Exploiting Local Optimality and Strong Inequalities for Solving Bilevel Combinatorial and Submodular Optimization Problems

by

Xueyu Shi

B.S. in Mathematics, University of Science and Technology of China, 2014

M.S. in Applied Mathematics, University of Science and Technology of China, 2016

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Xueyu Shi

It was defended on

September 27 2021

and approved by

Oleg A. Prokopyev, Ph.D., Professor, Department of Industrial Engineering

Bo Zeng, Ph.D., Associate Professor, Department of Industrial Engineering

Jayant Rajgopal, Ph.D., Professor, Department of Industrial Engineering

Ted K. Ralphs, Ph.D., Professor, Department of Industrial and Systems Engineering, Lehigh University

Dissertation Directors: Oleg A. Prokopyev, Ph.D., Professor, Department of Industrial Engineering

Bo Zeng, Ph.D., Associate Professor, Department of Industrial Engineering

Exploiting Local Optimality and Strong Inequalities for Solving Bilevel Combinatorial and Submodular Optimization Problems

Xueyu Shi, PhD

University of Pittsburgh, 2021

Bilevel combinatorial and submodular optimization problems arise in a broad range of real-life applications including price setting, network design, information gathering, viral marketing, and so on. However, the current state-of-the-art solution approaches still have difficulties to solve them exactly for many broad classes of practically relevant problems. In this dissertation, using the concepts of local optimality and strong valid inequalities, we explore the fundamental mathematical structure of these problems and boost the computational performance of exact solution methods for these two important classes of optimization problems.

In our initial study, we focus on a class of bilevel spanning tree (BST) problems, motivated by a hierarchical (namely, bilevel) generalization of the classical minimum spanning tree problem. We show that depending on the type of the objective function involved at each level, BST can be solved to optimality either in polynomial time by a specialized algorithm or via a mixed-integer linear programming (MILP) model solvable by an off-the-shelf solver. The latter case corresponds to an NP-hard class of the problem.

Our second study proposes a hierarchy of upper and lower bounds for the bilevel problems, where the follower's variables are all binary. In particular, we develop a generalized bilevel framework that explores the local optimality conditions at the lower level. Submodularity and disjunctive-based approach are then exploited to derive strong MILP formulations for the resulting framework. Computational experiments indicate that the quality of our newly proposed bounds is superior to the current standard approach. Furthermore, we generalize our aforementioned results for BST and show that the proposed bounds are sharp for bilevel matroid problems.

Finally, to address the computational challenges in the submodular maximization problem, we present the polyhedral study of its mixed 0–1 set. Specifically, we strengthen some existing results in the literature by finding two families of facet-defining inequalities through

the lens of sequence independent lifting. We further extend the scope of this work and describe the multi-dimensional sequence independent lifting for a more complex set. The developed polyhedral results complement the classical results from the literature for the mixed 0–1 knapsack and single-node flow sets.

Table of Contents

List of Tables

List of Figures

Preface

First and foremost, I would like to thank my advisors, Dr. Oleg Prokopyev and Dr. Bo Zeng. Without their dedicated support and guidance, the research in this dissertation would not be possible. During our many discussions over the last five years, Dr. Prokopyev could always point out interesting ideas and give his constructive advice to solve my difficulties with my research. I am also deeply thankful for Dr. Prokopyev dedicating so much time teaching me how to write research papers, how to do a good presentation and helping me integrate into the US culture. Dr. Zeng has been a very supportive and enlightening advisor, he gave me the freedom for selecting the research topic and always shared his sharp insights and invaluable experience on research directions, methodologies as well as research ethics. His passion and attitude for the scientific research encourage me to work on hard problems and pursue the original research. Dr. Prokopyev and Dr. Zeng are not only the advisors for me, but mentors and best friends during my PhD study. It is great honor to work and learn from them, which will definitely have a long term effect on my future career.

I would also like to thank Dr. Jayant Rajgopal and Dr. Ted Ralphs for serving on my dissertation defense committee. To Dr. Rajgopal, thank you for giving helpful advice and for taking your time to be on my committee. To Dr. Ralphs, I am so grateful to work with you on the bilevel projects. Your insightful comments and discussions helped me to shape my view of integer programming and bilevel optimization. I would also like to express my special thanks to Dr. Egon Balas, Dr. Gérard Cornuéjols and Dr. Andrés Gómez for their courses and guidance that lead me to the world of Integer Programming.

Thank you to my fellow past and present graduate students - Ruichen Sun, Yanfei Chen, Kai He, Shan Gong, Wei Wang, Liang Xu, Chaosheng Dong, Yuwen Yang, Zhaohui Geng, Colin Gillen, Hosein Zare, Juan Borrero, Ke Ren, Jing Yang, Yijia Wang, Shadi Sanoubar, Moataz Elsisy, Moataz Abdulhafez and among others.

Lastly, thank my family for all the unconditioned love and constant support. I am always proud to be the son of my parents and I cannot thank them enough. Thank my wife, Chujun, for her patience and understanding over these years. We established our home in Pittsburgh, and shared all those precious memory together. Looking forward to going through our next life journey!

1.0 Introduction

Over the last fifty years, numerous research efforts by the optimization community have been focused on the theoretical analysis and the development of computational methods for mixed-integer linear programming problems (MILPs). With the development of this important discipline, the current state-of-the-art solvers (e.g., CPLEX [51], Gurobi [47]) can tackle broad classes of MILPs with millions of decision variables and constraints. However, there exist many practically important real-life systems that require optimization models with integer decision variables that are not necessarily MILPs. In this dissertation we consider two classes of such optimization problems, namely, bilevel optimization and submodular maximization problems.

1.1 Bilevel Optimization

In bilevel optimization problems [27, 30], two independent decision-makers (referred to as the leader and the follower) with their own distinct objective functions are involved in a hierarchical decision-making process. The leader (the upper-level decision-maker) acts first, and then the follower (the lower-level decision-maker) determines the response in terms of their own optimization model, whose feasible region and objective function are parameterized on the leader's decision. Importantly, the follower's response also affects the leader's objective function. This is the reason the leader must take the follower's possible reactions into account when maximizing/minimizing their own benefits/costs.

Meanwhile, in the lower-level optimization problem, there may exist multiple optimal solutions, which result in different objective function values for the leader. If the follower selects the solution from the lower-level optimal solution set that is most favorable for the leader, then we often refer to this strategy as the optimistic formulation of the problem. The alternative strategy typically considered in the literature is the pessimistic formulation of the problem, where the follower chooses the least favorable solution for the leader from the lower-level optimal solution set.

Due to a broad range of important applications including interdiction [16, 32, 52], price setting [18, 68], and network design [13, 40, 119], among others, bilevel optimization problems have been extensively studied in the past two decades. Most of the research efforts have been focused on bilevel problems, where the lower-level problem is a linear optimization problem. The latter assumption allows for application of the necessary and sufficient optimality conditions (strong duality and complementary slackness) which, in turn, can be used to reduce the original bilevel problem into a single-level mixed integer optimization problem [10, 126]. In recent years there has been increased interest in the exact solution methods for generic bilevel problems, where the lower-level problem involves integer decision variables; see, e.g., [36, 77, 109, 128]. Integer decisions at the lower level require application of somewhat more sophisticated methods. In particular, one of the primary examples is MibS [32, 104], an open-source solver, that exploits advanced cutting plane based approaches [35, 36, 103] within a branch-and-bound framework.

Solving mixed-integer bilevel linear optimization problems (MIBLP) to optimality is quite challenging. The general MIBLP problem is shown to be hard in the complexity class of Σ_2^p [22, 75]. It implies that the problem cannot be reducible to a mixed-integer linear problem in polynomial time unless the hierarchy collapse to class NP; i.e., $\Sigma_2^p = \mathsf{NP}$. In terms of the computational practice, MibS [32, 104], an open source solver for the generic MIBLP problems, can solve medium-sized problems with up to several hundred integer decision variables at the lower level. We note that the initial upper and lower bounds within branch-and-bound and branch-and-cut frameworks are of critical importance for their overall performance. However, the traditional bounding approach based on the single-level relaxation for the generic bilevel problem typically yields relatively poor bounds.

1.2 Submodular Function Maximization

Submodularity is a property of set functions, i.e., $f : 2^N \to \mathbb{R}$, where N is the ground set. We say f is a submodular function if $f(S) + f(T) \geq f(S \cap T) + f(S \cup T)$ for any $S, T \subseteq N$. Optimization with submodular functions appears in several real-world applications, e.g., viral marketing [57, 93], risk-averse [62, 118], data summarization [74, 106], information gathering [66, 67], etc. In particular, many discrete optimization problems arising in the machine learning and artificial intelligence areas are essentially submodular optimization problems [33]. From its definition, the submodular maximization problem could be viewed, in a sense, as a discrete version of an easy concave maximization problem. However, the submodular maximization problem is, in general, NP-hard [76, 86]. In contrast, according to [76], the submodular minimization problem can be converted through Lovasz extension into a convex minimization problem, which is computationally tractable [89, 97]. Such drastically different results from the theoretical perspective and the great need from various applications have inspired a number of research studies on submodular optimization in recent decades, in particular, on its difficult maximization version [19, 39, 65].

When the submodular function is monotone (i.e., $f(S) \leq f(T)$ for any $S \subseteq T$), Nemhauser and Wolsey [87] propose a tight $(1 - 1/e)$-approximation greedy algorithm that incrementally includes into S an element with the largest improvement. By generalizing this classical result, a number of approximation algorithms have been developed in recent decades for the submodular maximization problem in various settings. We refer the readers to [19, 39, 65] for detailed surveys. As for the specific submodular function in the form of (4.2), Yu and Ahmed [121] propose an approximation algorithm for the expected utility maximization problem with a knapsack constraint that yields an approximation ratio better than $(1 - 1/e)$. In [5], Atamtürk and Gómez develop an 1/2-approximation algorithm for maximizing a class of the submodular function over the vertices of $\text{conv}(\mathcal{X})$ with general integer variable x.

In recent years, there are several celebrated results that provide a better understanding of the mathematical structure of submodular optimization problems. Ahmed and Atamtürk [2] employ the lifting technique to derive the first set of strong lifted inequalities for a class of submodular maximization problems. Yu and Ahmed [122] generalized their approach to a more difficult set involving one additional knapsack constraint. Besides the polyhedral results of the submodular maximization set in [2, 122], Wu and Küçükyavuz [117] propose the optimality cuts for the two-stage stochastic submodular maximization problem. A decomposition algorithm with several classes of valid inequalities [118] is then developed for a class of risk-averse submodular maximization problem in their subsequent work. For other polyhedral studies related to the submodular minimization set and the submodular knapsack polytope, we refer the readers to [7, 8, 60, 76, 123].

However, the computational experiments in [2] for a class of submodular functions indicate that the current state-of-the-art exact approaches can only deal with small-sized problems with up to one hundred binary decision variables. On the other hand, the best approximation ratio for the unconstrained submodular maximization problem is $1-1/e \approx 0.63$ when the submodular function is monotone [87]. We note that the ratio value of 0.63 implies that the worst-case bound given by the approximation algorithm is relatively weak.

1.3 Contribution & Outline

In this dissertation, we exploit the local optimality conditions and strong valid inequalities for solving bilevel combinatorial optimization and submodular maximization problems. Furthermore, we find some interesting connections between the bilevel optimization and submodular optimization, which enables us to develop advanced algorithms and approaches for the considered problems.

In particular, Chapter 2 studies a class of bilevel spanning tree problems (BSTs) that involve two independent decision-makers (DMs), the leader and the follower with different objectives, who jointly construct a spanning tree in a graph. The leader selects an initial subset of edges that do not contain a cycle, from the set under her control. The follower then selects the remaining edges to complete the construction of a spanning tree, but optimizes his own objective function. We study BSTs with the sum- and bottleneck-type objective functions for the DMs under both the optimistic and pessimistic settings. The polynomial-time algorithms are then proposed in both optimistic and pessimistic settings for BSTs in which at least one of the DMs has the bottleneck-type objective function. For BST with the sum-type objective functions for both the leader and the follower, we provide an equivalent single-level MILP formulation. A computational study is then presented to explore the efficacy of our reformulation.

