3	0.084	0.1075	210	-	2, 3, 5	1
4	0	4	0.018	0.1906	290	-	2, 3, 4, 5	1
5	0	5	0.017	0.0549	290	-	2, 3, 4, 5	1
6	0	6	0.017	0.0917	290	-	2, 3, 4, 5	1
7	0	7	0.017	0.0455	290	-	2, 3, 4, 5, 6, 7, 8	1
8	0	8	0.015	0.0511	290	-	1, 2, 3, 4, 5, 6, 7, 8	1
9	1	1	0.071	0.3062	31.08	5	2	2
10	1	2	0.252	0.5813	95	2	4, 5	4
11	1	3	0.409	0.6368	170	2	3, 4, 5	5
12	1	4	0.052	0.5546	170	2	3, 4, 5	2
13	1	5	0.124	0.4391	210	3	2, 4, 5, 8	4
14	1	6	0.117	0.2932	210	3	2, 4, 5, 6, 7, 8	4
15	1	7	0.107	0.3143	210	3	1, 2, 4, 5, 6, 7, 8	4
16	2	1	0.09	0.1733	25	2, 5	4	3
17	2	2	0.487	0.3357	80	2, 5	3, 8	7
18	2	3	0.29	0.7541	90	3, 5	2, 4, 7	4
19	2	4	0.451	0.8664	140	3, 4	2, 5, 6, 8	7
20	2	5	0.406	0.9754	140	3, 4	2, 5, 6, 7, 8	7
21	2	6	0.435	1.1947	140	2, 3	1, 4, 5, 6, 7, 8	7
22	3	1	0.213	0.2127	20	2, 4, 5	3	4
23	3	2	0.409	0.8593	60	2, 3, 4	5, 7	6
24	3	3	0.728	1.7721	70	3, 4, 5	2, 7, 8	10
25	3	4	0.734	2.3662	70	2, 3, 4	5, 6, 7, 8	8
26	3	5	0.519	2.687	70	2, 3, 4	1, 5, 6, 7, 8	7
27	4	1	0.225	0.2558	0	2, 3, 4, 5	-	5
28	4	2	0.214	1.0196	0	2, 3, 4, 5	-	5
29	4	3	0.263	1.9554	0	2, 3, 4, 5	-	4
30	4	4	0.051	4.1178	0	2, 3, 4, 5	1, 6, 7, 8	2
31	5	1	0.182	0.3319	0	2, 3, 4, 5	-	5
32	5	2	0.414	1.5541	0	2, 3, 4, 5	-	5
33	5	3	0.208	3.5982	0	2, 3, 4, 5, 8	1, 6, 7	5
34	6	1	0.197	0.4359	0	2, 3, 4, 5	-	5
35	6	2	0.196	1.7971	0	2, 3, 4, 5, 6, 8	1, 7	4
36	7	1	0.154	0.3681	0	2, 3, 4, 5	-	5

Table 5.4: Six-Bus System Instances using Algorithm 5.2.

Instance Num.	Def. Budget	Att. Budget	Avg. Run-time (sec)	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. LP Solved	Num. MBLP Solved	Optimal or Near-Optimal	Diff. to Optimal (MW)
1	0	1	0.3206	43.63	-	5	10	0	Optimal	-
2	0	2	0.4213	130	-	2, 5	19	0	Optimal	-
3	0	3	0.5162	210	-	2, 3, 5	27	0	Optimal	-
4	0	4	0.6048	290	-	2, 3, 4, 5	34	0	Optimal	-
5	0	5	0.8306	290	-	1, 2, 3, 4, 5	40	1	Optimal	-
6	0	6	0.8862	290	-	1, 2, 3, 4, 5, 6	45	1	Optimal	-
7	0	7	1.0266	290	-	2, 3, 4, 5, 6, 7, 8	49	2	Optimal	-
8	0	8	1.0688	290	-	1, 2, 3, 4, 5, 6, 7, 8	52	2	Optimal	-
9	1	1	0.4459	31.08	5	2	19	0	Optimal	-
10	1	2	0.6504	90	5	2, 4	27	0	Near-Optimal	5
11	1	3	0.9733	170	5	2, 3, 4	34	0	Optimal	-
12	1	4	1.4011	170	5	1, 2, 3, 4, 7	40	1	Optimal	-
13	1	5	1.8878	170	5	1, 2, 3, 4, 7	45	1	Near-Optimal	40
14	1	6	2.3447	220	5	1, 2, 3, 4, 6, 7	49	1	Near-Optimal	10
15	1	7	2.9098	220	5	1, 2, 3, 4, 6, 7, 8	52	2	Near-Optimal	10
16	2	1	0.4859	25	2, 5	4	27	0	Optimal	-
17	2	2	0.6193	58.89	2, 5	34	0	0	Near-Optimal	21.11
18	2	3	0.8111	50	2, 5	1,3,4	40	0	Near-Optimal	50
19	2	4	1.0423	90	2, 5	1,3,4,6	45	0	Near-Optimal	50
20	2	5	1.2998	150	2, 5	1,3,4,6,8	49	0	Near-Optimal	10
21	2	6	1.6758	150	2, 5	1,3,4,6,7,8	52	1	Near-Optimal	10
22	3	1	0.5776	25	2, 3, 5	4	34	0	Near-Optimal	5
23	3	2	0.7013	70	2, 3, 5	4,6	40	0	Near-Optimal	10
24	3	3	0.9721	70	2, 3, 5	1,4,6	45	1	Optimal	-
25	3	4	1.3301	70	2, 3, 5	1,4,6,7	49	1	Optimal	-
26	3	5	1.8242	70	2, 3, 5	1,4,6,7,8	52	2	Optimal	-
27	4	1	0.6784	0	2, 3, 4, 5	7	40	0	Optimal	-
28	4	2	0.8819	0	2, 3, 4, 5	7,8	45	1	Optimal	-
29	4	3	1.1431	0	2, 3, 4, 5	6,7,8	49	1	Optimal	-
30	4	4	1.5092	0	2, 3, 4, 5	1,6,7,8	52	2	Optimal	-
31	5	1	0.8826	0	1, 2, 3, 4, 5	7	45	1	Optimal	-
32	5	2	1.0586	0	1, 2, 3, 4, 5	7,8	49	2	Optimal	-
33	5	3	1.2944	0	1, 2, 3, 4, 5	6,7,8	52	2	Optimal	-
34	6	1	0.9245	0	1, 2, 3, 4, 5, 6	7	49	1	Optimal	-
35	6	2	1.0564	0	1, 2, 3, 4, 5, 6	7,8	52	2	Optimal	-
36	7	1	1.0555	0	1, 2, 3, 4, 5, 6, 7	8	52	2	Optimal	-

Table 5.5: Six-Bus System Instances using *HybridRank* Approach proposed in (Fakhry et al. 2022).

Instance Num.	Def. Budget	Att. Budget	Avg. Run-time (sec)	Avg. Run-time Enum. Alg.	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. Iter.
1	0	1	2	2.70	75.63	-	18	1
2	1	1	3.5	1.61	53.21	18	15	2
3	2	1	5.1	2.25	49.45	15, 18	41	3
4	3	1	5.4	2.75	44.15	15, 18, 41	8	4
5	4	1	6.8	3.2	44.02	8, 15, 18, 41	17	5
6	5	1	10.1	4.28	31.77	8, 15, 17, 18, 41	16	6
7	6	1	11.3	5.09	17.67	8, 15, 16, 17, 18, 41	40	7
8	7	1	12	5.48	14.23	8, 15, 16, 17, 18, 40, 41	39	8
9	8	1	13.5	6.1	11.3	8, 15, 16, 17, 18, 39, 40, 41	67	9
10	9	1	14.6	6.82	10.61	8, 15, 16, 17, 18, 39, 40, 41, 67	2	10
11	10	1	15.8	7.47	9.44	2, 8, 15, 16, 17, 18, 39, 40, 41, 67	25	11
12	0	2	6.7	3.62	131.23	-	18, 41	1
13	1	2	19.3	9.22	114.72	18	8, 41	3
14	2	2	30.7	17.02	109.67	18, 41	15, 17	5
15	3	2	64.3	35.40	95.99	15, 18, 41	8, 17	10
16	4	2	76	66.82	83.93	15, 17, 18, 41	8, 16	13
17	5	2	81	79.73	51.99	8, 15, 17, 18, 41	16, 40	12
18	6	2	106	96.77	49.34	8, 15, 16, 17, 18, 41	6, 22	17
19	7	2	89	149.63	39	6, 8, 15, 16, 17, 18, 41	63, 65	14
20	8	2	100	215.53	35.8	6, 8, 15, 16, 17, 18, 41, 63	67, 80	15
21	9	2	92	336	30.9	8, 15, 16, 17, 18, 22, 41, 63, 67	68, 80	13
22	10	2	93	100	29.7	8, 15, 16, 17, 18, 22, 41, 63, 67, 68	59, 61	15
23	0	3	45.8	19.45	197.05	-	8, 18, 41	1
24	1	3	251.5	83.02	172.07	41	8, 17, 18	5
25	2	3	5450	167.84	159.86	17, 41	8, 16, 18	9
26	3	3	827	403.79	136.49	16, 17, 41	8, 18, 40	19
27	4	3	705	727.43	108.59	8, 15, 16, 18	17, 26, 41	15
28	5	3	972	1210.35	85.13	8, 15, 16, 17, 18	14, 41, 72	19
29	6	3	1008	2261.37	70.83	8, 15, 16, 17, 18, 41	3, 7, 22	20
30	7	3	1132	4135.44	56.46	8, 15, 16, 17, 18, 22, 41	14, 28, 58	21
31	8	3	967	8310.31	50.15	8, 14, 15, 16, 17, 18, 22, 41	19, 20, 40	19
32	0	4	181	50.64	246.7	-	8, 17, 18, 41	1
33	1	4	775	249.28	224.09	8	15, 16, 17, 41	4
34	2	4	3239	1135.37	205.58	16, 41	8, 15, 17, 18	15
35	3	4	4513	2389.81	158.6	8, 15, 18	16, 17, 29, 41	18
36	4	4	5314	5627.45	128.01	8, 15, 17, 18	14, 16, 41, 58	21
37	0	5	365	85.5016	297.64	-	8, 15, 16, 17, 41	1
38	1	5	2384	488.43	276.8	16	8, 15, 17, 18, 41	7
39	2	5	15321	3368.99	229.6	8, 15	16, 17, 18, 29, 41	16
40	0	6	898	149.22	305.6	-	8, 15, 16, 17, 29, 41	1

Table 5.6: IEEE 57-Bus System Instances using Algorithm 5.2.

IEEE 57-BUS System

The third electrical transmission network consists of 57 buses, 80 transmission lines and 7 generators. The single line diagram and dataset used for the 57-Bus system is available in the Appendix of Jiang et al. (2019). For this system the DAD problem consists of 160 binary variables as we have 80 transmission lines and both defender and attacker have the ability to defend/attack any of the transmission lines. Table 5.6 outlines the details of 40 instances to contrast Algorithm 5.2 results with that of the enhanced enumeration technique. Both algorithms yield the same solution quality. However, the efficiency of algorithm 5.2 in terms of run-time fell short in 30 instances as detailed in Table 5.6.