In Chapter 3, we consider a class of mixed integer bilevel linear optimization problems in which the decision variables of the lower-level (follower's) problem are all binary. We propose a general modeling and solution framework motivated by the practical reality that in a bilevel problem, the follower does not always solve their optimization problem to optimality. They may instead implement a locally optimal solution with respect to a given upper-level decision. Such scenarios may occur when the follower's computational capabilities are limited, or when the follower is not completely rational. Our framework relaxes the typical assumption of perfect rationality that underlies the standard modeling framework by defining a hierarchy of increasingly stringent assumptions about the behavior of the follower. Namely, at level k of this hierarchy, it is assumed that the follower produces a k-optimal solution. Associated with this hierarchy is a hierarchy of upper and lower bounds that are in fact valid for the classical case in which complete rationality of the follower is assumed.

We exploit submodularity and disjuctive approach to develop mixed integer linear programming (MILP) formulations for the resulting optimization problems. Extensive computational results are provided to demonstrate the effectiveness of the proposed MILP formulations and the quality of the bounds produced. The latter are shown to dominate the standard approach based on a single-level relaxation at a reasonable computational cost.

Furthermore, we explore a general class of bilevel matroid problems for which 2-optimal lower-level solutions imply global optimality that is, the follower is fully rational. We also show that the bilevel spanning tree problem considered in Chapter 2 is a special case of the bilevel matroid problem. Single-level MILP formulations are further revisited and investigated for several variants of the bilevel spanning tree problems.

Finally, in Chapter 4, we study the polyhedral structure of a mixed 0-1 set arising from the submodular maximization problem, given by $P = \{(w, x) \in \mathbb{R} \times \{0, 1\}^n : w \leq f(x), x \in \mathcal{X}\}$, where submodular function $f(x)$ is represented by a concave function composed with a linear function, and $\mathcal{X}$ is the feasible region of binary variables x. For $\mathcal{X} = \{0, 1\}^n$, two families of facet-defining inequalities are proposed for the convex hull of P through restriction and lifting using submodular inequalities. When $\mathcal{X}$ involves multiple disjoint cardinality constraints, we propose a new class of facet-defining inequalities for the convex hull of P through multidimensional sequence independent lifting. The derived polyhedral results not only strengthen and generalize some existing developments in the literature, but are also linked to the classical results for the mixed 0-1 knapsack and single-node flow sets. Our computational study on a set of randomly generated submodular maximization instances demonstrates the superiority of the proposed facet-defining inequalities within a branch-and-cut scheme.

2.0 On Bilevel Minimum and Bottleneck Spanning Tree Problems

The contents of this chapter are mostly based on the published journal paper [99][1].

2.1 Motivation

In this chapter, we consider a class of bilevel spanning tree problems. Given a connected undirected edge-weighted graph $G = (V, E)$ with vertex set V and edge set E, the minimum spanning tree (MST) is defined as a connected subgraph that spans all vertices, does not contain any cycle and has the minimum possible total edge weight [3]. The problem of finding an MST, referred to as the MST problem, is one of the classical and well-known network optimization problems, which has numerous applications arising in a variety of important domains, such as network design [58, 73, 81], data storage [55], clustering [110, 120, 125], network reliability analysis [15, 44], etc.

The MST problem often arises in the design of physical systems (e.g., in communication, or transportation contexts), which are represented by a set of structural elements (i.e., vertices) that need to be fully connected in the simplest possible configuration with no redundancy (i.e., with no cycles); see, e.g., the discussions in [3, 4, 116] and the references therein. In the standard MST problem, a single decision-maker fully controls the edge selection process and optimizes his/her own objective function, i.e., the total sum of edge weights (costs) in the resulting tree. However, in many practical settings the decision-making process is decentralized and performed in a hierarchical manner. Consider the following motivating example. Assume that we are given a network design (e.g., communication, transportation) problem, where the overall goal is to connect n facilities by building $n - 1$ pairwise connections (e.g., communication or transportation links). Thus, the resulting network corresponds to a spanning tree. The upper-level decision-maker, e.g., the central government,

[1] Reprinted by permission from John Wiley & Son, Inc.: Wiley Periodicals, *Networks*: On bilevel minimum and bottleneck spanning tree problem. Shi, X., Zeng, B., & Prokopyev, O. A. (2019), 74(3), 251-273. ©2019

first constructs some of the required connections, for example, to ensure pairwise connectivity between some of the most important (from the perspective of the central government) facilities. Then the lower-level decision-maker, e.g., a local government, completes the construction by connecting the remaining facilities. The construction of each link is undertaken by one of these decision-makers and thus, their costs (or values) for each pairwise connection do not necessarily coincide. Furthermore, the objectives of the decision-makers involved in this hierarchical process can also be different. Therefore, the underlying optimization problem is, in fact, bilevel. Clearly, this example allows for a simple generalization if there are multiple decision-makers at the lower level.

This type of hierarchical decision-making processes is naturally captured by bilevel optimization models [27] and leads to a class of bilevel spanning tree (BST) problems, which can be viewed as a bilevel generalization of the MST problem. The BST problem involves two independent decision-makers who jointly construct a spanning tree in a graph through a two-level hierarchical decision-making process. The upper-level decision-maker (called the leader) acts first and selects a subset of edges that do not contain a cycle, from the edges of the graph under her control. In the second level, i.e., the lower level, the other decision-maker (called the follower) selects the remaining edges that are necessary to complete the spanning tree construction. Note that for convenience, we use "she/her" and "he/his" whenever we refer to the leader and the follower, respectively.

Each of the decision-makers has his/her own objective function, which is a function of their individual edge weights in the constructed spanning tree. Therefore, to optimize her objective function value, the leader needs to make decisions by taking into account the follower's actions (i.e., optimal solutions of the lower-level problem), which is referred to as the follower's reaction set. We refer readers to [27] for a survey on bilevel optimization that also overviews some basic concepts and algorithms in the area.

Formally, given a connected undirected graph $G = (V, E)$, let edge sets E_ℓ and E_f be subsets of E that are controlled by the leader and the follower, respectively. Denote by $\mathcal{T}(G)$ the set of all spanning trees in G. For a leader's decision $L \subseteq E_\ell$, the follower completes the construction of a spanning tree by selecting a set of edges $F \subseteq E_f$ such that $L \cup F \in \mathcal{T}(G)$. Let $g_\ell(L, F)$ and $g_f(L, F)$ be the objective functions of the leader and the follower, respectively, for $L \cup F \in \mathcal{T}(G)$. Then the bilevel spanning tree (BST) problem is formally given by:

$$\begin{aligned} [\text{BST}] \quad z^* = \text{“}\min_{L \subseteq E_\ell}\text{”} \; & g_\ell(L, F) \\ \text{s.t.} \quad & F \in \mathcal{R}(L) = \arg\min_{\tilde{F}} \{g_f(L, \tilde{F}) : \; \tilde{F} \in \mathcal{P}(L)\}, \end{aligned} \tag{2.1}$$

where $\mathcal{P}(L) = \{\tilde{F} \subseteq E_f : \; L \cup \tilde{F} \in \mathcal{T}(G)\}$ denotes the follower's feasible region given the leader's decision L. Note that $\mathcal{R}(L)$ is the collection of optimal solutions of the lower-level problem that form the follower's reaction set. We point out that there may exist multiple optimal solutions in the follower's reaction set, i.e., $|\mathcal{R}(L)| > 1$, which result in different objective function values for the leader. Then the follower needs to have a strategy to determine his action. In (2.1), similar to other related studies [31], we use the quotation marks to reflect this possibility of different approaches by the follower. Two major strategies, namely, the optimistic and pessimistic rules, are typically considered in the bilevel optimization literature [27].

Under the optimistic rule, the follower who is collaborative, selects the most favorable solution for the leader from his reaction set $\mathcal{R}(L)$. The optimistic reaction set, $\mathcal{R}^o(L)$, is then given by:

$$\mathcal{R}^o(L) = \{F \in \mathcal{R}(L) : \; g_\ell(L, F) \leq g_\ell(L, F') \; \forall F' \in \mathcal{R}(L)\}.$$

In contrast, the pessimistic rule specifies that the follower is adversarial and chooses the least favorable solution for the leader from his reaction set $\mathcal{R}(L)$. The pessimistic reaction set, $\mathcal{R}^p(L)$, is given by:

$$\mathcal{R}^p(L) = \{F \in \mathcal{R}(L) : \; g_\ell(L, F) \geq g_\ell(L, F') \; \forall F' \in \mathcal{R}(L)\}.$$

In this chapter, we focus on the two most popular objective functions that arise in different classes of the spanning tree optimization problems [3], namely, the sum- and bottleneck-type functions. For each edge $(i, j) \in E$, we denote the edge weights of the leader and the follower, by c_{ij} and d_{ij}, respectively. Then the objective function of the leader g_ℓ is either

$$c_S(L, F) = \sum_{(i,j) \in L \cup F} c_{ij} \qquad \text{or} \qquad c_B(L, F) = \max_{(i,j) \in L \cup F} c_{ij}, \tag{2.2}$$

where subscripts "S" and "B" are used to represent the "sum" and "bottleneck" function types, respectively. Similarly, the follower's objective function g_f is given by either

$$d_S(L, F) = \sum_{(i,j) \in L \cup F} d_{ij} \qquad \text{or} \qquad d_B(L, F) = \max_{(i,j) \in L \cup F} d_{ij}. \tag{2.3}$$

For convenience, we refer to different versions of the BST problem as $\mathrm{BST}_{(X,Y)}$, where $X \in \{S, B\}$ and $Y \in \{S, B\}$ denote the leader's and the follower's objective function types, respectively. If $g_\ell(L, F) = -g_f(L, F)$, then the BST problem reduces to the min-max spanning tree (MMST) problem, referred to as the MMST_X problem, where $X \in \{S, B\}$.

In general, the idea of extending classical combinatorial optimization problems into the bilevel settings has attracted substantial research attention in past decades. Some recent examples include the bilevel knapsack problem [12, 21] and the bilevel assignment problem [13, 42]. However, to the best of our knowledge, there are only few related studies that consider bilevel extensions of the spanning tree problems. In particular, Frederickson and Solis-Oba [38] present two versions of the minimum spanning tree interdiction problem. In the first version, referred to as the discrete one, the leader seeks to increase the MST weight by removing a subset of edges subject to a budgetary constraint. This problem turns out to be NP-hard. The other, continuous version, where instead of removing edges the leader can increase the weights of some edges, is polynomially solvable. Gassner [42] extends the setting of [38] to allow the leader and the follower have their own individual objective functions. Furthermore, the leader can either decrease or increase the edge weights, and her objective function consists of the weight modification costs and the function of the spanning tree completed by the follower. It is shown that the problem is NP-hard when both the upper-level and lower-level objective functions are of the sum-type; if the follower has the bottleneck-type objective function, then the problem is polynomially solvable. Cardinal et al. [24] introduce the Stackelberg pricing minimum spanning tree problem, in which the leader maximizes her revenue by setting prices for a set of edges, while the follower constructs the minimum spanning tree in the obtained graph. The problem is shown to be APX-hard but allows for an $O(\log n)$-approximation algorithm.

In view of our discussion above, the contribution of this chapter can be summarized as follows. We explore both the optimistic and pessimistic versions of the considered BST problems, where the leader's and the follower's objectives are either the sum- or bottleneck-type functions. More specifically, we develop polynomial-time algorithms for the MMST problem (both MMST_B and MMST_S) and almost all versions of the BST problems (namely, $\text{BST}_{(B,B)}$, $\text{BST}_{(B,S)}$ and $\text{BST}_{(S,B)}$), except the $\text{BST}_{(S,S)}$ case. The theoretical computational complexity of $\text{BST}_{(S,S)}$ remains an open question. However, for $\text{BST}_{(S,S)}$ we provide a single-level linear mixed-integer programming (MIP) formulation and perform a set of computational experiments with a standard MIP solver to explore its effectiveness. Admittedly, the considered bilevel setting can be viewed as somewhat stylized. However, given the importance and wide applicability of the MST problem and its variations, this study contributes to the growing body of literature on bilevel network and bilevel combinatorial optimization problems.