5.3.2 Capacitated Lot-Sizing Interdiction Problem with Fortification

The capacitated lot-sizing problem is considered a classic $\mathcal{NP}$-hard problem (Bitran & Yanasse 1982) that has been studied extensively in the literature. For instance, Karmarkar et al. (1987) proposed a dynamic mixed-integer programme with start-up and reservation costs that has been extended by Eppen & Martin (1987) using variable re-definition. Belvaux & Wolsey (2000) proposed a branch-and-bound system including lot-sizing specific preprocessing, cutting planes for different aspects of lot-sizing problems, in addition to cutting planes, and a lot-sizing-specific primal heuristic for models with set-up and start-up costs. The interested reader may refer to the work done by Brahimi et al. (2006) for a literature review on lot-sizing problems with a single item.

The capacitated lot-sizing problem with fortification (**CLSIPF**) does not necessarily have the adversarial nature as the previous application. However, fortification can be seen as a protective maintenance to avoid any disruptions that might occur in the planning horizon. In a manner similar to fortification, disruptions in this context might be an unexpected failure that might lead to capacity loss, thus affecting production planning. Hence, CLSIPF can be defined as the problem of optimally allocating preventive maintenance resources to a subset of time periods over the planning horizon, such that the total cost incurred from the worst-case disruptions is minimized. In this context, we are operating under the assumption that a time period cannot be disrupted if maintenance resources have been pre-allocated to it. Hence, the FG can be modelled by defining the following: 1) let $\boldsymbol{x} \in \mathcal{X}$ be the fortification decision variables, where $\boldsymbol{x} \in \mathcal{X} = \left\{\boldsymbol{x} \in \{0,1\}^{|T|}, \boldsymbol{w}^T\boldsymbol{x} \leq B_f\right\}$ is the set of constraints for the leader reflecting binary requirements and allowed maintenance budget; 2) let $\boldsymbol{y} \in \mathcal{Y}(\hat{\boldsymbol{x}}) = \left\{\boldsymbol{y} \in \{0,1\}^{|T|} : \boldsymbol{e}^T\boldsymbol{y} \leq B_A,\ y_t \leq 1 - \hat{x}_t,\ \ \forall\, t \in T\right\}$ be the virtual attacker decision variables to model the worst-case scenario of unexpected disruptions, where $\mathcal{Y}(\hat{\boldsymbol{x}})$ is the set of constraints for the virtual attacker given a feasible maintenance schedule $\hat{\boldsymbol{x}}$ reflecting the binary requirement for the attacker's decision variables, allowed attack budget, and a constraint reflecting the assumption that a time period cannot be disrupted, as long as, maintenance resources have been allocated to it; 3) the recourse decision vector $\boldsymbol{z}$, which is comprised of $\boldsymbol{B}, \boldsymbol{v}, \mathbf{I}$, and $\boldsymbol{s}$ denoting vectors of production units, set-up decisions, end of period inventory units, and shortage units in period t, respectively. Similarly, c_t, C_t, f_t, h_t, q_t are the production cost per unit, production capacity, setup cost, holding cost per unit, and shortage cost per unit in period t, respectively. The CLSIPF problem can be formally presented as:

$$\mathcal{F}: \quad D^* = \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \min_{\boldsymbol{z}\in\mathcal{Z}(\boldsymbol{y})} \sum_{t\in\mathcal{T}} c_tB_t + f_tv_t + h_tI_t + q_ts_t \tag{5.31}$$

Lastly, for a corresponding attack vector $\hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}})$, the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ can be defined as

$$\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}): \quad D^{\mathcal{R}}(\hat{\boldsymbol{y}}) = \min_{\boldsymbol{B},\boldsymbol{v},\mathbf{I},\boldsymbol{s}} \quad \sum_{t\in\mathcal{T}} c_tB_t + f_tv_t + h_tI_t + q_ts_t \tag{5.32}$$

$$\text{s.t.} \quad I_t = I_{t-1} + B_t + s_t - d_t, \qquad \forall\ t \in \mathcal{T} \tag{5.33}$$

$$B_t \leq C_tv_t, \qquad \forall\ t \in \mathcal{T} \tag{5.34}$$

$$v_t \leq 1 - \hat{y}_t, \qquad \forall\ t \in \mathcal{T} \tag{5.35}$$

$$B_t, I_t, s_t \geq 0, \qquad \forall\ t \in \mathcal{T} \tag{5.36}$$

$$v_t \in \{0, 1\}. \qquad \forall\ t \in \mathcal{T} \tag{5.37}$$

The main objective (5.32) is to minimize the total cost after disruption(s). Constraints 5.33 are for inventory balancing, while (5.34) dictate production capacity. Most importantly, constraints (5.35) prohibit production in an interdicted time period t. Lastly, constraints (5.36), and (5.37) enforce bounds and binary requirements on the recourse decision variables.

The recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ (5.32) is non-convex due to the presence of binary variables v_t, which means reducing the most two lower-levels (i.e., IG $\mathcal{I}(\hat{\boldsymbol{x}})$) into a one-level cannot be accomplished. Consequently, we are proposing to solve using Method 1; where we apply Algorithm 5.1 at the core of Algorithm 5.2.

Method 1

Applying Algorithm 5.1 to CLSIPF does not require solving the recourse problem using a specific methodology. Hence, problem 5.32 can be solved using state-of-the-art commercial solvers as a mixed-integer problem, which what we adopt in this chapter, or using dynamic programming where the time period t is the state variable. Recalling that Algorithm 5.1 iterates between a restricted master-problem and a sub-problem to solve the most two lower-levels, IG problem $\mathcal{I}(\hat{\boldsymbol{x}})$, for a given feasible maintenance schedule, much of the computational effort is spent on obtaining penalty terms $M_t^{\mathcal{I}}(\hat{\boldsymbol{z}}_i^*)$ as a function of the observed optimal recourse decision vector.,

$\hat{\boldsymbol{z}}_i^*$, in each iteration i for period t. The penalty terms reflect the virtual attacker's interest in disrupting a specific time period t. For that purpose, we introduced two linear constraints in the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ to measure the cost incurred upon targeting a specific time period. These costs are added as penalty terms to the restricted master problem $\text{RMP}^{\mathcal{I}}(\hat{x})$.

In order to determine the cost incurred upon disrupting a time period t, the following linear constraints are added to extract the needed information from the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$:

- The production quantity at period t, B_t, is decomposed into amounts as follows: $b_{tt}, \ldots, b_{t|\mathcal{T}|}, \forall\, t \in\, T$, where b_{tj} is the amount produced at period t that satisfies demand at period j, for $j \geq t$. Hence, the following constraint is added, while solving problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ in each iteration:

$$B_t = \sum_{j\, \in\, \mathcal{T}: j\, \geq\, t} b_{tj} \quad \forall\, t \in\, \mathcal{T}. \tag{5.38}$$

- In order to correctly reflect the end of period inventory, I_t, which is complementary to the aforementioned constraint, the following equation is added to the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ in each iteration:

$$I_t = \sum_{j\, \in\, \mathcal{T}: j\, >\, t} b_{tj} \quad \forall\, t \in\, \mathcal{T}. \tag{5.39}$$

Now, consider an attack occurring at period t, the cost incurred based on the solution from the recourse problem $\mathcal{R}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ can be calculated as follows:

- The produced quantity at period t will be set to zero: $B_t = 0$.
- There will be savings from eliminating the production at period t in two forms:
    - Savings due to eliminating fixed and variable costs: $f_t + c_t\ B_t$.
    - Savings due to eliminating any holding costs that would have been incurred if in fact B_t was produced: $\sum_{j\,\in\,\mathcal{T}:j\,>\,t}\sum_{l=t}^{j-1} h_l\ b_{tj}$.
- Shortage costs: $\sum_{j\,\in\,\mathcal{T}:j\,\geq\,t} q_j b_{tj}$.

Hence, the restricted master-problem $\mathrm{RMP}^{\mathcal{I}}(\hat{x})$ can now be defined from the attacker's perspective as follows:

$\mathrm{RMP}^{\mathcal{I}}(\hat{x})$

$$\max_{\boldsymbol{y},D^{\mathcal{I}}} \quad D^{\mathcal{I}} \tag{5.40}$$

$$\text{s. t.} \quad D^{\mathcal{I}} \leq f(\hat{\boldsymbol{z}}_i^*) + \sum_{t\,\in\,\mathcal{T}} M_t^{\mathcal{I}}(\hat{\boldsymbol{z}}_i^*)\ (y_t)_i \quad \forall\ \hat{\boldsymbol{z}}_i^* \in \mathcal{Z}(\hat{\boldsymbol{y}}), \hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}}), \tag{5.41}$$

$$\boldsymbol{y} \in \mathcal{Y}(\hat{\boldsymbol{x}}), \tag{5.42}$$

where $f(\hat{\boldsymbol{z}}_i^*)$ is the observed optimal cost 5.32 at iteration i, and $M_t^{\mathcal{I}}(\hat{\boldsymbol{z}}_i^*)$ is the penalty incurred from disrupting period t in iteration i, and defined as follows:

$$M_t^{\mathcal{I}}(\hat{\boldsymbol{z}}_i^*) = \sum_{j\,\in\,\mathcal{T}:j\,\geq\,t} q_j (b_{tj}^*)_i - f_t - c_t\ (B_t^*)_i - \sum_{j\,\in\,\mathcal{T}:j\,>\,t}\sum_{l=t}^{j-1} h_l\ (b_{tj}^*)_i. \tag{5.43}$$

Thus, the attacker is trying to maximize the inflicted damage from disruption (5.40). In each iteration, constraint (5.41) is added to the cuts repository, which represents the collective knowledge of the virtual attacker of the possible recourse moves.

Constraint (5.42) guarantees a feasible disruption vector, based on the fortification/maintenance decision strategy. Next, we present our numericals results from applying Algorithm 5.1 on a set of classical inventory problems previously discussed in Silver & Peterson (1985).

Instances Generation and Results

We take $|\mathcal{T}|= 10$ and randomly generate integers d_t, C_t, c_t, f_t, and q_t from uniform distribution between $[10, 210]$, $[150, 200]$, $[5, 10]$, $[44, 64]$, and $[2c_t, 3c_t]$, respectively. The holding cost, h_t, is randomly selected in the interval $[0.3, 0.5]$. These intervals follow the same parameter structure introduced by Lozano & Smith (2017). As a preliminary study, 10 random instances have been generated based on the aforementioned structure, where we solve them using Method 1. We apply Algorithm 5.1 on each of the ten instance and show the cost increase, inflicted damage, by incrementing the attacking budget. We have validated our results using brute force enumeration and dynamic programming. Our major goal from the numerical results is to show the applicability of Algorithms 5.1 and 5.2 on FGs with a convex and non-convex recourse problems. Table 5.7 shows the disrupted periods for the randomly generated ten instances. These periods have been validated using a dynamic program. Additionally, for each instance, Algorithm 5.1 was used to determine the periods with the most drastic effect on cost, under different attack/disruption budget resources, B_A .

Furthermore, in order to gauge the disruption significance on cost, Table 5.8 shows the cost incurred for each instance and the corresponding disruption resource, B_A. The first column with $B_A = 0$ provides the cost incurred to meet the demand and

Instance Num.	$B_A = 0$	$B_A = 1$	$B_A = 2$	$B_A = 3$	$B_A = 4$	$B_A = 5$	$B_A = 6$	$B_A = 7$	$B_A = 8$	$B_A = 9$	$B_A = 10$
1	-	1	1,4	4,5,8	4,5,6,8	4,5,6,7,8	3,4,5,6,7,8	1,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8,9	All
2	-	1	1,2	1,2,3	1,2,3,4	1,2,3,4,5	1,2,3,4,5,7	1,2,3,4,5,7,9	1,2,3,4,5,7,8,9	1,2,3,4,5,7,8,9,10	All
3	-	1	2,3	2,3,4	2,3,4,5	1,2,3,4,5	1,2,3,4,5,6	1,2,3,4,5,6,7	2,3,4,5,6,7,8,9	1,2,3,4,5,6,7,8,9	All
4	-	1	1,2	1,2,3	1,2,3,4	1,2,3,4,6	1,2,3,4,6,	1,2,3,4,5,6,7	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8,10	All
5	-	7	1,2	1,2,7	7,8,9,10	6,7,8,9,10	1,6,7,8,9,10	1,5,6,7,8,9,10	3,4,5,6,7,8,9,10	2,3,4,5,6,7,8,9,10	All
6	-	2	1,2	1,2,3	1,2,3,4	1,2,3,4,6	1,2,3,4,6,7	1,2,3,4,6,7,8	1,2,3,4,6,7,8	1,2,3,4,6,7,8,10	All
7	-	1	1,2	1,2,3	1,2,3,4	1,2,3,5,10	1,2,3,4,5,7	1,2,3,5,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8,10	All
8	-	2	2,4	1,2,4	1,2,4,5	1,2,4,6,7	1,2,4,5,6,7	1,2,4,5,6,7,8	1,2,4,5,6,7,8,9	1,2,3,4,5,6,7,8,10	All
9	-	1	1,7	1,2,3	2,3,4,5	1,2,3,4,5	1,2,3,4,5,6	1,2,3,4,5,6,8	1,2,3,4,5,6,7,9	1,2,3,4,5,6,7,8,9	All
10	-	1	1,2	1,2,3	5,6,7,8	4,5,6,7,8	3,4,5,6,7,8	2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8,10	All

Table 5.7: Disrupted Periods for Capacitated Lot-Sizing Interdiction Problem Instances with $|\mathcal{T}|= 10$. Results using Algorithm 5.1.

shortages without any disruptions. For any instance in Table 5.8, as the disruption budget increases, i.e., moving along the right side of the table, the cost incurred by the planner increases because of the disruption in production and inability to meet the demand and shortage costs. The entries in Table 5.8 are of the form (*Cost-Iterations*), where the number after the hyphen indicates how many iterations Algorithm 5.1 took to converge to the optimal solution.

Instance Num.	$B_A = 0$	$B_A = 1$	$B_A = 2$	$B_A = 3$	$B_A = 4$	$B_A = 5$	$B_A = 6$	$B_A = 7$	$B_A = 8$	$B_A = 9$	$B_A = 10$
1	11499.8-1	12788.8-11	13508.2-47	15166.8-122	17578.7-208	19970.7-254	22326.8-207	23929.9-123	25646-50	26924-3	28111-3
2	8036.9-1	9154.94-10	10254.1-25	11435.6-48	12259.8-11	14139.8-68	14629-73	16104.4-61	16298-24	16733.1-3	6733.1-3
3	11427.6-1	12609.3-2	13992.2-13	16091.6-93	18399.6-109	20230.4-105	21111.2-116	22405.2-78	23481.2-39	25312-9	26130-3
4	7471.5-1	8416.96-9	9439.67-27	12413.7-48	13368.8-66	14239.5-65	15295.1-61	16245-50	17663-27	18270-21	18781-3
5	10718.5-1	11539.7-11	12368.8-38	1320.5-117	14415.9-178	15870.1-179	17040.3-153	18544.7-92	20431.5-51	22436.5-12	24512-3
6	9636.7-1	10385.3-10	12300.8-34	13571.8-93	14898.8-149	15899.9-189	17187.9-145	19180.2-67	20893-28	22427-12	22655-3
7	7372.8-1	8655.01-9	11319-24	12144.9-56	13955.7-81	16619.7-64	17641.8-58	17914.3	19182-41	20949-16	21011-3
8	10417.8-1	11160.8-12	12672.1-47	14493.8-106	15999.8-169	17581.4-175	19245.5-129	21045.4-70	22730.8	24717-3	26280-2
9	9038-1	10723.3-10	11457-33	12404.6- 82	137864.7-104	15820.8-84	17388.6- 72	18431.5-54	19973-31	21216-12	21561-3
10	10363.8-1	12035.9-10	12743.3-40	13704.2- 106	15818-184	18198.1-222	20273.6-194	22329.9-116	24166-49	25220-12	26250-3

Table 5.8: Costs and Iterations till Convergence for Capacitated Lot-Sizing Interdiction Problem Instances with $|\mathcal{T}|= 10$. Results using Algorithm 5.1.

5.4 Conclusions and Future Work

This research work proposed a decomposition-based approach for solving a special class of tri-level programmes known as fortification games. Depending on the convexity of the recourse problem, we propose an algorithm that is guaranteed to terminate finitely with the optimal solution for the tri-level problem. We propose two methodologies for solving fortification games with convex recourse problems. The first one depends on reducing the most two lower-levels into a single-level and then uses a decomposition approach. The second method, while computationally expensive, uses a nested-decomposition approach to obtain the overall optimal solution. For tri-level programmes with non-convex recourse problem we use the nested-decomposition where we deploy problem-specific methodologies, such as dynamic programming, or state-of-the-art commercial solvers to tackle the recourse problem and feed the results to the proposed approach. We test our algorithm on two types of fortification games: protecting critical infrastructure of electrical transmission networks characterized with a convex recourse problem and a capacitated lot-sizing problem with fortification associated with a non-convex recourse problem.

There are several avenues for future research. Machine learning and artificial intelligence techniques can be used to extract features from the operator problem and can then be used as weights for penalizing the decision variables used in the added cuts of the decomposition approach. Combining heuristics based on the problem domain with decomposition-based approaches is a fertile area of research. Additionally, revisiting traditional branch-and-cut procedures that were applied for single-level optimization problems and combining them with the proposed decomposition approach is worth investigating.

Bibliography

Akbari-Jafarabadi, M., Tavakkoli-Moghaddam, R., Mahmoodjanloo, M. & Rahimi, Y. (2017), ‘A tri-level r-interdiction median model for a facility location problem under imminent attack’, *Computers & Industrial Engineering* **114**, 151–165.

Akgün, İ., Tansel, B. Ç. & Wood, R. K. (2011), ‘The multi-terminal maximum-flow network-interdiction problem’, *European Journal of Operational Research* **211**(2), 241–251.

Alguacil, N., Delgadillo, A. & Arroyo, J. M. (2014), ‘A trilevel programming approach for electric grid defense planning’, *Computers & Operations Research* **41**, 282–290.

Arroyo, J. M. (2010), ‘Bilevel programming applied to power system vulnerability analysis under multiple contingencies’, *IET generation, transmission & distribution* **4**(2), 178–190.

Arroyo, J. M. & Galiana, F. D. (2005), ‘On the solution of the bilevel programming formulation of the terrorist threat problem’, *IEEE transactions on Power Systems* **20**(2), 789–797.

Aussel, D. & Svensson, A. (2019), ‘Is pessimistic bilevel programming a special case of a mathematical program with complementarity constraints?’, *Journal of Optimization Theory and Applications* **181**(2), 504–520.

Babick, J. P. (2009), Tri-level optimization of critical infrastructure resilience, Technical report, Naval Postgraduate School Monterey United States.

Bard, J. F. (1991), 'Some properties of the bilevel programming problem', *Journal of optimization theory and applications* **68**(2), 371–378.

Bard, J. F. & Moore, J. T. (1990), 'A branch and bound algorithm for the bilevel programming problem', *SIAM Journal on Scientific and Statistical Computing* **11**(2), 281–292.

Belvaux, G. & Wolsey, L. A. (2000), 'bc—prod: A specialized branch-and-cut system for lot-sizing problems', *Management Science* **46**(5), 724–738.

Bitran, G. R. & Yanasse, H. H. (1982), 'Computational complexity of the capacitated lot size problem', *Management Science* **28**(10), 1174–1186.

Brahimi, N., Dauzere-Peres, S., Najid, N. M. & Nordli, A. (2006), 'Single item lot sizing problems', *European Journal of Operational Research* **168**(1), 1–16.

Brown, G., Carlyle, M., Salmerón, J. & Wood, K. (2006), 'Defending critical infrastructure', *Interfaces* **36**(6), 530–544.

Candler, W. & Norton, R. (1977), 'Multilevel programming and developing policy', *Technical Report 258, World Bank Development Research Center* .

Cappanera, P. & Scaparra, M. P. (2011), 'Optimal allocation of protective resources in shortest-path networks', *Transportation Science* **45**(1), 64–80.

Caprara, A., Carvalho, M., Lodi, A. & Woeginger, G. J. (2016), 'Bilevel knapsack with interdiction constraints', *INFORMS Journal on Computing* **28**(2), 319–333.

Chen, G., Dong, Z. Y., Hill, D. J. & Xue, Y. S. (2011), 'Exploring reliable strategies for defending power systems against targeted attacks', *IEEE Transactions on Power Systems* **26**(3), 1000–1009.

Church, R. L. & Scaparra, M. P. (2007), 'Protecting critical assets: the r-interdiction median problem with fortification', *Geographical Analysis* **39**(2), 129–146.

Delgadillo, A., Arroyo, J. M. & Alguacil, N. (2009), 'Analysis of electric grid interdiction with line switching', *IEEE Transactions on Power Systems* **25**(2), 633–641.

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

Dempe, S. & Dutta, J. (2012), 'Is bilevel programming a special case of a mathematical program with complementarity constraints?', *Mathematical programming* **131**(1), 37–48.

Dempe, S., Mordukhovich, B. S. & Zemkoho, A. B. (2014), 'Necessary optimality conditions in pessimistic bilevel programming', *Optimization* **63**(4), 505–533.

Dijkstra, E. W. et al. (1959), 'A note on two problems in connexion with graphs', *Numerische mathematik* **1**(1), 269–271.

Eppen, G. D. & Martin, R. K. (1987), 'Solving multi-item capacitated lot-sizing problems using variable redefinition', *Operations Research* **35**(6), 832–848.

Fakhry, R., Hassini, E., Ezzeldin, M. & El-Dakhakhni, W. (2022), 'Tri-level mixed-binary linear programming: Solution approaches and application in defending critical infrastructure', *European Journal of Operational Research* **298**(3), 1114–1131.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2017), ‘A new general-purpose algorithm for mixed-integer bilevel linear programs’, *Operations Research* **65**(6), 1615–1637.

Fischetti, M., Ljubić, I., Monaci, M. & Sinnl, M. (2019), ‘Interdiction games and monotonicity, with application to knapsack problems’, *INFORMS Journal on Computing* **31**(2), 390–410.

Florensa, C., Garcia-Herreros, P., Misra, P., Arslan, E., Mehta, S. & Grossmann, I. E. (2017), ‘Capacity planning with competitive decision-makers: trilevel milp formulation, degeneracy, and solution approaches’, *European Journal of Operational Research* **262**(2), 449–463.

Held, H. & Woodruff, D. L. (2005), ‘Heuristics for multi-stage interdiction of stochastic networks’, *Journal of Heuristics* **11**(5), 483–500.

Holmgren, A. J., Jenelius, E. & Westin, J. (2007), ‘Evaluating strategies for defending electric power networks against antagonistic attacks’, *IEEE Transactions on Power Systems* **22**(1), 76–84.

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

Jiang, P., Huang, S. & Zhang, T. (2019), ‘Optimal deception strategies in power system fortification against deliberate attacks’, *Energies* **12**(3), 342.