The remainder of the chapter is organized as follows. Section 2.2 introduces some necessary notations and describe basic structural properties of the BST problem that are required for our further derivations. In Section 2.3, we develop a polynomial-time algorithm for solving the MMST problem. Next, we discuss solution approaches for the $\text{BST}_{(X,B)}$ and $\text{BST}_{(X,S)}$ problems in Sections 2.4 and 2.5, respectively. Finally, our concluding remarks are presented in Section 2.6.

2.2 Technical Preliminaries

Given an edge subset $T \subseteq E$, let $G[T] = (\widetilde{V}, T)$ be the subgraph of G induced by T, where $\widetilde{V} \subseteq V$ is the set of vertices that are endpoints of edges in T, i.e., $\widetilde{V} = \{i \in V : \exists\, j \in V, \text{ such that } (i,j) \in T\}$. Note that if $G[T]$ is a spanning tree, then $\widetilde{V} = V$.

Throughout this chapter, we make the following assumption.

Assumption. *The follower controls all edges in graph G, that is, $E_f = E$.*

This assumption ensures that for any leader's decision L, the follower's reaction problem is always feasible as long as $G[L]$ is acyclic, which is a technical standard in the related bilevel optimization literature [27].

Definition 2.1. *For the BST problem, we say that (L, F) is a bilevel feasible solution with respect to the optimistic (pessimistic) rule, if $L \subseteq E_\ell$, and $F \in \mathcal{R}^o(L)$ $(F \in \mathcal{R}^p(L), respectively)$. Let $z(L) = g_\ell(L, F)$, where (L, F) is a bilevel feasible solution. If $G[L]$ contains cycles, we set $z(L) = +\infty$.*

Next, we present two technical results used further in the chapter, which also illustrate that the BST problem is well-defined.

Lemma 2.1. *If $L \cup F = L' \cup F'$, then $g_\ell(L, F) = g_\ell(L', F')$ and $g_f(L, F) = g_f(L', F')$.*

Proof. If follows directly from the definition of the objective functions in (2.2) and (2.3). ■

Lemma 2.1 implies that if two bilevel feasible solutions define the same spanning tree in the graph, then these two solutions also produce the same objective function value in both levels. We next show that there may exist multiple distinct bilevel feasible solutions that result in the same objective function value.

Lemma 2.2. *If (L, F) is a bilevel feasible solution of the BST problem under the optimistic (pessimistic) rule, then $(L \cup L', F \setminus L')$ is also a bilevel feasible solution under the optimistic (pessimistic, respectively) rule for any subset $L' \subseteq E_\ell \cap F$.*

Proof. If $L' = \emptyset$, then the statement is trivial. Otherwise, it is sufficient to show that $F \setminus L'$ is contained in the follower's reaction set $\mathcal{R}^o(L \cup L')$ (or $\mathcal{R}^p(L \cup L')$) under the optimistic (pessimistic, respectively) rule. Firstly, since $L \cup L' \subseteq E_\ell$ and $(L \cup L') \cup (F \setminus L') = L \cup F \in \mathcal{T}(G)$, we have $F \setminus L' \in \mathcal{P}(L \cup L')$.

Next, assume the optimistic problem. Suppose that $F \setminus L' \notin \mathcal{R}^o(L \cup L')$. Then there must exist a better follower's response that is a set $F' \in \mathcal{R}^o(L \cup L')$, such that either

$$g_f(L \cup L', F \setminus L') > g_f(L \cup L', F')$$

or

$$g_f(L \cup L', F \setminus L') = g_f(L \cup L', F') \quad \text{and} \quad g_\ell(L \cup L', F \setminus L') > g_\ell(L \cup L', F').$$

The latter holds as we consider the optimistic version. Then note from Lemma 2.1 that $g_f(L \cup L', F \setminus L') = g_f(L, F)$ and $g_\ell(L \cup L', F \setminus L') = g_\ell(L, F)$. Thus, we have either

$$g_f(L, F) > g_f(L \cup L', F'), \quad \text{or} \quad g_f(L, F) = g_f(L \cup L', F') \text{ and } g_\ell(L, F) > g_\ell(L \cup L', F').$$

On the other hand, recall that $L' \cup F' \in \mathcal{P}(L)$ and $F \in \mathcal{R}^o(L)$. Then by the definition of $\mathcal{R}^o(L)$ we have either

$$g_f(L, F) < g_f(L \cup L', F'), \quad \text{or} \quad g_f(L, F) = g_f(L \cup L', F') \text{ and } g_\ell(L, F) \le g_\ell(L \cup L', F').$$

Hence, we have a contradiction. Therefore, $(L \cup L', F \setminus L')$ is a bilevel feasible solution of the BST problem under the optimistic rule. The pessimistic version can be proved similarly. ■

2.3 The MMST Problem

In this section, we consider a version of the BST problem in which the objective functions of two decision-makers are opposite, i.e., $g_\ell(L, F) = -g_f(L, F)$. Simply speaking, the follower is adversarial to the leader. This problem is formally given by:

$$\text{[MMST]} \quad \min_{L \subseteq E_\ell} \max_{F \subseteq E_f} g_\ell(L, F)$$
$$\text{s.t.} \quad L \cup F \in \mathcal{T}(G).$$

The objective function $g_\ell(L, F)$ is either sum- or bottleneck-type functions as in (2.2), respectively. We note that for the min-max version, the optimistic and pessimistic cases coincide, that is $\mathcal{R}(L) = \mathcal{R}^o(L) = \mathcal{R}^p(L) = \{F \in \mathcal{P}(L) : \ g_\ell(L, F) \ge g_\ell(L, F') \ \forall F' \in \mathcal{P}(L)\}$.

Lemma 2.3. *Assume $E = \{(i_1, j_1), \ldots, (i_m, j_m)\}$, where $m = |E|$. If there exists an ordering of edges, such that $c_{i_1 j_1} < c_{i_2 j_2} < \cdots < c_{i_m j_m}$ and $d_{i_1 j_1} > d_{i_2 j_2} > \cdots > d_{i_m j_m}$, then the problem $BST_{(S,S)}$ reduces to the $MMST_S$ problem with weight vector c.*

Proof. Denote by problem $P1$, the $\text{BST}_{(S,S)}$ problem with the leader's and follower's edge weights given by c and d, respectively. Define $\widehat{d}$ as $\widehat{d}_{ij} = -c_{ij}$ for all $(i, j) \in E$. Then denote by problem $P2$, the $\text{BST}_{(S,S)}$ problem with the leader's and follower's edge weights given by c and $\widehat{d}$, respectively. Clearly, by construction $P2$ reduces to the MMST_S problem with weight vector c. Therefore, to establish the required result it suffices to show that for any leader's decision L, the followers in $P1$ and $P2$ have the same rational reaction set.

By the definition of $\widehat{d}$, we have d and $\widehat{d}$ have same ordering of edges, that is $\widehat{d}_{i_1j_1} > \widehat{d}_{i_2j_2} > \cdots > \widehat{d}_{i_mj_m}$. Therefore, an MST with respect to d is also an MST with respect to $\widehat{d}$; see, e.g., [3]. Furthermore, by construction $P1$ and $P2$ have same leader's objective functions. Consequently, for both the optimistic and pessimistic cases in $P1$, any follower's feasible solution in $P2$ is contained in the follower's reaction set in $P2$ if and only if it is in the follower's reaction set in $P1$. Then the result follows. ■

Let $\mathcal{R}_X(L)$ be the follower's reaction set corresponding to the leader's decision L of the MMST$_X$ problem, where $X \in \{S, B\}$.

Lemma 2.4. *For any leader's decision $L \subseteq E_\ell$, we have $\mathcal{R}_S(L) \subseteq \mathcal{R}_B(L)$.*

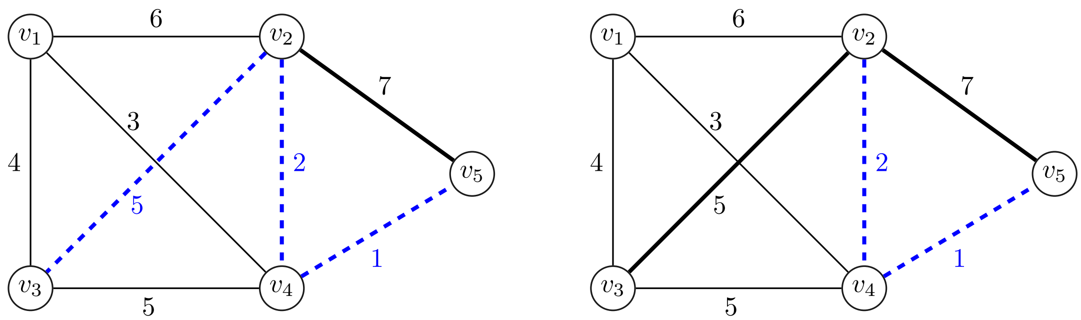
Proof. It is sufficient to show that if $F \in \mathcal{R}_S(L)$, then $F \in \mathcal{R}_B(L)$. Let $F' \in \mathcal{R}_B(L)$ and $(\bar{u}, \bar{v}) \in \arg\max_{(i,j)\in L\cup F'} \ c_{ij}$. If $(\bar{u}, \bar{v}) \in L$, then $\max_{(i,j)\in L\cup F} c_{ij} \geq \max_{(i,j)\in L} c_{ij} = c_{\bar{u}\bar{v}}$. Hence, $F \in \mathcal{R}_B(L)$ by the definition of $\mathcal{R}(L)$.

If $(\bar{u}, \bar{v}) \notin L$, we claim that there exists an edge (u, v) in F, such that $c_{uv} = c_{\bar{u}\bar{v}}$. To establish the latter, suppose that for any edge $(i, j) \in F$, $c_{ij} < c_{\bar{u}\bar{v}}$. Then we add edge $(\bar{u}, \bar{v})$ into $F \cup L \in \mathcal{T}(G)$. Note that $F \cup L \cup \{(\bar{u}, \bar{v})\}$ contains a cycle, which we denote by $\mathcal{C}$. Furthermore, recall that $(\bar{u}, \bar{v}) \in F'$. Thus, $L \cup \{(\bar{u}, \bar{v})\}$ is a forest and does not contain a cycle, which implies that $\mathcal{C}$ contains at least one edge, say (u', v'), from F. Then $L \cup (F \setminus (u', v')) \cup (\bar{u}, \bar{v}) \in \mathcal{T}(G)$, i.e., $(F \setminus (u', v')) \cup (\bar{u}, \bar{v}) \in \mathcal{P}(L)$. It follows that $\sum_{(i,j)\in L\cup F} c_{ij} < \sum_{(i,j)\in L\cup(F\setminus(u',v'))\cup(\bar{u},\bar{v})} c_{ij}$ as $c_{\bar{u}\bar{v}} > c_{u'v'}$ and thus, $F \notin \mathcal{R}_S(L)$, which is a contradiction.

Consequently, the aforementioned claim holds and we have $\max_{(i,j)\in L\cup F} c_{ij} \geq c_{\bar{u}\bar{v}}$, which implies that $F \in \mathcal{R}_B(L)$. This observation completes the proof. ■

The above result is not surprising given that a minimum spanning tree is also a bottleneck spanning tree [3]. Next, we establish several properties of the leader's optimal decisions that can be further exploited to show that MMST can be decomposed into two simpler problems.

Definition 2.2. *We say that edge set $L \subseteq E_\ell$ is maximal if $G[L]$ is acyclic and either $L = E_\ell$, or $G[L \cup \{e\}]$ contains a cycle for any edge $e \in E_\ell \setminus L$. Define set $\mathcal{L} = \{L \subseteq E_\ell :$ L is maximal$\}$ that consists of all maximal edge sets of E_ℓ.*



(a) $L_1 = \{(v_2, v_3), (v_2, v_4), (v_4, v_5)\}$ is an optimal solution of the leader that is a maximal edge set.

(b) $\{(v_2, v_4), (v_4, v_5)\}$ is an alternative leader's optimal solution that is not a maximal edge set.

Figure 1: An illustrative example for the MMST_S problem. Bold and bold dashed edges (the latter in blue) are controlled by the leader, while the bold dashed edges (in blue) correspond to a leader's optimal solution.

Figure 1 provides an illustrative example of the MMST_S problem, where the leader's edge set $E_\ell = \{(v_2, v_3), (v_2, v_4), (v_2, v_5), (v_4, v_5)\}$; see the edges depicted in bold. According to Definition 2.2, the maximal edge sets of E_ℓ are $L_1 = \{(v_2, v_3), (v_2, v_4), (v_4, v_5)\}$, $L_2 = \{(v_2, v_3), (v_2, v_4), (v_2, v_5)\}$ and $L_3 = \{(v_2, v_3), (v_2, v_5), (v_4, v_5)\}$, i.e., $\mathcal{L} = \{L_1, L_2, L_3\}$. Set L_1 is depicted in Figure 1(a).

Furthermore, we can verify that L_1 is also an optimal leader's solution. Next, we formally show that there always exists an optimal leader's solution that is a maximal edge set. Note, however, that there may exist alternative optimal leader's solutions that do not form maximal edge sets; see Figure 1(b) for an illustrative example.

Proposition 2.1. *For the MMST problem, let (L, F) and (L', F') be two bilevel feasible solutions. Then the following statements hold:*

(i) *If $L' = L \cup \{e\}$, where $e \in E_\ell \setminus L$, then $z(L') \leq z(L)$.*

(ii) *If $L, L' \subseteq \mathcal{L}$ and $L' = (L \setminus \{e\}) \cup \{e'\}$, where $e \in L, e' \in E_\ell \setminus L$ and $c_e \geq c_{e'}$, then $z(L') \leq z(L)$.*

Proof. The MMST_S problem is discussed first. The proof for MMST_B can be shown in a similar manner by applying Lemma 2.4.

(i) Observe that $F' \cup \{e\} \in \mathcal{P}(L)$ as $L' \cup F' = L \cup (F' \cup \{e\}) \in \mathcal{T}(G)$. Also, since $F \in \mathcal{R}_S(L)$ and from Lemma 2.1, we have $z(L) = g_\ell(L, F) \geq g_\ell(L, F' \cup \{e\}) = g_\ell(L', F') = z(L')$.

(ii) Note that L and L' are maximal sets in E_ℓ and $e' \in E_\ell \setminus L$. Thus, $e \in E_\ell \setminus L'$. It follows that $L' \cup \{e\}$ and $L \cup \{e'\}$ contain cycles. In addition, we observe that $L \cup \{e'\} = L' \cup \{e\}$, which implies that e' is contained in the cycle of $G[L' \cup \{e\}]$. Hence, $F' \cup (L' \setminus \{e'\}) \cup \{e\}$ is a spanning tree in G, i.e., $F' \in \mathcal{P}(L)$. Similarly, $F \in \mathcal{P}(L')$.
Recall that $F \in \mathcal{R}(L)$ and $F' \in \mathcal{R}(L')$. Then by definition $\sum_{(i,j)\in L\cup F} c_{ij} \geq \sum_{(i,j)\in L\cup F'} c_{ij}$ and $\sum_{(i,j)\in L'\cup F} c_{ij} \leq \sum_{(i,j)\in L'\cup F'} c_{ij}$, which yields that $\sum_{(i,j)\in F} c_{ij} = \sum_{(i,j)\in F'} c_{ij}$.
On the other hand, $\sum_{(i,j)\in L} c_{ij} \geq \sum_{(i,j)\in L'} c_{ij}$ due to $c_e \geq c_{e'}$. Therefore, $z(L) = g_\ell(L, F) = \sum_{(i,j)\in L} c_{ij} + \sum_{(i,j)\in F} c_{ij} \geq \sum_{(i,j)\in L'} c_{ij} + \sum_{(i,j)\in F'} c_{ij} = g_\ell(L', F') = z(L')$.

■

Based on the first statement in Proposition 2.1, for the leader, it is always favorable to select more edges in E_ℓ, and there exists an optimal leader's decision in $\mathcal{L}$. Furthermore, the second statement illustrates that an optimal decision of the leader can be obtained by minimizing the total edge weights of the maximal edge set of E_ℓ for both MMST_S and MMST_B. Thus, we can consider the following single-level problem:

$$\min_{L\subseteq E_\ell} \Big\{ \sum_{(i,j)\in L} c_{ij} : \ L \in \mathcal{L} \Big\}. \tag{2.4}$$

The problem (2.4) seeks for an edge set in $\mathcal{L}$ whose total edge weights are as small as possible from among all other maximal sets in $\mathcal{L}$. Similar to the minimum spanning tree problem, Kruskal's algorithm [3] can be applied for solving the reformulation (2.4). After the optimal upper-level's decisions are fixed, the optimal value of the MMST problem can be obtained by solving the corresponding lower-level problem. The formal pseudo-code of this approach is outlined in Algorithm 1.

Proposition 2.2. *Algorithm 1 runs in* $O(|E_\ell|^2 + |E| \cdot \log |V|)$ *time.*

Algorithm 1 Algorithm for the MMST problem

Input $G = (V, E)$, E_ℓ, c
1: $L \leftarrow \emptyset$
2: $\mathcal{E}_\ell \leftarrow E_\ell$ and sort $\mathcal{E}_\ell$ in the nondecreasing order with respect to c
3: **while** $\mathcal{E}_\ell \neq \emptyset$ **do**
4: $\quad e \in \arg\min_{(i,j)\in\mathcal{E}_\ell}\{c_{ij}\}$
5: $\quad$ **if** $L \cup \{e\}$ is acyclic **then**
6: $\quad\quad L \leftarrow L \cup \{e\}$
7: $\quad$ **end if**
8: $\quad \mathcal{E}_\ell \leftarrow \mathcal{E}_\ell \setminus \{e\}$
9: **end while**
10: $z \leftarrow \max_{F\subseteq E\setminus\mathcal{E}_\ell}\{g_\ell(L, F) : L \cup F \in \mathcal{T}(G)\}$
Return z and L

Proof. First, sorting E_ℓ in the nondecreasing order requires $O(|E_\ell| \cdot \log |E_\ell|)$ time. In lines 3-9 of each iteration, the operation to verify whether there exists a cycle needs $O(|E_\ell|)$ time. Therefore, the total time required to find the leader's optimal decision is $O(|E_\ell|^2)$ as there are at most $|E_\ell|$ iterations. Recall that the single-level spanning tree problem over graph $G = (V, E)$ can be solved within $O(|E| \cdot \log |V|)$ time [3]. Thus, in line 10 we obtain an optimal solution for MMST by solving a single-level maximum spanning tree problem over graph $(V, E \setminus \mathcal{E}_\ell)$, which requires $O(|E| \cdot \log |V|)$ time. ■

2.4 The BST$_{(X,B)}$ Problem

In this section, we first consider structural properties of BST$_{(X,B)}$ and derive its equivalent reformulation in Section 2.4.1. Next, polynomial-time algorithms are proposed in Section 2.4.2 and 2.4.3 for the optimistic and pessimistic cases of the problem, respectively.

2.4.1 Structural Properties

Given the leader's decision L, the follower solves the single-level bottleneck spanning tree problem. If $L \neq \emptyset$, then define the maximum edge weight of L under the weight d as:

$$d_L = \max_{(i,j)\in L} d_{ij}.$$

Define subgraph $G^L = (V, E^L)$, where $E^L = \{(i,j) \in E : \ d_{ij} \leq d_L\}$. If the leader does not choose any edge, i.e., $L = \emptyset$, then the optimal objective function value of the lower-level problem is the maximum edge weight of the bottleneck spanning tree in G, given by:

$$\bar{d} = \min_{T \subseteq E} \{ \max_{(i,j) \in T} d_{ij} : \ T \in \mathcal{T}(G)\}.$$

Define subgraph $\bar{G} = (V, \bar{E})$, where $\bar{E} = \{(i,j) \in E : \ d_{ij} \leq \bar{d}\}$. We note that $\bar{G}$ is a connected graph, which can be computed in polynomial time for any G.

For a feasible leader's decision L, if $d_L \geq \bar{d}$, then $\bar{G} \subseteq G^L$, which implies that G^L is connected. Then there exists at least one spanning tree containing L in G^L. Hence, for any follower's reaction $F \in \mathcal{R}(L)$, the objective function value of the lower-level problem satisfies that $d_L = g_f(L, \emptyset) \leq g_f(L, F) \leq d_L$. Thus, the optimal objective function value of the lower-level problem is d_L, and the reaction set can be represented as:

$$\mathcal{R}(L) = \{F \subseteq E^L : \ L \cup F \in \mathcal{T}(G)\}.$$

Similarly, when $d_L \leq \bar{d}$, due to the connectivity of $\bar{G}$, we can find a feasible solution of the lower-level problem, say $\hat{F}$, such that $\hat{F} \subseteq \bar{E}$ and $L \cup \hat{F} \in \mathcal{T}(G)$. Since the maximum edge weight of any spanning tree in G is no less than $\bar{d}$, then for any follower's solution $F \in \mathcal{P}(L)$, we have $g_f(L, \hat{F}) \leq \bar{d} \leq g_f(L, F)$. It follows that $\hat{F} \in \mathcal{R}(L)$ and the optimal value of the lower-level problem is $\bar{d}$. Then the follower's reaction set is given by:

$$\mathcal{R}(L) = \{F \subseteq \bar{E} : \ L \cup F \in \mathcal{T}(G)\}.$$

Therefore, the $\text{BST}_{(X,B)}$ problem can be reformulated as:

$$z^* = \min \{z_1^*, z_2^*\}.$$

where

$$z_1^* = \text{``}\min_{L \subseteq E_\ell}\text{''} \{g_\ell(L, F) : \ d_L \leq \bar{d}, F \subseteq \bar{E}, L \cup F \in \mathcal{T}(G)\}, \tag{2.5}$$

$$z_2^* = \text{``}\min_{L \subseteq E_\ell}\text{''} \{g_\ell(L, F) : \ d_L \geq \bar{d}, F \subseteq E^L, L \cup F \in \mathcal{T}(G)\}, \tag{2.6}$$

where the quotation marks are used with the same meaning as in (2.1). We next explore the optimistic and pessimistic versions in detail.

2.4.2 Optimistic Case

Under the optimistic rule, (2.5) and (2.6) are single-level optimization problems. We first establish that $z_1^* = z(\emptyset)$.