Karmarkar, U. S., Kekre, S. & Kekre, S. (1987), ‘The dynamic lot-sizing problem with startup and reservation costs’, *Operations Research* **35**(3), 389–398.

Lazzaro, G. L. (2016), Tri-level optimization algorithms for solving defender-attacker-defender network models, Technical report, Naval Postgraduate School Monterey United States.

Liu, S., Wang, M., Kong, N. & Hu, X. (2021), 'An enhanced branch-and-bound algorithm for bilevel integer linear programming', *European Journal of Operational Research* **291**(2), 661–679.

Lozano, L. & Smith, J. C. (2017), 'A backward sampling framework for interdiction problems with fortification', *INFORMS Journal on Computing* **29**(1), 123–139.

Ma, C. Y., Yau, D. K., Lou, X. & Rao, N. S. (2013), 'Markov game analysis for attack-defense of power networks under possible misinformation', *IEEE Transactions on Power Systems* **28**(2), 1676–1686.

Mahmoodjanloo, M., Parvasi, S. P. & Ramezanian, R. (2016), 'A tri-level covering fortification model for facility protection against disturbance in r-interdiction median problem', *Computers & Industrial Engineering* **102**, 219–232.

Nemati, H., Latify, M. A. & Yousefi, G. R. (2018), 'Tri-level transmission expansion planning under intentional attacks: virtual attacker approach–part i: formulation', *IET Generation, Transmission & Distribution* **13**(3), 390–398.

Prasad, A. S., Goel, T. & Kotecha, P. (n.d.), Comparative performance of constraint programming solvers: Ibm ilog cplex optimization studio and eclipse, *in* 'Artificial Intelligence and Soft Computing', p. 223.

Prince, M., Smith, J. C. & Geunes, J. (2013), ‘A three-stage procurement optimization problem under uncertainty’, *Naval Research Logistics (NRL)* **60**(5), 395–412.

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

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

Sarhadi, H., Tulett, D. M. & Verma, M. (2017), ‘An analytical approach to the protection planning of a rail intermodal terminal network’, *European Journal of Operational Research* **257**(2), 511–525.

Scaparra, M. P. & Church, R. (2012), ‘Protecting supply systems to mitigate potential disaster: a model to fortify capacitated facilities’, *International Regional Science Review* **35**(2), 188–210.

Scaparra, M. P. & Church, R. L. (2008), ‘A bilevel mixed-integer program for critical infrastructure protection planning’, *Computers & Operations Research* **35**(6), 1905–1923.

Silver, E. A. & Peterson, R. (1985), *Decision systems for inventory management and production planning*, Vol. 18, Wiley.

Smith, J. C. (2010), ‘Basic interdiction models’, *Wiley Encyclopedia of Operations Research and Management Science* .

Smith, J. C. & Lim, C. (2008), Algorithms for network interdiction and fortification games, *in* ‘Pareto optimality, game theory and equilibria’, Springer, pp. 609–644.

Tang, Y., Richard, J.-P. P. & Smith, J. C. (2016), 'A class of algorithms for mixed-integer bilevel min–max optimization', *Journal of Global Optimization* **66**(2), 225–262.

von Stackelberg, H. (2011), *Market Structure and Equilibrium*, Springer Berlin Heidelberg, Berlin, Heidelberg.

Wood, R. K. (2010), 'Bilevel network interdiction models: Formulations and solutions', *Network* **174**, 175.

Wu, X. & Conejo, A. J. (2017), 'An efficient tri-level optimization model for electric grid defense planning', *IEEE Transactions on Power Systems* **32**(4), 2984–2994.

Xiang, Y. & Wang, L. (2018), 'An improved defender-attacker-defender model for transmission line defense considering offensive resource uncertainties', *IEEE Transactions on Smart Grid* **10**(3), 2534–2546.

Xu, P. & Wang, L. (2014), 'An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions', *Computers & operations research* **41**, 309–318.

Chapter 6

Conclusion

This chapter summarizes our major findings from the main thesis Chapters. This is followed by a general discussion of how the Chapters link to each other, discussing the implications of this research and its limitation that in turn lead to future research endeavours.

6.1 Main Results

Chapter 2

Chapter 2 provided a summary for bi-level programming, which is an important building stone for solving tri-level programmes. We also attempted to clarify common misconceptions by introducing definitions and classifying highly cited and co-cited research works pertaining to tri-level programmes. This was followed up with a taxonomy of tri-level solution methodologies and applications. By systematically reviewing articles that have been influential in shaping tri-level programmes and providing a bibliometric analysis for multi-level programmes with a focus on tri-level programmes, a list of the most impactful research has been compiled. This was done through searching for possible keywords to pull relevant literature from the Web of Science core collection. A meta-analysis was done using the R bibliometrix package to extract useful knowledge from the data and depict it through intuitive visualizations. We found that the majority of developments are happening in the fields of electrical engineering. In an effort to cross the disciplines, we directed our focus to the Operations Research & Management Science (ORMS) area for a systematic review. This allowed us to clear some common misconceptions, disseminate the TLP literature to the ORMS community, and provide some definitions to structure the taxonomy of TLPs.

Chapter 3

In Chapter 3 we presented a new class of tri-level mixed integer linear programming. We discussed both its dual and KKT reformulations and presented some structural analysis properties. Given the complexity of the problem, we presented three solution approaches as well as an exact enumeration method with a warm-start for benchmarking purposes. As a case study, the solutions approaches were applied to improve the resilience of three different electrical transmission networks that varied in size. Our proposed algorithms provided optimal solutions in most of the test instances. They proved to offer a good substitute when obtaining exact solutions for large problems instances.

Chapter 4

In Chapter 4 we proposed a branch-and-bound approach for solving general problems of bi-level mixed-integer linear programmes with two assumptions and a general class of tri-level mixed-integer programmes with a convex optimization problem in their most lower-level. Furthermore, we provided a detailed literature review on the most recent efforts on developing general-purpose bi-level mixed-integer linear programmes. We have tested our algorithm on randomly generated instances that have been previously from the literature for validation. We reported on computational efficiency and provided a rich data of analytics on the solution of any BMILP instance. The reporting on the solution for BMILP instances was done on the instance level and within the instance. The instance level reported on the numbers and types of relaxation problems solved to reach the optimal solution(s). We reported the number of alternative optima if applicable. Within the instance level we reported on data specific to how the solution(s) was reached. These data include but not limited to the branching tree, number of nodes created, explored, and how it was fathomed. Furthermore, we tested our algorithm on a specific class of tri-level problems which can be reduced to a mixed-integer bi-level linear program. We focused on the application of defending electrical transmission networks. Our main contribution was to guarantee bi-level optimality, provide alternative optimal solutions if they exist, and develop a general-purpose tool that can be tuned to account for different TLP constraints or objectives. Additionally, in order to enrich the test bed of bi-level mixed-integer linear problems, we provided a Matlab live editor that converts any electrical transmission network to bi-level mixed-integer programmeinstance that can be used for enhancing the resilience of the electrical network under consideration.

Chapter 5

In Chapter 5 we proposed a decomposition-based approach for solving a special class of tri-level programmes knows as fortification games. Depending on the convexity of the recourse problem, we proposed an algorithm that is guaranteed to terminate finitely with the optimal solution for the tri-level problem. In other words, the algorithm terminates with a solution that belongs to the optimal reaction set of each decision-maker in the tri-level hierarchical structure. We also proposed two methodologies for solving fortification games with convex recourse problems. The first one depended on reducing the most two lower-levels into a single-level and then used a decomposition approach. The second method, while computationally expensive, used a nested-decomposition approach to obtain the overall optimal solution. For tri-level programmes with a non-convex recourse problem, we used the nested-decomposition where we deployed problem-specific methodologies, such as dynamic programming, or commercial solvers to tackle the recourse problem. The results were then fed to the proposed approach. We tested our algorithm on two types of fortification games: protecting critical infrastructure of electrical transmission networks characterized with a convex recourse problem and capacitated lot-sizing problem with fortification associated with a non-convex recourse problem.

6.2 General Discussion and Directions for Future Research Endeavours

In this thesis, we focused on bi-level and tri-level programming problems in terms of two main aspects: solution algorithms and applications. All Chapters have a general theme which is proposing a solution approach(es) suited for a class of tri-level programme and then applies that approach to a relevant application. The main aim of this discussion is to help the reader understand the progression of the Chapters followed by the limitation of each study, and how this might guide future research. Chapter 2, while it is placed at the beginning of the thesis; this was last Chapter to be written. The reason being is that we wanted to have a complete birds-eye view of multi-level programmes before embarking on writing this literature review paper. Chapter 3 provides efficient and very fast heuristic-based solutions, and as such, finding the optimal solution is not always guaranteed. Nevertheless, our numerical results proved the efficiency of those algorithms.The heuristic-based approaches can only be applied on a special class of tri-level programmes where we make use of the problem's special structure to provide near-optimal/optimal solutions. These heuristics are presented in a generalized way to so as to increase their applicability.

The limitations of the aforementioned study motivated us to find exact solutions and generalize the class of tri-level problems that we can solve. This led to the contributions in Chapter 4. We started working on a general class of tri-level problem by discarding the assumptions we made in Chapter 3. We proposed a branch-and-bound approach that can be used for solving tri-level mixed-integer linear problems as long as it has a convex optimization problem in its most lower-level. The proposed branch-and-bound algorithm can also be used to solve bi-level mixed-integer problems. While providing exact solutions, and alternative optima if they exist, the algorithm suffers from the curse of dimensionality. The proposed branch-and-bound algorithm is affected significantly by the number of constraints in the followers problem. We are certain that the branching rule can be significantly enhanced. We can also make use of the rich data analytics provided by our algorithm to gain insights about the problem's structure and how to solve it efficiently to optimality. This paved the way for Chapter 5. The motivation behind this Chapter was to provide an exact approach for solving tri-level programmes in an efficient way. We needed an exact algorithm to solve a tri-level programme with a non-convex recourse problem at the most lower-level. This added complexity was very challenging because the tri-level problem becomes irreducible. The only way was to design a decomposition-based approach that adds problem-specific cuts to form the feasible region. However we can only apply this algorithm on a special class of tri-level programmes to gain advantage of the problem-specific knowledge. The main advantage of that algorithm is providing an optimal solution in a computationally-efficient manner. However the penalty terms must be tuned to model the problem-specific structure. In order to tackle that, we provided a framework on how those penalty terms can be designed so the algorithm can be used in different applications.

Developing algorithms for solving tri-level programming problems is still in its infancy, and it comes with many challenges and opportunities. The use of multi-level programming is becoming more prominent due to the increase of decentralized decision-making applications. This raises the need for future research in this area pertaining to four major categories as identified in Chapter 2. Modelling is the first of those categories. As tri-level programming is a discipline that flourished because of specific applications, the majority of research works call for extensions on modelling and applications. This category was further classified into four classes: 1) competition, where it is desired to capture and model the interactions between the decision-makers at the same level, 2) complex systems, where the need arises to capture the interdependence between systems of systems, 3) dynamic systems, such as bargaining, to take into account situations where decision-makers iteratively resolve their conflict, and modelling the time dimension in TLPs, and lastly 4) incomplete information to reflect real situations where some of the information may not be available or is strategically hidden by one decision-maker.

The second category is algorithm development. Several research works call for efficient and exact algorithms capable of handling large-sized instances and complex models. It is clear that modelling realistic decision-making goes hand-in-hand with algorithm development. Parametric programming, decision diagrams, decomposition methods, meta-heuristic techniques and machine learning are interesting venues of research for solving TLPs. Using multi-parametric programming theory can be explored to develop algorithms either in the general sense or suited to specific multi-level programming applications such as dealing with multi-dimensional knapsack constraints in more than one level. Decision diagrams have been used recently to solve optimization problems and can be further investigated to be applied on discrete multi-level programming problems.