Lemma 2.5. *For the $BST_{(X,B)}$ problem under the optimistic rule, let L be the leader's decision and let $G[L]$ be acyclic. If $L \neq \emptyset$ and $d_L \leq \bar{d}$, then $z(L) \geq z(\emptyset)$. Thus, $z_1^* = z(\emptyset)$.*

Proof. Let $F \in \mathcal{R}^o(L)$ and $\bar{F} \in \mathcal{R}^o(\emptyset)$. Since $d_L \leq \bar{d}$, then the optimal objective function value of the follower's problem corresponding to L is $\bar{d}$ because of the connectivity of $\bar{G}$. That is, $g_f(L,F) = g_f(\emptyset, \bar{F}) = \bar{d}$. Also, $L \cup F \in \mathcal{T}(G) = \mathcal{P}(\emptyset)$. Thus, $L \cup F \in \mathcal{R}(\emptyset)$. Based on the definition of $\mathcal{R}^o(\emptyset)$, we have $z(\emptyset) = g_\ell(\emptyset, \bar{F}) \leq g_\ell(\emptyset, L \cup F) = z(L)$. ■

Let $\mathcal{E}_\ell = \{(i,j) \in E_\ell : d_{ij} \geq \bar{d}\}$. For a leader's decision L, if edge e of the maximum edge weight over d in L is known and $e \in \mathcal{E}_\ell$, then $L \subseteq E_\ell^e := \{(i,j) \in E_\ell :\ d_{ij} \leq d_e\}$ and G^L becomes $G^e = (V, E^e)$, where $E^e := \{(i,j) \in E :\ d_{ij} \leq d_e\}$. Then problem (2.6) can be reformulated as follows:

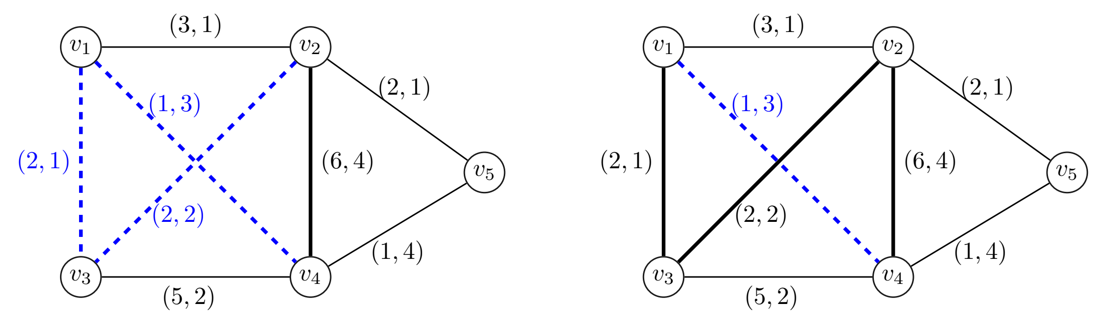
$$
\begin{aligned}
z_2^* &= \min_{L \subseteq E_\ell, F \subseteq E^L} \{g_\ell(L,F) :\ d_L \geq \bar{d}, L \cup F \in \mathcal{T}(G)\} \\
&= \min_{e \in \mathcal{E}_\ell} \min_{L \subseteq E_\ell^e, F \subseteq E^e} \{g_\ell(L,F) :\ e \in L, L \cup F \in \mathcal{T}(G)\} \\
&= \min_{e \in \mathcal{E}_\ell} \min_{T \subseteq E^e} \{g_\ell(\emptyset, T) :\ e \in T, T \in \mathcal{T}(G)\},
\end{aligned} \tag{2.7}
$$

where the first equality holds under the optimistic rule and the last equality follows from Lemma 2.1 as $E_\ell^e \subseteq E^e$ and $T = L \cup F$.

Next, assume the optimal solution of (2.7) is (e^*, T^*) with $d_{e^*} \geq \bar{d}$. Let $L^* = \{e^*\}$ and $F^* = T^* \setminus \{e^*\}$. Then we can show that (L^*, F^*) is an optimal solution of $\mathrm{BST}_{(X,B)}$ as follows.

Proposition 2.3. *For the $BST_{(X,B)}$ problem under the optimistic rule, there exists an optimal solution (L^*, F^*), such that either $L^* = \emptyset$ or $L^* = \{(i,j)\}$ such that $d_{ij} \geq \bar{d}$.*

Proof. Assume (L,F) is an optimal solution of the bilevel problem. If $d_L \leq \bar{d}$, then let $L^* = \emptyset$. From Lemma 2.5, we conclude that L^* is also an optimal decision of the leader.



(a) $\{(v_1, v_3), (v_1, v_4), (v_2, v_3)\}$ is an optimal solution of the leader in both optimistic and pessimistic cases.

(b) $\{(v_1, v_4)\}$ is an alternative leader's optimal solution in the optimistic case that contains only one edge.

Figure 2: An illustrative example for $\mathrm{BST}_{(S,B)}$ and Proposition 2.3. For each edge (i, j), pair (c_{ij}, d_{ij}) depicted above the edge denotes the leader's and the follower's edge weights, respectively. Bold and bold dashed edges (the latter in blue) are controlled by the leader, while the bold dashed edges (in blue) correspond to a leader's optimal solution.

On the other hand, if $d_L \geq \bar{d}$, suppose $(i_1, j_1) \in L$ and $d_L = d_{i_1 j_1} \geq d_{uv}$ for every $(u, v) \in L$. Let $L^* = \{(i_1, j_1)\}$ and $F^* \in \mathcal{R}^o(L^*)$. Since $d_{i_1 j_1} = d_L \geq \bar{d}$, then $g_f(L, F) = g_f(L^*, F^*) = d_{i_1 j_1}$. Observe that $F \cup (L \setminus L^*) \in \mathcal{P}(L^*)$ and $g_f(L^*, F \cup (L \setminus L^*)) = g_f(L, F) = d_{i_1 j_1}$. Hence, we have $z(L^*) = g_\ell(L^*, F^*) \leq g_\ell(L^*, F \cup (L \setminus L^*)) = g_\ell(L, F) = z(L)$ as $F^* \in \mathcal{R}^o(L^*)$. Therefore, L^* is an optimal decision of the leader, and the result follows. ■

Figure 2 provides an illustrative example for Proposition 2.3 with an instance of the $\mathrm{BST}_{(S,B)}$ problem, where $E_\ell = \{(v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4)\}$. Note that $\bar{d} = 2$ and the subset of edges $\{(v_1, v_3), (v_1, v_4), (v_2, v_3)\}$ depicted in Figure 2(a) is an optimal leader's solution for both the optimistic and pessimistic versions of $\mathrm{BST}_{(S,B)}$. The corresponding lower-level reaction is $\{(v_2, v_5)\}$. Furthermore, according to Proposition 2.3 we observe that $\{(v_1, v_4)\}$ is also an optimal leader's decision, with the corresponding follower's decision given by $\{(v_1, v_3), (v_2, v_3), (v_2, v_5)\}$.

Based on reformulation (2.7) and Proposition 2.3, we conclude that to solve the $\mathrm{BST}_{(X,B)}$ problem it is sufficient to consider edges in $\mathcal{E}_\ell = \{(i, j) \in E_\ell : d_{ij} \geq \bar{d}\}$ as possible leader's decisions, L^*. Then the optimal solution is returned by comparing the obtained objective function values and $z(\emptyset)$. This approach results in a polynomial-time algorithm formalized with the pseudo-code given in Algorithm 2.

Algorithm 2 Algorithm for the $\mathrm{BST}_{(X,B)}$ problem under the optimistic rule

Input $G = (V, E), E_\ell, c, d$
1: $L \leftarrow \emptyset, L^* \leftarrow \emptyset$
2: $z^* \leftarrow g_\ell(\emptyset, F)$, where $F \in \mathcal{R}^o(\emptyset)$
3: $\bar{d} \leftarrow \max_{(i,j)\in F} d_{ij}$
4: $\mathcal{E}_\ell \leftarrow \{(i,j) \in E_\ell : d_{ij} \geq \bar{d}\}$
5: **while** $\mathcal{E}_\ell \neq \emptyset$ **do**
6: $\quad L \leftarrow \{e\}$, where edge e is chosen from $\mathcal{E}_\ell$
7: $\quad z(L) \leftarrow g_\ell(L, F)$, where $F \in \mathcal{R}^o(L)$
8: $\quad$ **if** $z(L) \leq z^*$ **then**
9: $\quad\quad z^* \leftarrow z(L)$
10: $\quad\quad L^* \leftarrow L$
11: $\quad$ **end if**
12: $\quad \mathcal{E}_\ell \leftarrow \mathcal{E}_\ell \setminus \{e\}$
13: **end while**
Return z^* and L^*

Proposition 2.4. *Algorithm 2 runs in $O(|E_\ell| \cdot |E| \cdot \log |V|)$ time.*

Proof. The algorithm requires at most $|E_\ell|$ iterations in lines 5-13. In every iteration, after fixing an edge in L, in line 7 we solve the follower's reaction problem that is a single-level spanning tree problem. The latter requires $O(|E| \cdot \log |V|)$ time and the result follows. ■

2.4.3 Pessimistic Case

For the pessimistic version, the reformulations (2.5) and (2.6) can be re-written as:

$$z_1^* = \min_{L \subseteq \{(i,j)\in E_\ell:\ d_{ij} \leq \bar{d}\}} \max_{F \subseteq \bar{E}} \{g_\ell(L, F) :\ L \cup F \in \mathcal{T}(G)\}, \tag{2.8}$$

$$z_2^* = \min_{L \subseteq E_\ell} \max_{F \subseteq E^L} \{g_\ell(L, F) :\ d_L \geq \bar{d}, L \cup F \in \mathcal{T}(G)\}, \tag{2.9}$$

which allows us to observe that problem (2.8) is, in fact, an instance of the MMST problem over graph $\bar{G} = (V, \bar{E})$, where the leader's edge set is limited to $\{(i,j) \in E_\ell \ :\ d_{ij} \leq \bar{d}\}$ instead of E_ℓ. Therefore, z_1^* can be computed using Algorithm 1 within $O(|E_\ell|^2 + |\bar{E}| \cdot \log |V|)$ time; see Proposition 2.2.

Regarding problem (2.9), recall first that for fixed edge $e \in \mathcal{E}_\ell = \{(i,j) \in E_\ell : \ d_{ij} \geq \bar{d}\}$, we have $E_\ell^e = \{(i,j) \in E_\ell : \ d_{ij} \leq d_e\}$, $E^e = \{(i,j) \in E : \ d_{ij} \leq d_e\}$ and $G^e = (V, E^e)$. Let e be the maximum weight edge in the leader's decision L. Then problem (2.9) can be reformulated as follows:

$$z_2^* = \min_{e \in \mathcal{E}_\ell} \left\{ \min_{L \subseteq E_\ell^e} \max_{F \subseteq E^e} \{ g_\ell(L,F) : \ e \in L, L \cup F \in \mathcal{T}(G) \} \right\}. \tag{2.10}$$

Next, we observe that for fixed $e \in \mathcal{E}_\ell$, reformulation (2.10) reduces to the MMST problem over graph $G^e = (V, E^e)$, where the leader controls E_ℓ^e and $e \in L$. Thus, we can simply enumerate over all possible choices of e and use Algorithm 1 from Section 2.3 as a sub-procedure.

Given the above discussion, the formal pseudo-code for solving the $\text{BST}_{(X,B)}$ problem under the pessimistic rule is given in Algorithm 3. An optimal solution is obtained by comparing the objective function values z_1^* and z_2^*. The former is computed in line 3, while the latter is evaluated iteratively in lines 6-14.

Algorithm 3 Algorithm for the $\text{BST}_{(X,B)}$ problem under the pessimistic rule

Input $G = (V, E), E_\ell, c, d$
1: $L \leftarrow \emptyset, L^* \leftarrow \emptyset$
2: $\bar{d} \leftarrow \min_{T \in \mathcal{T}(G)} \{\max_{(i,j) \in T} d_{ij}\}$
3: $L \in \arg\min_{L \subseteq E_\ell} \{\max_{F \subseteq E} g_\ell(L,F) : \ d_{ij} \leq \bar{d} \ \ \forall (i,j) \in L \cup F, L \cup F \in \mathcal{T}(G)\}$
// solved as MMST by Algorithm 1
4: $z^* \leftarrow z(L), L^* \leftarrow L$
5: $\mathcal{E}_\ell \leftarrow \{(i,j) \in E_\ell : \ d_{ij} \geq \bar{d}\}$
6: **while** $\mathcal{E}_\ell \neq \emptyset$ **do**
7: Choose an edge $e \in \mathcal{E}_\ell$
8: $L \in \arg\min_{L \in E_\ell} \{\max_{F \in E} g_\ell(L,F) : \ d_{ij} \leq d_e \ \forall (i,j) \in L \cup F, e \in L, L \cup F \in \mathcal{T}(G)\}$
// solved as MMST by Algorithm 1
9: **if** $z(L) \leq z^*$ **then**
10: $z^* \leftarrow z(L)$
11: $L^* \leftarrow L$
12: **end if**
13: $\mathcal{E}_\ell \leftarrow \mathcal{E}_\ell \setminus \{e\}$
14: **end while**
Return z^* and L^*

Proposition 2.5. *Algorithm 3 runs in* $O(|E_\ell|^3 + |E_\ell| \cdot |E| \cdot \log |V|)$ *time.*

Proof. To compute z_1^* in line 3, the algorithm solves an instance of the MMST problem in $O(|E_\ell|^2 + |\bar{E}| \cdot \log |V|)$ time; see Proposition 2.2. The algorithm terminates in at most $O(|E_\ell|)$ iterations from lines 6 to 14. In each iteration, an instance of the MMST problem is solved in $O(|E_\ell|^2 + |E| \cdot \log |V|)$ time in line 8; see Proposition 2.2. Then the result follows. ■

2.5 The $\text{BST}_{(X,S)}$ Problem

In Section 2.5.1, we derive structural properties of the reaction set for the lower-level problem in $\text{BST}_{(X,S)}$. The results are obtained for both the optimistic and pessimistic cases. Thus, in Sections 2.5.2 and 2.5.3 we focus on solution methods for the optimistic case, which can be adapted to the pessimistic case in a simple manner. Specifically, in Section 2.5.2 we develop a polynomial-time algorithm for the $\text{BST}_{(B,S)}$ problem. In Section 2.5.3, we first show that $\text{BST}_{(S,S)}$ can be formulated as a single-level mixed-integer program and then provide a preliminary computational study to explore its performance. The theoretical computational complexity of $\text{BST}_{(S,S)}$ remains an open question.