The third category is concerned with TLPs with uncertainty. Many environments where TLPs have been applied have inherent uncertainty in their parameters such as attack and defence resources, supply and demand quantities, and capacity. Optimizing TLPs with stochastic parameters is still in its early stages and further research advancements in terms of modelling and algorithms are required in this area.

The last category is theory development where the type of equilibrium between decision-makers can be investigated (e.g., generalized Nash equilibrium). Additionally, different equilibrium selection strategies can be studied similar to those proposed in Chapter 2 (e.g., sequentially optimistic). Sufficient and necessary optimality conditions for TLPs are promising areas of research. Since most TLPs are currently implemented on special network structures, it is desirable to design exact or approximate algorithms that can lead to more generic theoretical results.

Appendix A

Equivalence between RTLP-Dual and RTLP-KKT in Chapter 3

A.1 Results

In Result A.1.1 we show the two reformulations: **RTLP-Dual** and **RTLP-KKT** are equivalent.

Result A.1.1. *If ($\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*, \mathbf{u}^*$) is optimal for **RTLP-Dual**, then it has to be optimal for **RTLP-KKT**.*

Proof. Except for constraint (3.31), which represents the complementary slackness, and (3.22), which represents strong duality, it is easy to see that all other constraints are the same in **RTLP-KKT** and **RTLP-Dual**. Thus to show their equivalence we only need to show that (3.31) and (3.22) are equivalent. To do so note that $\mathbf{u}^T(A_{3\mathbf{y}}\mathbf{y} - \mathbf{b}_3) = -\mathbf{u}^T A_{3\mathbf{z}}\mathbf{z} = \mathbf{c}^T\mathbf{z} \implies (A_{3\mathbf{y}}\mathbf{y} - \mathbf{b}_3)^T\mathbf{u} = \mathbf{c}^T\mathbf{z}$ □

In Result A.1.2 we show the optimality conditions for the two reformulations.

Result A.1.2. *If* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*, \mathbf{u}^*)$ *is optimal for* ***RTLP-Dual*** *or* ***RTLP-KKT****, then* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in OP$ *is also optimal solution* ***TLP****.*

Proof. Since **TLP-L3** is a linear program defined on a convex set of constraints (primal problem), then at optimality it has to satisfy the strong duality conditions, implying that the duality gap between the primal and dual problems is zero. This condition is satisfied by including strong duality as constraint (3.22) in **RTLP-Dual**, thus by satisfying that constraint we ensure that $\mathbf{z}^* \in M(\mathbf{x}, \mathbf{y})$, the rational reaction set of the third level. Hence if $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*, \mathbf{u}^*)$ is optimal for **RTLP-Dual**, then $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ has to be in OP for **TLP**. The same result applies for **RTLP-KKT** due to the equivalence established in Theorem A.1.1. □

In Result A.1.3 we establish feasibility relationships for the two reformulations.

Result A.1.3. *If* $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{z}}, \overline{\mathbf{u}})$ *is feasible for* ***RTLP-Dual*** *or* ***RTLP-KKT****, then* $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{z}}) \in S$*,* $(\overline{\mathbf{y}}, \overline{\mathbf{z}}) \in S(\mathbf{x})$ *and* $\overline{\mathbf{z}} \in M(\mathbf{x}, \mathbf{y})$ *for* ***TLP****.*

Proof. If $(\overline{\mathbf{z}}, \overline{\mathbf{u}})$ is feasible for **RTLP-Dual**, it means that strong duality constraint (3.22) is satisfied, and optimality conditions are met for **TLP-L3**. This implies that $\overline{\mathbf{z}} \in M(\mathbf{x}, \mathbf{y})$, the rational reaction set of the third level for **TLP**. Moreover, since $\overline{\mathbf{y}}$ is feasible for **RTLP-Dual**, and $\overline{\mathbf{z}}$ is optimal for **TLP-L3**, this implies that $(\overline{\mathbf{y}}, \overline{\mathbf{z}}) \in S(\mathbf{x})$ for **TLP**. Finally, if $\overline{\mathbf{x}}$ is feasible to **RTLP-Dual** and $(\overline{\mathbf{y}}, \overline{\mathbf{z}}) \in S(\mathbf{x})$ for **TLP** then $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{z}}) \in S$ for **TLP**. □

Appendix B

Heuristics Flowcharts

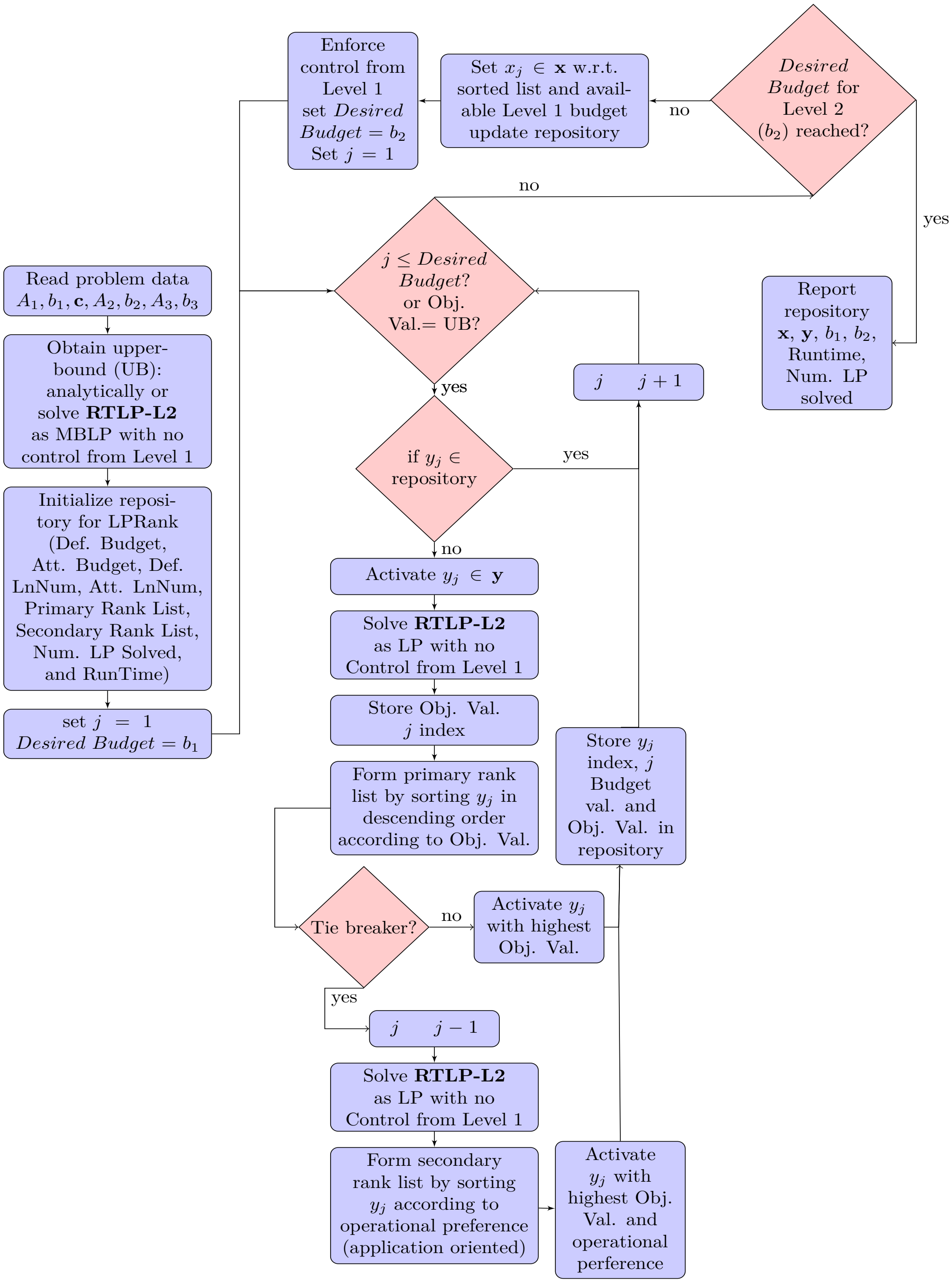

Enforce control from Level 1 set *Desired Budget* = b_2 Set j = 1
Set $x_j \in \mathbf{x}$ w.r.t. sorted list and available Level 1 budget update repository
no
Desired Budget for Level 2 (b_2) reached?
no
yes
$j \leq$ *Desired Budget?* or Obj. Val.= UB?
Read problem data $A_1, b_1, \mathbf{c}, A_2, b_2, A_3, b_3$
Report repository $\mathbf{x}$, $\mathbf{y}$, b_1, b_2, Runtime, Num. LP solved
Obtain upper-bound (UB): analytically or solve **RTLP-L2** as MBLP with no control from Level 1
yes
j $j+1$
if $y_j \in$ repository
yes
Initialize repository for LPRank (Def. Budget, Att. Budget, Def. LnNum, Att. LnNum, Primary Rank List, Secondary Rank List, Num. LP Solved, and RunTime)
no
Activate $y_j \in \mathbf{y}$
Solve **RTLP-L2** as LP with no Control from Level 1
Store Obj. Val. j index
set j = 1 *Desired Budget* = b_1
Store y_j index, j Budget val. and Obj. Val. in repository
Form primary rank list by sorting y_j in descending order according to Obj. Val.
Tie breaker?
no
Activate y_j with highest Obj. Val.
yes
j $j-1$
Solve **RTLP-L2** as LP with no Control from Level 1
Form secondary rank list by sorting y_j according to operational preference (application oriented)
Activate y_j with highest Obj. Val. and operational perference

Figure B.1: Flowchart for LPRank Heuristic.