2.5.1 Structural Properties

For a given leader's decision, assume that the follower needs to choose either (i, j) or (u, v) in E_f and $d_{ij} = d_{uv}$. If $c_{ij} < c_{uv}$, then the follower selects (i, j) under the optimistic rule, as it is preferable for the leader. Otherwise, the follower selects (u, v) under the pessimistic rule. Based on the above observation, we then define the following edge ordering:

Definition 2.3. *For any pair of edges (i, j) and (u, v) in E, define the order of (i, j) and (u, v) under the optimistic rule:*

(i) *if either $d_{ij} < d_{uv}$, or $d_{ij} = d_{uv}$ and $c_{ij} \leq c_{uv}$, then $(i, j) \preceq (u, v)$;*

(ii) *if $(i, j) \preceq (u, v)$ and $(u, v) \preceq (i, j)$ then $(i, j) = (u, v)$;*

(iii) *if $(i, j) \preceq (u, v)$ and $(i, j) \neq (u, v)$, then $(i, j) \prec (u, v)$.*

Similarly, we define the order of (i, j) and (u, v) under the pessimistic rule as follows:

(i) *if either $d_{ij} < d_{uv}$, or $d_{ij} = d_{uv}$ and $c_{ij} \geq c_{uv}$, then $(i, j) \preceq (u, v)$;*

(ii) *if* $(i,j) \preceq (u,v)$ *and* $(u,v) \preceq (i,j)$ *then* $(i,j) = (u,v)$*;*

(iii) *if* $(i,j) \preceq (u,v)$ *and* $(i,j) \neq (u,v)$*, then* $(i,j) \prec (u,v)$*.*

The path- and cut-optimality conditions for the minimum spanning tree, see [3], can be directly adopted, see Lemma 2.6 below, to provide optimality conditions for the lower-level problem of $\text{BST}_{(S,S)}$ for a given leader's decision. The proof of Lemma 2.6 is omitted as it is similar to those provided in [3].

Definition 2.4. *Let* $G[T] = (V,T)$ *be a spanning tree in* G*. If we delete an edge* (i,j) *in* T*, then the resulting graph* $G[T \setminus (i,j)]$ *becomes disconnected with two disjoint connected components. Denote the sets of vertices in these two components as* $S_T(i,j)$ *and* $S'_T(i,j)$*, respectively.*

Lemma 2.6. *For the* $BST_{(S,S)}$ *problem, let* $G[T] = (V,T)$ *be a spanning tree of graph* G*, where* $T = L \cup F, L \subseteq E_\ell$ *and* $F \subseteq E_f$*. Then the following statements are equivalent:*

(i) $F \in \mathcal{R}^o(L)$ $(F \in \mathcal{R}^p(L))$*.*

(ii) (Cut optimality condition) *Let* $(i,j) \in F$*. Then* $(i,j) \preceq (u,v)$ *for any* $(u,v) \in E \setminus T$*, where* $u \in S_T(i,j), v \in S'_T(i,j)$ *and the operator* $\preceq$ *is defined under the optimistic (pessimistic, respectively) rule.*

(iii) (Path optimality condition) *Let* $(u,v) \in E \setminus T$*. Then* $(u,v) \succeq (i,j)$ *for any* $(i,j) \in \Delta \cap F$*, where* Δ *is the unique path in* $G[T]$ *connecting* u *and* v*, and the operator* $\succeq$ *is defined under the optimistic (pessimistic, respectively) rule.*

Denote by $\mathcal{R}^o_{(B,S)}(L)$ and $\mathcal{R}^o_{(S,S)}(L)$ the optimistic reactions sets of the $\text{BST}_{(B,S)}$ and the $\text{BST}_{(S,S)}$ problems, respectively. Similarly, denote by $\mathcal{R}^p_{(B,S)}(L)$ and $\mathcal{R}^p_{(S,S)}(L)$ the corresponding reaction sets for the pessimistic versions of the problems. Similar to Lemma 2.4 for MMTS, there exists a relationship between the reaction sets of $\text{BST}_{(B,S)}$ and $\text{BST}_{(S,S)}$.

Lemma 2.7. *Given the leader's decision* L*, we have* $\mathcal{R}^o_{(S,S)}(L) \subseteq \mathcal{R}^o_{(B,S)}(L)$*, and* $\mathcal{R}^p_{(S,S)}(L) \subseteq \mathcal{R}^p_{(B,S)}(L)$*.*

Proof. We provide the proof for the optimistic case. The proof of the pessimistic case is similar and omitted for brevity. Suppose $F \in \mathcal{R}^o_{(S,S)}(L)$, $F' \in \mathcal{R}^o_{(B,S)}(L)$, and let $T = L \cup F$. It is sufficient to show that $F \in \mathcal{R}^o_{(B,S)}(L)$, which is the case if $\max_{(i,j)\in F} c_{ij} \leq \max_{(i,j)\in F'} c_{ij}$.

We note first that $F, F' \in \mathcal{R}(L)$, which implies that $\sum_{(i,j)\in L\cup F} d_{ij} = \sum_{(i,j)\in L\cup F'} d_{ij}$. It follows that for any edge $(i,j) \in F \setminus F'$, there exists an edge $(u',v') \in F'$, such that $u' \in S_T(i,j)$, $v' \in S'_T(i,j)$ and $d_{u'v'} = d_{ij}$. Based on the optimality conditions of F in Lemma 2.6, we have $(u',v') \succeq (i,j)$. Hence, from Definition 2.3 we have $d_{ij} = d_{u'v'}$ and $c_{ij} \leq c_{u'v'}$.

Therefore, for each edge $(i,j) \in F\setminus F'$, there exists edge (u',v') in F', such that $d_{ij} = d_{u'v'}$ and $c_{ij} \leq c_{u'v'}$, which implies $\max_{(i,j)\in F} c_{ij} \leq \max_{(i,j)\in F'} c_{ij}$ and the result follows. ■

Next, we show that for any leader's decision, the edges selected by the follower are contained in a particular subset of E; see Corollary 2.1 of Theorem 2.1. First, we derive the following technical result.

Lemma 2.8. *Let $T = L \cup F$, where (L, F) is a bilevel feasible solution of the $BST_{(S,S)}$ problem. If $L^{'} = L \cup (i',j')$, where $(i',j') \in E_\ell \setminus L$ and $G[L']$ is acyclic, then:*

- (i) *if $(i',j') \in F$, then $(L', F \setminus (i',j'))$ is a bilevel feasible solution.*
- (ii) *if $(i',j') \notin F$, then $(L', F \setminus (\bar{u},\bar{v}))$ is a bilevel feasible solution, where $(\bar{u},\bar{v})$ is the edge of largest order in $\Delta' \cap F$ and Δ' is the unique path in $G[T]$ connecting i' and j', i.e., $(u,v) \in \Delta' \cap F$ and $(\bar{u},\bar{v}) \succeq (u,v)$ for any $(u,v) \in \Delta' \cap F$.*

The result holds for both the optimistic and pessimistic versions of the $BST_{(S,S)}$ problem.

Proof. We assume the optimistic case. The proof for the pessimistic version is similar.

- (i) Clearly, if $(i',j') \in F$, then the result is a special case of Lemma 2.2.
- (ii) If $(i',j') \notin F$, then let $F' = F \setminus (\bar{u},\bar{v})$, where $(\bar{u},\bar{v})$ is the edge of largest order in $\Delta' \cap F$ and Δ' is the unique path in $G[T]$ connecting i' and j'. Let $T' = L' \cup F'$. To prove that $(L', F \setminus (\bar{u},\bar{v}))$ is bilevel feasible, it is sufficient to show that F' satisfies the path optimality condition in Lemma 2.6, and we establish this result next.

  For edge $(\bar{u},\bar{v})$, denote the unique path in $G[T']$ connecting $\bar{u}$ and $\bar{v}$ as $\bar{\Delta}$. Clearly, $\bar{\Delta}$ consists of edges in the set $(i',j') \cup \Delta' \setminus (\bar{u},\bar{v})$. Since $(\bar{u},\bar{v})$ is the edge of largest order in $\Delta' \cap F$, then $(\bar{u},\bar{v}) \succeq (u,v)$ for all $(u,v) \in \bar{\Delta} \cap F'$, as by construction $\bar{\Delta} \cap F' = \Delta' \cap F \setminus (\bar{u},\bar{v})$. This means that F' satisfies the path optimality condition for edge $(\bar{u},\bar{v})$.

To establish the condition for any other edge $(u, v) \in E \setminus T'$, denote by Δ_1 and Δ_2 the unique paths in $G[T]$ and $G[T']$, respectively, that connect u and v. Recall that (L, F) is bilevel feasible. Thus, $(u, v) \succeq (i, j)$ for any $(i, j) \in \Delta_1 \cap F$ because of the path optimality condition in Lemma 2.6 for F. Then there are two possible cases:

(a) If $(\bar{u}, \bar{v}) \notin \Delta_1$, then $\Delta_1 \subseteq T'$ as $T \setminus (\bar{u}, \bar{v}) = T' \setminus (i', j')$. Since the path in $G[T']$ connecting $\bar{u}$ and $\bar{v}$ is unique, it follows that $\Delta_1 = \Delta_2$. Thus, $(u, v) \succeq (i, j)$ for any (i, j) in $\Delta_2 \cap F'$ due to $F' \subseteq F$.

(b) If $(\bar{u}, \bar{v}) \in \Delta_1$, then $(u, v) \succeq (\bar{u}, \bar{v})$ and $\Delta_1 \setminus (\bar{u}, \bar{v}) \subseteq T'$. Observe that $\Delta_1 \setminus (\bar{u}, \bar{v})$ contains two disconnected paths; assume that one path is between u and $\bar{u}$, and the other is between v and $\bar{v}$. Then in spanning tree $G[T']$, we can walk from u to $\bar{u}$, from $\bar{u}$ to $\bar{v}$ through $\bar{\Delta}$, and then from $\bar{v}$ to v. It follows that the unique path in $G[T']$ connecting u and v belongs to $\Delta_1 \cup \bar{\Delta}$, i.e., $\Delta_2 \subseteq \Delta_1 \cup \bar{\Delta}$. Recall that $(u, v) \succeq (i, j)$ for any $(i, j) \in \Delta_1 \cap F$, and $(\bar{u}, \bar{v}) \succeq (i, j)$ for any $(i, j) \in \bar{\Delta} \cap F'$. Thus, we have $(u, v) \succeq (i, j)$ for any $(i, j) \in (\Delta_1 \cup \bar{\Delta}) \cap F'$, which implies that F' satisfies the path optimality condition for (u, v).