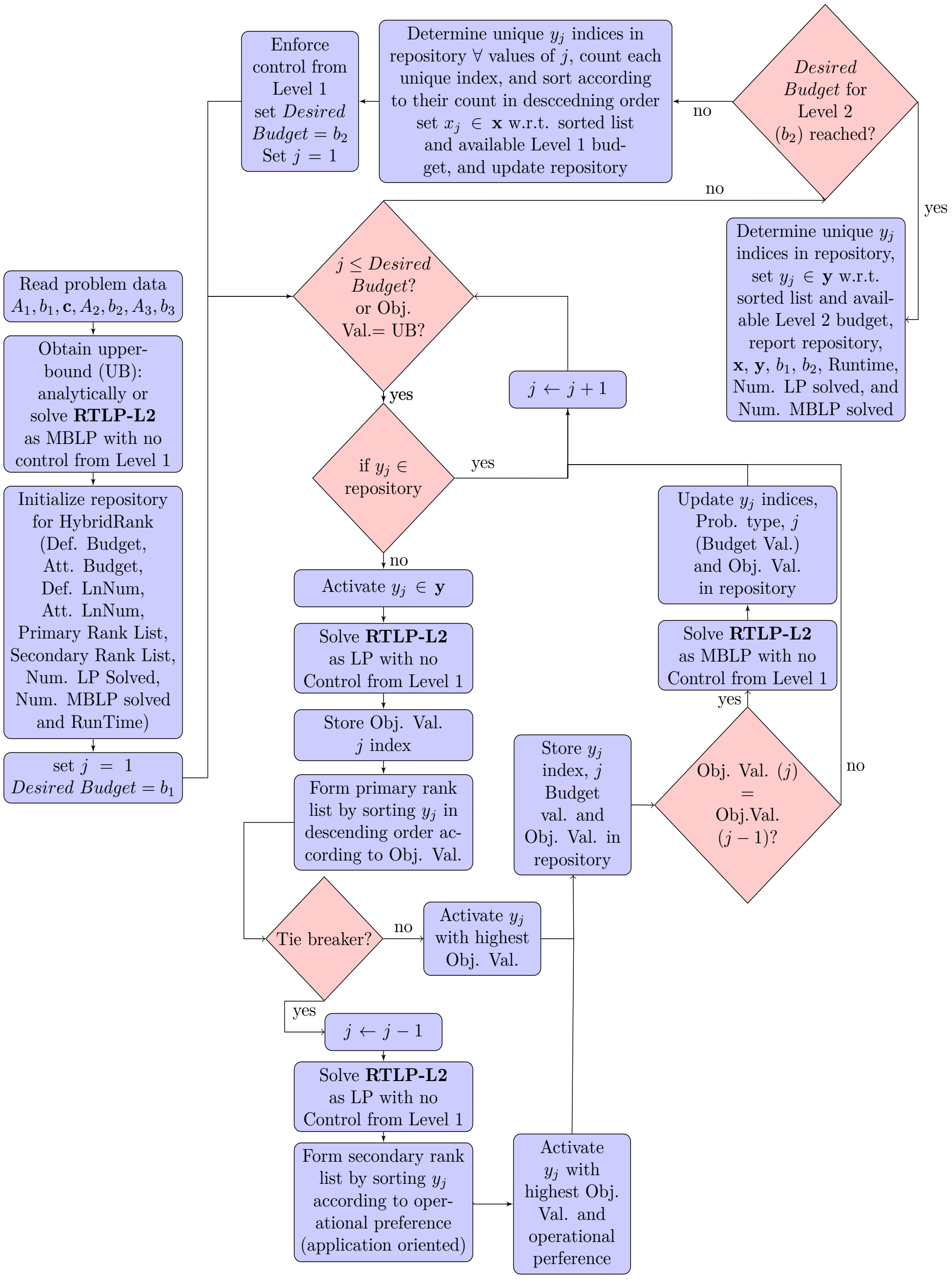

Read problem data $A_1, b_1, \mathbf{c}, A_2, b_2, A_3, b_3$
Obtain upper-bound (UB): analytically or solve **RTLP-L2** as MBLP with no control from Level 1
Initialize repository for HybridRank (Def. Budget, Att. Budget, Def. LnNum, Att. LnNum, Primary Rank List, Secondary Rank List, Num. LP Solved, Num. MBLP solved and RunTime)
set $j = 1$ *Desired Budget* $= b_1$
Enforce control from Level 1 set *Desired Budget* $= b_2$ Set $j = 1$
Determine unique y_j indices in repository ∀ values of j, count each unique index, and sort according to their count in desccedning order set $x_j \in \mathbf{x}$ w.r.t. sorted list and available Level 1 budget, and update repository
no
Desired Budget for Level 2 (b_2) reached?
no
yes
$j \leq$ *Desired Budget*? or Obj. Val.= UB?
yes
Determine unique y_j indices in repository, set $y_j \in \mathbf{y}$ w.r.t. sorted list and available Level 2 budget, report repository, $\mathbf{x}$, $\mathbf{y}$, b_1, b_2, Runtime, Num. LP solved, and Num. MBLP solved
$j \leftarrow j + 1$
if $y_j \in$ repository
yes
no
Update y_j indices, Prob. type, j (Budget Val.) and Obj. Val. in repository
Activate $y_j \in \mathbf{y}$
Solve **RTLP-L2** as LP with no Control from Level 1
Solve **RTLP-L2** as MBLP with no Control from Level 1
yes
Store Obj. Val. j index
Store y_j index, j Budget val. and Obj. Val. in repository
Obj. Val. (j) = Obj.Val. ($j - 1$)?
no
Form primary rank list by sorting y_j in descending order according to Obj. Val.
Tie breaker?
no
Activate y_j with highest Obj. Val.
yes
$j \leftarrow j - 1$
Solve **RTLP-L2** as LP with no Control from Level 1
Form secondary rank list by sorting y_j according to operational preference (application oriented)
Activate y_j with highest Obj. Val. and operational perference

Figure B.2: Flowchart for HybridRank Heuristic.

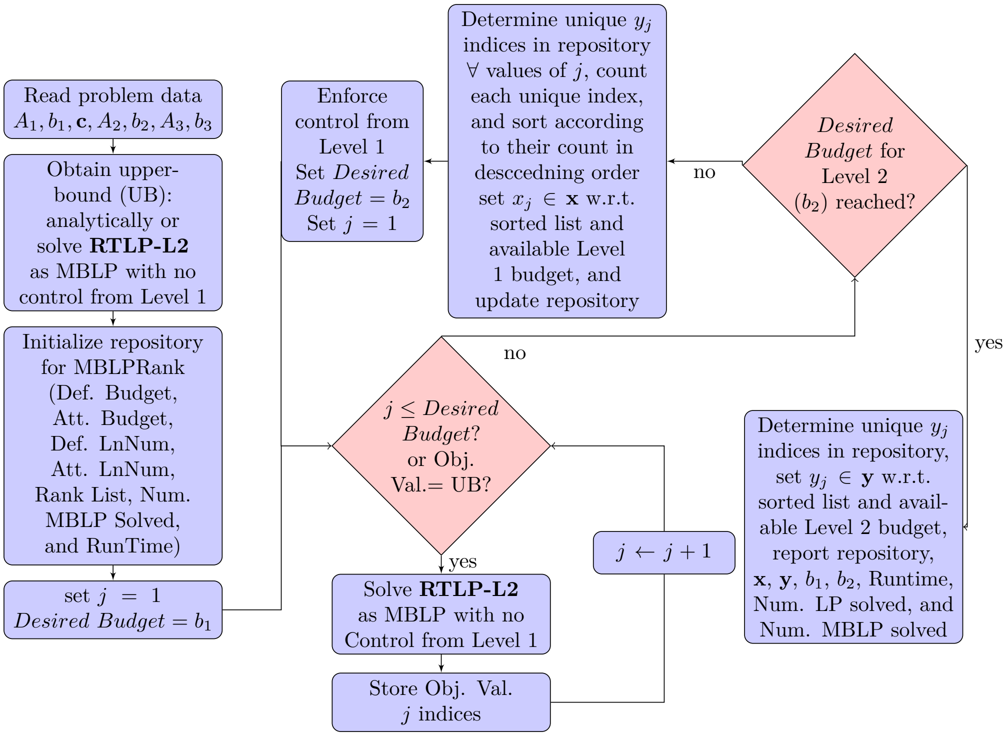

Read problem data $A_1, b_1, \mathbf{c}, A_2, b_2, A_3, b_3$
Obtain upper-bound (UB): analytically or solve **RTLP-L2** as MBLP with no control from Level 1
Initialize repository for MBLPRank (Def. Budget, Att. Budget, Def. LnNum, Att. LnNum, Rank List, Num. MBLP Solved, and RunTime)
set $j = 1$ *Desired Budget* $= b_1$
Enforce control from Level 1 Set *Desired Budget* $= b_2$ Set $j = 1$
Determine unique y_j indices in repository ∀ values of j, count each unique index, and sort according to their count in desccedning order set $x_j \in \mathbf{x}$ w.r.t. sorted list and available Level 1 budget, and update repository
no
Desired Budget for Level 2 (b_2) reached?
yes
no
$j \leq$ *Desired Budget*? or Obj. Val.= UB?
yes
Solve **RTLP-L2** as MBLP with no Control from Level 1
Store Obj. Val. j indices
$j \leftarrow j + 1$
Determine unique y_j indices in repository, set $y_j \in \mathbf{y}$ w.r.t. sorted list and available Level 2 budget, report repository, $\mathbf{x}$, $\mathbf{y}$, b_1, b_2, Runtime, Num. LP solved, and Num. MBLP solved

Figure B.3: Flowchart for MBLPRank Heuristic.

Appendix C

Multi-dimensional Knapsack Constraints

Algorithm C.1 LP Ranking with multi-dimensional Budget Constraints

```
1: procedure LPRank
2:     Read problem data: $A_1, \mathbf{b_1}, \mathbf{c}, A_2, \mathbf{b_2}, A_3, b_3$
3:     UB $\leftarrow \infty$.
4:     Solve RTLP-L2 model as MBLP without any control from upper-level (i.e., remove constraint (3.33)) and relaxed Budget $\mathbf{b_2} = n_1 \times ones(size(\mathbf{b_2}))$.
5:     UB $\leftarrow$ obj. val. .
6:     for k=1: size($\mathbf{b_1}$) do
7:         Initialize Repository(k).
8:         Add Budget Constraint k to RTLP-L2 and remove all other Budget Constraints
9:         for $j = 1 : (\mathbf{b_1})_k$ do
10:            if obj. val.$(j)$ = UB then
11:                Break.
12:            end if
13:            for $i = 1 : n_1$ do
14:                if $y_i \notin Repository(k)$ then
15:                    Set $y_i \in \mathbf{y} \leftarrow 0$.
16:                    Solve RTLP-L2 model without constraint 3.33 as an LP by setting $(\mathbf{b_2})_k \leftarrow j$.
17:                    Store obj.val. and the corresponding $y_i$'s index.
18:                end if
19:                Sort $y_i$'s indices in a descending order according to obj.val.
20:                if Two or more binary variables ($y_i$) have the same highest obj.val. then
21:                    $(\mathbf{b_2})_k \leftarrow j - 1$.
22:                    Solve RTLP-L2 model as LP without constraint 3.33.
23:                    Sort $y_i$ indices in a descending order w.r.t an operational preference.
24:                    Set $y_i \leftarrow 0$ for $i$ (index) with the highest obj.val. and operational effect.
25:                else Set $y_i \leftarrow 0$ for $i$ (index) with the highest obj.val..
26:                end if
27:                Store $y_i$ index, $j$ (Budget value), obj.val. in Repository(k).
28:            end for
29:        end for
30:    end for
31:    Determine unique $y_i$ in all Repositories (i.e., from $k = 1 : \text{size}(\mathbf{b_1})$) $\forall j$
32:    Count each unique index in all Repositories.
33:    Sort indices according to their count in a descending order.
34:    Sort indices in a greedy manner w.r.t. sorted list and Budget constraints till in-feasibility is achieved.
35:    Set $x_i \in \mathbf{x}$ in a greedy manner w.r.t sorted list and available budget $\mathbf{b_1}$.
36:    Repeat steps 7 to 27 while enforcing constraint 3.33 and available budget $\mathbf{b_2}$ after revealing $\mathbf{x}$ to get $\mathbf{y}$.
37: end procedure
```

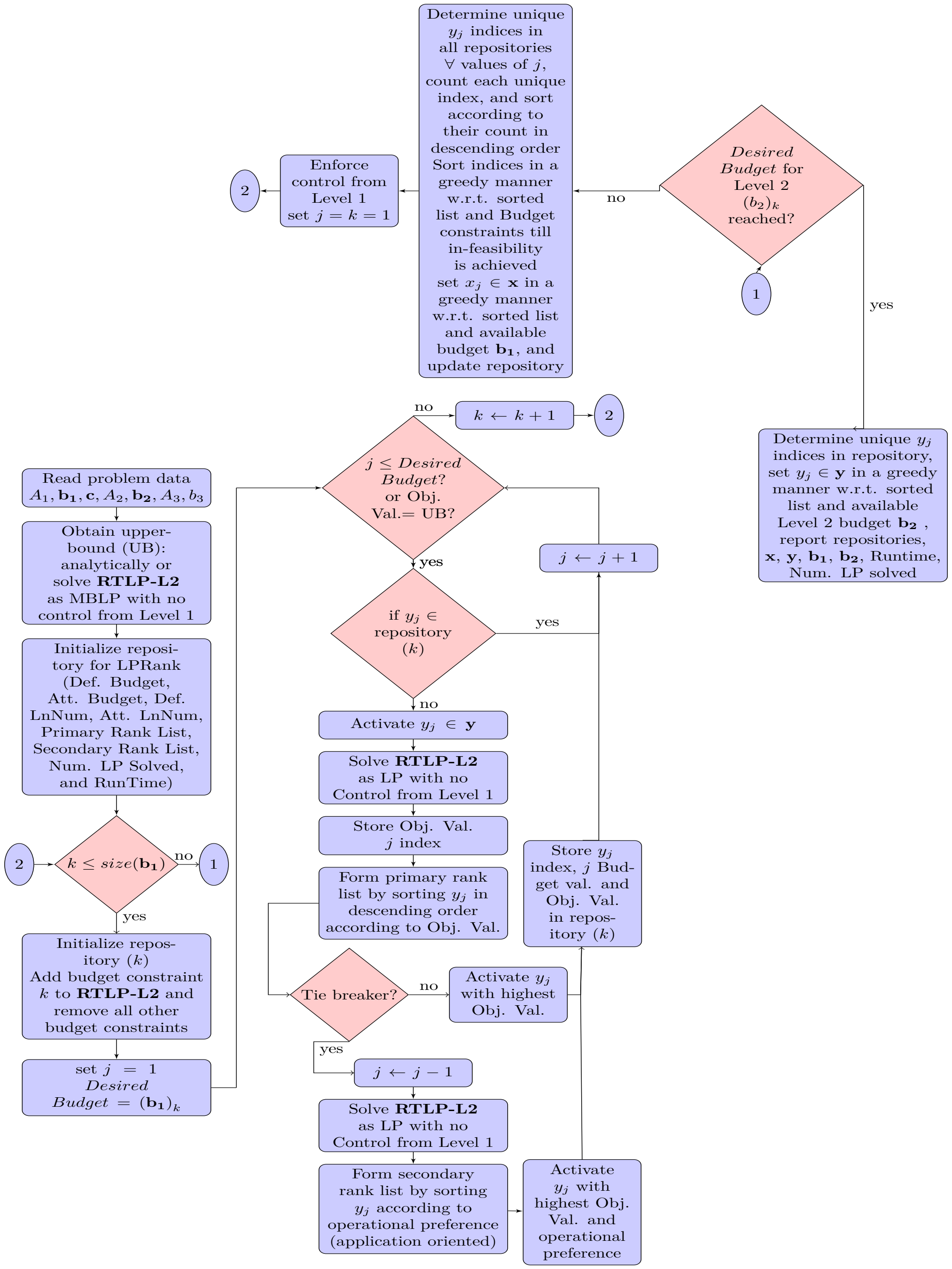

Determine unique y_j indices in all repositories $\forall$ values of j, count each unique index, and sort according to their count in descending order Sort indices in a greedy manner w.r.t. sorted list and Budget constraints till in-feasibility is achieved set $x_j \in \mathbf{x}$ in a greedy manner w.r.t. sorted list and available budget $\mathbf{b_1}$, and update repository
Enforce control from Level 1 set $j = k = 1$
2
Desired Budget for Level 2 $(b_2)_k$ reached?
no
yes
1
Determine unique y_j indices in repository, set $y_j \in \mathbf{y}$ in a greedy manner w.r.t. sorted list and available Level 2 budget $\mathbf{b_2}$, report repositories, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{b_1}$, $\mathbf{b_2}$, Runtime, Num. LP solved
no
$k \leftarrow k+1$
2
$j \leq$ *Desired Budget*? or Obj. Val.= UB?
yes
$j \leftarrow j+1$
Read problem data $A_1, \mathbf{b_1}, \mathbf{c}, A_2, \mathbf{b_2}, A_3, b_3$
Obtain upper-bound (UB): analytically or solve **RTLP-L2** as MBLP with no control from Level 1
Initialize repository for LPRank (Def. Budget, Att. Budget, Def. LnNum, Att. LnNum, Primary Rank List, Secondary Rank List, Num. LP Solved, and RunTime)
if $y_j \in$ repository (k)
yes
no
Activate $y_j \in \mathbf{y}$
Solve **RTLP-L2** as LP with no Control from Level 1
Store Obj. Val. j index
Form primary rank list by sorting y_j in descending order according to Obj. Val.
Store y_j index, j Budget val. and Obj. Val. in repository (k)
2
$k \leq size(\mathbf{b_1})$
no
1
yes
Initialize repository (k) Add budget constraint k to **RTLP-L2** and remove all other budget constraints
Tie breaker?
no
Activate y_j with highest Obj. Val.
yes
$j \leftarrow j-1$
set $j = 1$ *Desired Budget* $= (\mathbf{b_1})_k$
Solve **RTLP-L2** as LP with no Control from Level 1
Form secondary rank list by sorting y_j according to operational preference (application oriented)
Activate y_j with highest Obj. Val. and operational preference

Figure C.1: LP Ranking with multi-dimensional Budget Constraints

Appendix D

Additional Numerical Results for Chapter. 3

Instance Num.	Def. Budget	Att. Budget	Avg. Run-time (sec)	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. LP Solved	Optimal or Near-Optimal	Diff. to Optimal (MW)
1	0	1	0.2654	50	-	6	8	Optimal	-
2	0	2	0.324	150	-	5, 6	15	Optimal	-
3	0	3	0.3757	150	-	3, 5, 6	21	Optimal	-
4	0	4	0.4261	150	-	3, 4, 5, 6	26	Near-Optimal	20
5	0	5	0.4720	170	-	1, 3, 4, 5, 6	30	Optimal	-
6	0	6	0.5027	170	-	1, 2, 3, 4, 5, 6	33	Optimal	-
7	1	1	0.3371	50	6	5	15	Optimal	-
8	1	2	0.3898	50	6	4, 5	21	Optimal	-
9	1	3	0.4490	70	6	1, 4, 5	26	Optimal	-
10	1	4	0.5180	70	6	1, 2, 4, 5	30	Optimal	-
11	1	5	0.5019	70	6	1, 2, 3, 4, 5	33	Optimal	-
12	2	1	0.3715	0	5, 6	4	21	Optimal	-
13	2	2	0.4191	20	5, 6	1, 4	36	Optimal	-
14	2	3	0.4526	20	5, 6	1, 3, 4	30	Optimal	-
15	2	4	0.4798	20	5, 6	1, 2, 3, 4	33	Optimal	-
16	3	1	0.4334	0	3, 5, 6	4	26	Optimal	-
17	3	2	0.4586	20	3, 5, 6	1, 4	30	Near-Optimal	20
18	3	3	0.4740	20	3, 5, 6	1, 2, 4	33	Near-Optimal	20
19	4	1	0.4584	0	3, 4, 5, 6	1	30	Optimal	-
20	4	2	0.4971	10	3, 4, 5, 6	1, 2	33	Near-Optimal	10
21	5	1	0.5290	10	1, 3, 4, 5, 6	2	33	Optimal	-

Table D.1: Five-Bus System Instances using LPRank Approach.

Instance Num.	Def. Budget	Att. Budget	Avg. Run-time (sec)	Obj. Val. (MW)	Def. Ln. Num.	Att. Ln. Num.	Num. LP Solved	Num. MBLP Solved	Optimal or Near-Optimal	Diff. to Optimal (MW)
1	0	1	0.3206	43.63	-	5	10	0	Optimal	-
2	0	2	0.4213	130	-	2, 5	19	0	Optimal	-
3	0	3	0.5162	210	-	2, 3, 5	27	0	Optimal	-
4	0	4	0.6048	290	-	2, 3, 4, 5	34	0	Optimal	-
5	0	5	0.8306	290	-	1, 2, 3, 4, 5	40	1	Optimal	-
6	0	6	0.8862	290	-	1, 2, 3, 4, 5, 6	45	1	Optimal	-
7	0	7	1.0266	290	-	2, 3, 4, 5, 6, 7, 8	49	2	Optimal	-
8	0	8	1.0688	290	-	1, 2, 3, 4, 5, 6, 7, 8	52	2	Optimal	-
9	1	1	0.4459	31.08	5	2	19	0	Optimal	-
10	1	2	0.6504	90	5	2, 4	27	0	Near-Optimal	5
11	1	3	0.9733	170	5	2, 3, 4	34	0	Optimal	-
12	1	4	1.4011	170	5	1, 2, 3, 4, 7	40	1	Optimal	-
13	1	5	1.8878	170	5	1, 2, 3, 4, 7	45	1	Near-Optimal	40
14	1	6	2.3447	220	5	1, 2, 3, 4, 6, 7	49	1	Near-Optimal	10
15	1	7	2.9098	220	5	1, 2, 3, 4, 6, 7, 8	52	2	Near-Optimal	10
16	2	1	0.4859	25	2, 5	4	27	0	Optimal	-
17	2	2	0.6193	58.89	2, 5	34	0	0	Near-Optimal	21.11
18	2	3	0.8111	50	2, 5	1,3,4	40	0	Near-Optimal	8.89
19	2	4	1.0423	90	2, 5	1,3,4,6	45	0	Near-Optimal	50
20	2	5	1.2998	150	2, 5	1,3,4,6,8	49	0	Near-Optimal	10
21	2	6	1.6758	150	2, 5	1,3,4,6,7,8	52	1	Near-Optimal	10
22	3	1	0.5776	25	2, 3, 5	4	34	0	Near-Optimal	5
23	3	2	0.7013	70	2, 3, 5	4,6	40	0	Near-Optimal	10
24	3	3	0.9721	70	2, 3, 5	1,4,6	45	1	Optimal	-
25	3	4	1.3301	70	2, 3, 5	1,4,6,7	49	1	Optimal	-
26	3	5	1.8242	70	2, 3, 5	1,4,6,7,8	52	2	Optimal	-
27	4	1	0.6784	0	2, 3, 4, 5	7	40	0	Optimal	-
28	4	2	0.8819	0	2, 3, 4, 5	7,8	45	1	Optimal	-
29	4	3	1.1431	0	2, 3, 4, 5	6,7,8	49	1	Optimal	-
30	4	4	1.5092	0	2, 3, 4, 5	1,6,7,8	52	2	Optimal	-
31	5	1	0.8826	0	1, 2, 3, 4, 5	7	45	1	Optimal	-
32	5	2	1.0586	0	1, 2, 3, 4, 5	7,8	49	2	Optimal	-
33	5	3	1.2944	0	1, 2, 3, 4, 5	6,7,8	52	2	Optimal	-
34	6	1	0.9245	0	1, 2, 3, 4, 5, 6	7	49	1	Optimal	-
35	6	2	1.0564	0	1, 2, 3, 4, 5, 6	7,8	52	2	Optimal	-
36	7	1	1.0555	0	1, 2, 3, 4, 5, 6, 7	8	52	2	Optimal	-

Table D.2: Six-Bus System Instances using HybridRank Approach.