Therefore, $F' \in \mathcal{R}^o_{(S,S)}(L)$ and the result follows. ∎

Theorem 2.1. *Let (L, F) be a bilevel feasible solution of $BST_{(S,S)}$ under the optimistic (pessimistic) rule. Assume that $L \subseteq L' \subseteq E_\ell$ and $G[L']$ is acyclic. Then there exists a reaction solution of the follower $F' \in \mathcal{R}^o_{(S,S)}(L')$ ($F' \in \mathcal{R}^p_{(S,S)}(L')$, respectively), such that $F' \subseteq F$.*

Proof. The theorem can be proved by induction using Lemma 2.8. ∎

Recall that when the leader does not select any edge, i.e., $L = \emptyset$, we use $\mathcal{R}^o(\emptyset)$ and $\mathcal{R}^p(\emptyset)$ to denote the follower's optimistic and pessimistic reaction sets, respectively. Then:

Corollary 2.1. *For the $BST_{(X,S)}$ problem under the optimistic (pessimistic) rule, for every $T \in \mathcal{R}^o_{(S,S)}(\emptyset)$ ($T \in \mathcal{R}^p_{(S,S)}(\emptyset)$, respectively), there exists an optimal solution (L^*, F^*) of $BST_{(X,S)}$ such that $F^* \subseteq T$, where $X \in \{S, B\}$.*

Proof. By setting $L = \emptyset$ in Theorem 2.1 the result follows directly for the $\mathrm{BST}_{(S,S)}$ problem. That is, for any tree $T \in \mathcal{R}^o_{(S,S)}(\emptyset)$, there exists $F^* \in \mathcal{R}^o_{(S,S)}(L^*)$, such that $F^* \subseteq T$. By applying Lemma 2.7, we have $F^* \in \mathcal{R}^o_{(S,S)}(L^*) \subseteq \mathcal{R}^o_{(B,S)}(L^*)$. The proof for the pessimistic case is similar. ■

Our initial assumption for the models in this chapter is that $E_f = E$ (see Section 2.2), i.e., the follower controls all edges in the graph. From the above analysis, we conclude that the BST$_{(X,S)}$ problem can be simplified by setting $E_f = \bar{T}$, where $\bar{T}$ is arbitrarily chosen from either $\mathcal{R}^o_{(S,S)}(\emptyset)$ or $\mathcal{R}^p_{(S,S)}(\emptyset)$ under the optimistic or pessimistic rule, respectively. This reduction of the follower's edge set simplifies the solution procedures discussed next.

Finally, we note that the results of this subsection are established for both the optimistic and pessimistic rules simultaneously. Therefore, in the following discussion, we focus on the optimistic version of the BST$_{(X,S)}$ problem, $X \in \{B, S\}$. The pessimistic version can be handled similarly.

2.5.2 Algorithm for the BST$_{(B,S)}$ Problem

For the BST$_{(B,S)}$ problem, recall that the leader's objective is to minimize the maximum edge weight of the resulting spanning tree with respect to the leader's weight vector c. Assume $\bar{T} \in \mathcal{R}^o_{(S,S)}(\emptyset)$, and $(\bar{u}, \bar{v})$ is the maximum weight edge of $\bar{T}$ with respect to the leader's cost vector c. Hence, we obtain an upper bound on the optimal objective function of the bilevel problem as $g_\ell(\emptyset, \bar{T}) = c_{\bar{u}\bar{v}}$, which corresponds to $L = \emptyset$. Therefore, in order to decrease her objective function value, the leader needs to prevent the follower from selecting $(\bar{u}, \bar{v})$. According to Lemma 2.8, the only possibility for the leader to remove $(\bar{u}, \bar{v})$ from the follower's decision is to select some edge (i, j) in E_ℓ such that the cycle in $G[\bar{T} \cup (i, j)]$ contains $(\bar{u}, \bar{v})$.

Based on this general idea, we propose a greedy algorithm for solving BST$_{(B,S)}$, where we attempt to decrease the objective function value of the upper level in every iteration. After obtaining the follower's reaction at the most recent iteration, the leader adds the lowest weight edge (with respect to the weights given by c) that allows the leader to "forbid" the maximum weight edge of the follower's current solution. The algorithm terminates if there is no further improvement is possible for the leader. The sketch of the algorithm is provided in Algorithm 4.

Algorithm 4 Algorithm for the $\text{BST}_{(B,S)}$ problem under the optimistic rule

Input $G = (V, E), E_\ell, c, d$
1: $L^0 \leftarrow \emptyset; k \leftarrow 0; \mathcal{E}_\ell \leftarrow E_\ell$
2: **while** $|\mathcal{E}_\ell| \neq 0$ **do**
3: $\quad F^k \in \mathcal{R}^o_{(S,S)}(L^k)$
4: $\quad T^k \leftarrow L^k \cup F^k$
5: $\quad$ Choose $(\bar{u}, \bar{v}) \in \text{argmax}_{(i,j)\in T^k} c_{ij}$ and find sets $S_{T^k}(\bar{u}, \bar{v})$ and $S'_{T^k}(\bar{u}, \bar{v})$
6: $\quad E'_\ell \leftarrow \{(i,j) \in \mathcal{E}_\ell : \ i \in S_{T^k}(\bar{u}, \bar{v}), j \in S'_{T^k}(\bar{u}, \bar{v}), c_{ij} < c_{\bar{u}\bar{v}}\}$
7: $\quad$ **if** $E'_\ell \neq \emptyset$ **then**
8: $\qquad$ Choose edge $e \in \arg\min_{(i,j)\in E'_\ell}\{c_{ij}\}$
9: $\qquad L^{k+1} \leftarrow L^k \cup \{e\}$
10: $\qquad \mathcal{E}_\ell \leftarrow \mathcal{E}_\ell \setminus \{e\}$
11: $\qquad k \leftarrow k + 1$
12: $\quad$ **else**
13: $\qquad$ **break**
14: $\quad$ **end if**
15: **end while**
Return $z(L^k)$ and L^k

Note that in line 3 of Algorithm 4, the lower-level solution F^k corresponds to the follower's reaction in the $\text{BST}_{(S,S)}$ problem. By Lemma 2.7, F^k is also contained in $\mathcal{R}^o_{(B,S)}(L^k)$. Furthermore, since we only add one edge to L^{k-1} in line 9, then by Lemma 2.8 F^k can be constructed by simply deleting one edge (u', v') of F^{k-1}. If $(u', v') \in \arg\max_{(i,j)\in F^{k-1}} c_{ij}$, then the leader's objective function g_ℓ decreases. Otherwise, g_ℓ does not change. Hence, the objective function value of the upper-level problem is nonincreasing after every iterations. Next, we establish the correctness of the outlined algorithm.

Lemma 2.9. *For the $BST_{(B,S)}$ problem, let $L \subseteq E_\ell$, and define $E'_\ell = \{(i,j) \in E_\ell : \ c_{ij} \leq z(L)\}$. If $L' \subseteq E'_\ell$ and $G[L \cup L']$ is acyclic, then $z(L \cup L') \leq z(L)$.*

Proof. Let $F \in \mathcal{R}^o_{(S,S)}(L)$. Then by Lemma 2.7, we have $F \in \mathcal{R}^o_{(B,S)}(L)$ and $z(L) = \max_{(u,v)\in L\cup F} c_{uv}$. Furthermore, because $L \cup L'$ is a feasible leader's decision, then by Theorem 2.1 there exists $F' \in \mathcal{R}^o_{(S,S)}(L \cup L') \subseteq \mathcal{R}^o_{(B,S)}(L \cup L')$ such that $F' \subseteq F$. Note that by construction, $\max_{(u,v)\in F'} c_{uv} \leq \max_{(u,v)\in F} c_{uv}$ and $\max_{(u,v)\in L\cup L'} c_{uv} = \max_{(u,v)\in L} c_{uv}$. Hence, $z(L \cup L') = \max_{(u,v)\in L\cup L'\cup F'} c_{uv} \leq \max_{(u,v)\in L\cup F} c_{uv} = z(L)$. ∎

Lemma 2.10. *Let (L, F) be a bilevel feasible solution of the $BST_{(B,S)}$ problem. Assume that* *there exists an edge $(u,v) \in E_\ell$ such that $G[L \cup (u,v)]$ contains cycle C. If $c_{uv} \leq z(L)$ and $(u',v') \in C$, then $z((L \setminus (u',v')) \cup (u,v)) \leq z(L)$.*

Proof. Let $L' = (L \setminus (u',v')) \cup (u,v)$ and $F' \in \mathcal{R}^o_{(B,S)}(L')$. Observe that $L' \cup F \in \mathcal{T}(G)$ and thus, $F \in \mathcal{P}(L')$. Then $g_f(L',F') \leq g_f(L',F)$, i.e., $\sum_{(i,j)\in F'} d_{ij} \leq \sum_{(i,j)\in F} d_{ij}$.

Similarly, $F' \in \mathcal{P}(L)$, and $F \in \mathcal{R}^o_{(B,S)}(L)$ by their definition. Then we have $\sum_{(i,j)\in F} d_{ij} \leq \sum_{(i,j)\in F'} d_{ij}$. It immediately follows that $\sum_{(i,j)\in F} d_{ij} = \sum_{(i,j)\in F'} d_{ij}$.

Finally, recall that $F' \in \mathcal{R}^o_{(B,S)}(L')$ and $F \in \mathcal{P}(L')$. By construction, $\max_{(i,j)\in L'} c_{ij} \leq z(L)$. Therefore, summarizing the above derivations we conclude that $z(L') = \max_{(i,j)\in L'\cup F'} c_{ij} \leq \max_{(i,j)\in L'\cup F} c_{ij} \leq z(L)$, which implies the result. ■

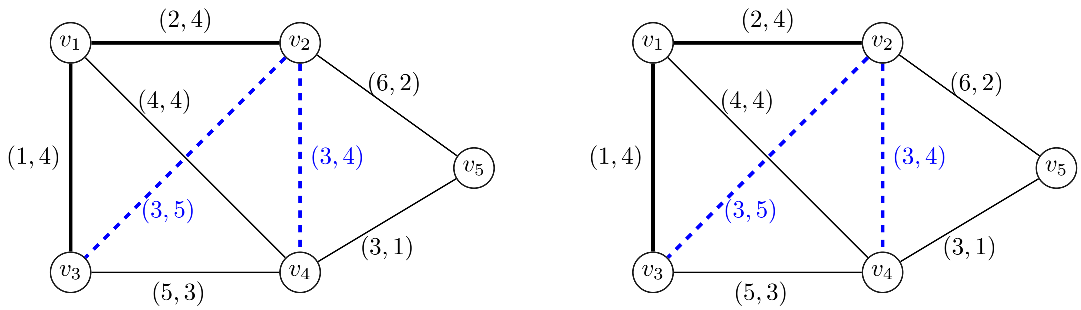
Denote by $\mathcal{L}^* = \{L^* \subseteq E_\ell : \ z(L^*) \leq z(L) \ \forall L \subseteq E_\ell\}$ the set of all optimal leader's decisions for the $\text{BST}_{(B,S)}$ problem. Based on the above two technical results provided by Lemmas 2.9 and 2.10, we conclude:

Proposition 2.6. *Let L^* be an optimal leader's decision for the $BST_{(B,S)}$ problem. Assume that edge $(u,v) \in E_\ell$ and $c_{uv} \leq z(L^*)$. If $G[L^* \cup (u,v)]$ contains cycle C and $(u',v') \in C \cap L^*$, then $(L^* \setminus (u',v')) \cup (u,v) \in \mathcal{L}^*$. Otherwise, $G[L^* \cup (u,v)]$ is acyclic and $L^* \cup (u,v) \in \mathcal{L}^*$.*

From the above analysis, we conclude that the $\text{BST}_{(B,S)}$ problem may have multiple optimal solutions. We provide an example to demonstrate this observation in Figure 3, where the leader's edge set $E_\ell = \{(v_1,v_2),(v_1,v_3),(v_2,v_3),(v_2,v_4)\}$. Observe that $L^* = \{(v_2,v_3),(v_2,v_4)\}$ is an optimal solution of the leader, which is depicted in Figure 3(a) with dashed blue edges. The corresponding reaction of the follower $F^* = \{(v_1,v_3),(v_4,v_5)\}$.