Instance	Def.	Att.	MEA with Warm-starting Sol.		LPRank Heuristic			HybridRank Heuristic			MBLPRank Heuristic		
Num.	Budget	Budget	Run-time (s)	Obj. Val. (MW)	Avg. Run-time (s)	Obj. Val. (MW)	Percentage of Abs. Dev.	Avg. Run-time (s)	Obj. Val. (MW)	Percentage of Abs. Dev.	Avg. Run-time (s)	Obj. Val. (MW)	Percentage of Abs. Dev.
1	**0**	**4**	**50.6**	**246.7**	**9.3**	**246.7**	**0**	**9.78**	**203.7**	**15.5**	**729**	**246.7**	**0**
2	**1**	**4**	**294.28**	**224.09**	**12.3**	**237.1**	**5.08**	**567.02**	**237.11**	**5.8**	**4792**	**224.09**	**0**
3	2	4	1135.4	205.58	15.01	224.2	9.06	610	224.69	9.2	1129	224.09	0.06
4	3	4	2389.8	158.6	16.6	188.22	18.6	647.78	151.69	4.3	2875	182.09	14.8
5	**4**	**4**	**5627.4**	**128.01**	**19.26**	**146.22**	**14.2**	**3334**	**86.98**	**32.05**	**4842**	**128.01**	**0**
6	**0**	**5**	**85.5**	**297.64**	**11.53**	**292**	**1.8**	**12.1636**	**276.8**	**7**	**1148**	**297.64**	**0**
7	1	5	488.43	276.8	15.49	297.64	7.5	587.3	297.64	7.5	5975	288.86	4.35
8	2	5	3369	229.6	16.43	244.85	6.6	623.37	225.04	1.9	17267	230.6	0.44
9	**3**	**5**	**13918**	**189.87**	**18.78**	**207.13**	**9.07**	**659.3**	**151.9**	**19.9**	**34213**	**189.87**	**0**
10	4	5	47820	169.86	20.92	165.17	2.7	3345.7	87.88	43.17	53217	189.87	11.8
11	**0**	**6**	**149.211**	**305.6**	**13.86**	**304.6**	**0.32**	**14.56**	**303.6**	**0.65**	**2639.8**	**305.6**	**0**
12	1	6	1730.3	297.6	16.76	305.6	2.68	600.78	297.89	0.097	10709	301.6	1.3
13	2	6	10401	257.6	18.83	280.89	9.04	638	225.23	12.5	27951	265.85	3.1
14	3	6	52252	222.87	20.85	213.6	4.15	3325	265.87	19.2	47084	223.87	0.45
15	**0**	**1**	**2.7**	**75.63**	**2.9**	**75.63**	**0**	**2.5**	**75.63**	**0**	**3.12**	**75.63**	**0**
16	**1**	**1**	**1.611**	**53.21**	**5.49**	**53.21**	**0**	**7.72**	**53.21**	**0**	**13.1**	**75.63**	**42.13**
17	**2**	**1**	**2.25**	**49.45**	**7.8**	**53.21**	**7.6**	**10.76**	**49.45**	**0**	**34**	**53.21**	**7.6**
18	3	1	2.75	44.15	10.05	53.21	20.5	14.1	49.45	12	145.7	53.21	20.5
19	4	1	3.2	44.02	12.23	53.21	20.8	17.4	49.45	24.9	460.1	49.45	12.3
20	7	1	5.48	14.23	19.24	17.67	24.1	27.5	17.67	24.1	4049	49.45	247.5
21	**0**	**2**	**3.62**	**131.23**	**5.237**	**131.23**	**0**	**4.9**	**82.85**	**36.8**	**20.6**	**131.23**	**0**
22	1	2	9.22	114.72	8.06	109.67	4.4	14	107.37	6.4	38.1	131.23	14.4
23	**2**	**2**	**17.02**	**109.67**	**10.148**	**109.67**	**0**	**16.77**	**114.72**	**4.5**	**58.1**	**109.67**	**0**
24	3	2	35.4	95.99	12.35	109.67	14.25	20	99.42	3.45	173	107.37	11.9
25	4	2	66.82	83.93	14.43	97.3	15.92	23.23	87.26	4	483.5	87.26	3.96
26	7	2	149.63	39	21.38	34.02	12.7	33.12	18.42	52.76	4073	63.35	62.4
27	8	2	215.53	35.8	23.76	34.02	4.9	35.9	18.42	44.5	4195	63.35	76.9
28	9	2	336	5.01	25.87	34.02	0.03	97.74	18.42	40.4	4267	63.35	105
29	10	2	512	29.7	27.56	34.02	14.5	100.6	18.42	38	4356	58.52	49.2
30	**0**	**3**	**19.4**	**197.05**	**7.5**	**197.05**	**0**	**7.3**	**141.74**	**28.1**	**228.7**	**197.05**	**0**
31	1	3	83	172.07	10.28	165.11	4	23.07	175.52	2	184	177.4	3.1
32	2	3	167.8	159.86	12.56	163.84	2.48	25.7	169.4	6	186.54	165.11	3.28
33	3	3	403.8	136.49	14.4	163.84	20.03	28.84	151.53	11	294.16	152.83	12
34	4	3	727	108.59	16.75	121.84	12.2	31.9	86.93	19.9	655.42	109.59	0.92
35	**5**	**3**	**1210**	**85.13**	**19.63**	**94.24**	**10.7**	**35.48**	**52.08**	**38.8**	**1211.2**	**85.13**	**0**
36	7	3	4135	56.46	24	47.09	16.5	41.48	34.27	39.93	4049	49.45	12.42
37	8	3	8310	50.15	26	47.09	6.1	44.12	34.27	31.6	4389	85.13	69.7

Table D.3: Proposed Heuristic Approaches Applied on 57-Bus System

Instance	Def.	Att.	MEA with Warm-starting Sol.		LPRank Heuristic			HybridRank Heuristic				MBLPRank Heuristic		
Num.	Budget	Budget	Avg. Run-time (s)	Obj. Val. (MW)	Avg. Run-time (s)	Obj. Val. (MW)	Num. of LP Solved	Avg. Run-time (s)	Obj. Val. (MW)	Num. of LP Solved	Num. of MBLP Solved	Avg. Run-time (s)	Obj. Val. (MW)	Num. of MBLP Solved
1	**0**	**1**	**0.3**	**43.63**	**0.3**	**43.63**	**10**	**0.32**	**43.63**	**10**	**0**	**0.24**	**43.6**	**1**
2	**0**	**2**	**0.17**	**130**	**0.39**	**130**	**19**	**0.42**	**130**	**19**	**0**	**0.34**	**130**	**2**
3	**0**	**3**	**0.09**	**210**	**0.45**	**210**	**27**	**0.52**	**210**	**27**	**0**	**0.47**	**210**	**3**
4	**0**	**4**	**0.16**	**290**	**0.64**	**290**	**34**	**0.6**	**290**	**34**	**0**	**0.56**	**290**	**4**
5	**0**	**5**	**0.08**	**290**	**0.57**	**290**	**40**	**0.83**	**290**	**40**	**1**	**0.65**	**290**	**5**
6	**0**	**6**	**0.09**	**290**	**0.61**	**290**	**45**	**0.89**	**290**	**45**	**1**	**0.71**	**290**	**6**
7	**0**	**7**	**0.04**	**290**	**0.65**	**290**	**49**	**1.03**	**290**	**49**	**2**	**0.75**	**290**	**7**
8	**0**	**8**	**0.05**	**290**	**0.67**	**290**	**52**	**1.07**	**290**	**52**	**2**	**0.79**	**290**	**8**
9	**1**	**1**	**0.14**	**31.08**	**0.39**	**31.08**	**19**	**0.45**	**31.08**	**19**	**0**	**0.32**	**31.08**	**2**
10	1	2	0.19	95	0.45	110.78	27	0.65	90	27	0	0.47	110.78	3
11	**1**	**3**	**0.30**	**170**	**0.55**	**170**	**34**	**0.97**	**170**	**34**	**0**	**0.72**	**170**	**4**
12	**1**	**4**	**0.31**	**170**	**0.64**	**170**	**40**	**1.4**	**170**	**40**	**1**	**1.01**	**170**	**5**
13	**1**	**5**	**0.27**	**170**	**0.6**	**220**	**45**	**1.89**	**170**	**45**	**1**	**1.41**	**220**	**6**
14	1	6	0.29	210	0.64	220	49	2.34	220	49	1	1.82	220	7
15	1	7	0.29	210	0.64	220	52	2.91	220	52	2	2.35	220	8
16	**2**	**1**	**0.27**	**25**	**0.46**	**25**	**27**	**0.49**	**25**	**27**	**0**	**0.42**	**25**	**3**
17	**2**	**2**	**0.33**	**80**	**0.51**	**70**	**34**	**0.62**	**58.89**	**34**	**0**	**0.55**	**80**	**4**
18	**2**	**3**	**0.54**	**90**	**0.56**	**90**	**40**	**0.81**	**50**	**40**	**0**	**0.79**	**90**	**5**
19	2	4	0.81	140	0.62	150	45	1.04	90	45	0	1.1	150	6
20	2	5	0.93	140	0.64	150	49	1.3	150	49	0	1.4	150	7
21	2	6	1.22	140	0.67	150	52	1.68	150	52	1	1.75	150	8
22	3	1	0.22	20	0.5	25	34	0.58	25	34	0	0.54	25	4
23	3	2	1.05	60	0.56	70	40	0.7	70	40	0	0.69	70	5
24	**3**	**3**	**1.05**	**70**	**0.61**	**70**	**45**	**0.97**	**70**	**45**	**1**	**0.93**	**70**	**6**
25	**3**	**4**	**2.58**	**70**	**0.64**	**70**	**49**	**1.33**	**70**	**49**	**1**	**1.28**	**70**	**7**
26	**3**	**5**	**3.05**	**70**	**0.67**	**70**	**52**	**1.82**	**70**	**52**	**2**	**1.61**	**70**	**8**
27	4	**1**	**0.46**	**0**	**0.57**	**0**	**40**	**0.68**	**0**	**40**	**0**	**0.66**	**0**	**5**
28	4	**2**	**1.17**	**0**	**0.65**	**0**	**45**	**0.88**	**0**	**45**	**1**	**0.83**	**0**	**6**
29	4	**3**	**2.69**	**0**	**0.64**	**0**	**49**	**1.14**	**0**	**49**	**1**	**1.06**	**0**	**7**
30	4	**4**	**4.23**	**0**	**0.71**	**0**	**52**	**1.51**	**0**	**40**	**0**	**1.34**	**0**	**8**
31	**5**	**1**	**0.43**	**0**	**0.64**	**0**	**45**	**0.88**	**0**	**45**	**1**	**0.74**	**0**	**6**
32	**5**	**2**	**1.77**	**0**	**0.67**	**0**	**49**	**1.06**	**0**	**49**	**2**	**0.89**	**0**	**7**
33	**5**	**3**	**3.55**	**0**	**0.71**	**0**	**52**	**1.29**	**0**	**52**	**2**	**1.07**	**0**	**8**
34	**6**	**1**	**0.36**	**0**	**0.66**	**0**	**49**	**0.92**	**0**	**49**	**1**	**0.77**	**0**	**7**
35	**6**	**2**	**1.99**	**0**	**0.69**	**0**	**52**	**1.06**	**0**	**52**	**2**	**0.87**	**0**	**8**
36	**7**	**1**	**0.4**	**0**	**0.71**	**0**	**52**	**1.06**	**0**	**52**	**2**	**0.79**	**0**	**8**

Table D.4: Proposed Heuristic Approaches Applied on 6-Bus System