Observe that the maximum weight with respect to c in $L^* \cup F^*$ is 3. Furthermore, $c_{v_1v_2} = 2$ and $c_{v_1v_3} = 1$. Thus, based on Proposition 2.6, we conclude that $\{(v_1,v_3),(v_2,v_3),(v_2,v_4)\}$, $\{(v_1,v_2),(v_1,v_3),(v_2,v_4)\}$, $\{(v_1,v_2),(v_1,v_3),(v_2,v_4)\}$ are also optimal leader's decisions.

If we apply Algorithm 4, then we first compute $T^0 = \{(v_1,v_3),(v_3,v_4),(v_4,v_5),(v_2,v_5)\}$. To remove the maximum weight edge (v_2,v_5) in T^0 from the follower's decision, the algorithm prescribes choosing edge (v_1,v_2), while (v_1,v_3) is then removed from the follower's decision. Consequently, by performing two additional iterations, edges (v_1,v_3) and (v_2,v_4) are added to the leader's decision, and the follower's reaction solution reduces to $\{(v_4,v_5)\}$. Since



(a) $\{(v_2, v_3), (v_2, v_4)\}$ is a leader's optimal solution.

(b) $\{(v_1, v_2), (v_1, v_3), (v_2, v_4)\}$ is an alternative optimal solution of the leader obtained by Algorithm 4.

Figure 3: An illustrative example of multiple optimal leader's decisions for the $\mathrm{BST}_{(B,S)}$ problem. For every edge (i, j), pair (c_{ij}, d_{ij}) depicted above each edge denotes the leader's and the follower's edge weights, respectively. Bold and bold dashed edges (the latter in blue) are controlled by the leader, while the bold dashed edges (in blue) correspond to a leader's optimal solution.

there does not exist any edge to remove (v_4, v_5) from the follower's decision, the algorithm terminates with the leader's decision $\{(v_1, v_2), (v_1, v_3), (v_2, v_4)\}$ shown in Figure 3(b).

Theorem 2.2. *Algorithm 4 returns an optimal solution of the* $BST_{(B,S)}$ *problem.*

Proof. Define z^* to be the optimal objective function value. Let L^k be the solution obtained after iteration k, and denote $\mathcal{L}^*_k = \{L \in \mathcal{L}^* : \ L^k \subseteq L\}$ as the set of optimal leader's decision containing L^k. We next show that $\mathcal{L}^*_k$ is not empty for each k by induction.

(i) For $k = 0$, $\mathcal{L}^*_0 = \mathcal{L}^*$ is clearly not empty.

(ii) Suppose that the statement holds for iteration k, we then prove that the statement also holds for iteration $k+1$. Let $T^k = L^k \cup F^k$, where $F^k \in \mathcal{R}^o_{(S,S)}(L^k)$, and $(\bar{u}, \bar{v})$ is the maximum weight edge of T^k with respect to c.

Next, assume that the algorithm adds edge (i', j') at iteration $k+1$, that is $L^{k+1} = L^k \cup (i', j')$. If $L^k \in \mathcal{L}^*_k$, then $z^* = z(L^k) = c_{\bar{u}\bar{v}}$, where $(\bar{u}, \bar{v})$ is obtained in line 5 during iteration k. Since $c_{i'j'} < c_{\bar{u}\bar{v}}$, then based on Proposition 2.6, L^{k+1} is also an optimal leader's decision. Thus, $L^{k+1} \in \mathcal{L}^*_{k+1}$.

On the other hand, if $L^k \notin \mathcal{L}_k^*$, we have $z^* < c_{\bar{u}\bar{v}}$. For each $L^* \in \mathcal{L}_k^*$, let $\hat{L} = L^* \setminus L^k$. Observe that there must exist an edge $(\hat{u}, \hat{v}) \in \hat{L}$, such that $c_{\hat{u}\hat{v}} < c_{\bar{u}\bar{v}}$, and the path in $G[T^k]$ connecting $\hat{u}$ and $\hat{v}$ contains $(\bar{u}, \bar{v})$. Note that in line 8 of Algorithm 4 we choose the minimum weight edge whose path in $G[T^k]$ contains $(\bar{u}, \bar{v})$, hence $c_{i'j'} \leq c_{\hat{u}\hat{v}} \leq z(L^*)$. Thus, there are three possible cases to consider:

(a) If $(i', j') \in \hat{L}$, then $L^{k+1} \subseteq L^* \in \mathcal{L}_{k+1}^*$.

(b) If $(i', j') \notin \hat{L}$ and $G[L^* \cup (i', j')]$ is acyclic, then $L^* \cup (i', j')$ is also optimal by Proposition 2.6. Therefore, $L^{k+1} \subseteq L^* \cup (i', j') \in \mathcal{L}_{k+1}^*$.

(c) If $(i', j') \notin \hat{L}$ and $G[L^* \cup (i', j')]$ contains cycle C, then consider $(i, j) \in C$. Then $(L^* \setminus (i, j)) \cup (i', j')$ is also optimal by Proposition 2.6. It follows that $L^{k+1} \subseteq (L^* \setminus (i, j)) \cup (i', j') \in \mathcal{L}_{k+1}^*$.

Therefore, after every iteration k, there exists at least one optimal leader's decisions containing L^k. Since $|E_\ell|$ is finite, then the algorithm terminates at L^K for some integer K.

Suppose $L^K \notin \mathcal{L}^*$. Then there exists $L^* \in \mathcal{L}^*$ such that $L^K \subseteq L^*$. Since $z(L^K) > z(L^*)$, then we can find an edge in $L^* \setminus L^K$ to satisfy the property in line 6 of Algorithm 4. Therefore, we have a contradiction with the termination assumption of the algorithm and the result follows. ■

Proposition 2.7. *Algorithm 4 runs in $O(|E_\ell| \cdot |V| + |E| \cdot \log |V|)$ time.*

Proof. In the first iteration, we need to obtain $T^0 \in \mathcal{R}^o_{(S,S)}(\emptyset)$ in line 3, which requires $O(|E| \cdot \log |V|)$ operations. At subsequent iterations, in line 3, by Lemma 2.8, we can solve the follower's reaction problem by deleting one edge, referred to as (u', v'). Recall that (u', v') is the edge of largest order occurring in the path connecting i' and j' in the tree $G[T^k]$. Thus, it takes only $O(|V|)$ time. The algorithm terminates in at most $|E_\ell|$ iterations, since at every step the algorithm removes one edge from E_ℓ. Consequently, the implementation of the proposed algorithm requires $O(|E_\ell| \cdot |V| + |E| \cdot \log |V|)$ time. ■

2.5.3 The BST$_{(S,S)}$ Problem

As briefly outlined in Section 2.1 there are several examples of NP-hard bilevel optimization problems that are two-level generalizations of polynomially solvable single-level problems. Linear programming is, perhaps, the most well-known example as bilevel linear programs are known to be NP-hard [14]. Another interesting example is the bilevel linear assignment problem [13, 43].

The versions of the bilevel spanning problems discussed in Sections 2.3, 2.4 and 2.5.2 are polynomially solvable. However, in the development of the proposed polynomial-time algorithms we exploit the bottleneck structure of either the leader's or the follower's objective functions. On the other hand, Buchheim et al. [20] show that the BST$_{(S,S)}$ problem, where the objective functions at both levels are of the sum type, is NP-hard. It implies that there does not exist a polynomial-time algorithm for solving the BST$_{(S,S)}$ problem unless $\mathsf{P} = \mathsf{NP}$.

In the remainder of the chapter we focus on a solution approach for BST$_{(S,S)}$ that does not have a polynomial-time worst-case performance guarantee. Specifically, in Section 2.5.3.1 we provide a single-level mixed-integer programming (MIP) reformulation of BST$_{(S,S)}$ that allows us to solve this problem by using standard MIP solvers. An iterative preprocessing procedure is then proposed in Section 2.5.3.2 to reduce the size of the MIP model. The scalability of this solution method is further explored with computational experiments in Section 2.5.3.3.

2.5.3.1 Linear Mixed-Integer Formulation

There exist several MIP models for the single-level minimum spanning tree problem and its variations [79, 107]. These MIP models typically exploit either subtour-elimination or flow-based ideas. On the other hand, single-level reformulations of bilevel problems mostly focus on using the optimality conditions (e.g., strong duality of linear programs) to replace the lower-level problem with additional linear or nonlinear constraints [10, 126]. The models proposed in this section represent a combination of these ideas in the context of the bilevel spanning tree problem.

Recall from our discussion in Section 2.5.1 that $\text{BST}_{(S,S)}$ can be simplified by setting E_f to any edge set in $\mathcal{R}^o_{(S,S)}(\emptyset)$. For the remainder of the section, let

$$E_\ell = \{(u_1, v_1), (u_2, v_2), \ldots, (u_{m_\ell}, v_{m_\ell})\},$$

$$E_f = \{(i_1, j_1), (i_2, j_2), \ldots, (i_{n-1}, j_{n-1})\},$$

where edges in E_f are assumed to be ordered under the optimistic rule as $(i_1, j_1) \preceq (i_2, j_2) \preceq \cdots \preceq (i_{n-1}, j_{n-1})$; see Definition 2.3.

Suppose (L, F) is a solution such that $L \cup F \in \mathcal{T}(G)$, where $L \subseteq E_\ell$ and $F \subseteq E_f$. For any edge $(i_t, j_t) \in E_f$, we define directed graph induced by arcs $\mathcal{A}_f^{<t}$ as $\mathcal{G}[\mathcal{A}_f^{<t}] := (V, \mathcal{A}_f^{<t})$, where $\mathcal{A}_f^{<t} = \{(u_q, v_q), (v_q, u_q) : \ (u_q, v_q) \in L\} \cup \{(i_p, j_p), (j_p, i_p) : \ (i_p, j_p) \in F, \ 1 \le p < t\}$. Based on the optimality conditions in Lemma 2.6, we have that (L, F) is a bilevel feasible solution if and only if the following conditions are satisfied:

- (i) $G[L]$ does not contain cycles;
- (ii) For any edge $(i_t, j_t) \in E_f$, if vertices i_t and j_t are disconnected in the directed graph $\mathcal{G}[\mathcal{A}_f^{<t}]$, then $(i_t, j_t) \in F$. Otherwise, $(i_t, j_t) \notin F$.

To enforce that $G[L]$ is acyclic, we first define the directed graph $\mathcal{G}[\mathcal{A}_\ell^{\neq k}] := (V, \mathcal{A}_\ell^{\neq k})$, where $\mathcal{A}_\ell^{\neq k} = \{(u_q, v_q), (v_q, u_q) : \ (u_q, v_q) \in L, q \neq k\}$, $k \in \{1, \ldots, m_\ell\}$. Then for any edge (u_k, v_k) in E_ℓ, we consider the shortest path problem from u_k to v_k in graph $\mathcal{G}[\mathcal{A}_\ell^{\neq k}]$ and its dual as the following linear programs (LPs):

$$[P_1] \ \min \sum_{(u,v) \in \mathcal{A}_\ell^k} x_{uv}^k \qquad\qquad [D_1] \ \max \pi_{u_k}^k$$

$$\text{s.t.} \ \ A_\ell^{\neq k} x^k = \begin{cases} 1, & \text{for vertex } u_k \\ 0, & \text{for other vertices} \\ -1, & \text{for vertex } \ v_k \end{cases}, \qquad \text{s.t.} \ \ \pi_u^k - \pi_v^k \le 1 \quad \forall\, (u,v) \in \mathcal{A}_\ell^{\neq k},$$

$$x_{uv}^k \ge 0 \quad \forall\, (u,v) \in \mathcal{A}_\ell^{\neq k}. \qquad\qquad \pi_{v_k}^k = 0.$$

Matrix $A_\ell^{\neq k}$ is the node-arc matrix of graph $\mathcal{G}[\mathcal{A}_\ell^{\neq k}]$. If the leader selects (u_k, v_k), then u_k and v_k need to be disconnected in $\mathcal{G}[\mathcal{A}_\ell^{\neq k}]$ due to the acyclic condition on $G[L]$. Hence, the shortest path problem $[P_1]$ needs to be infeasible. Observe that $\pi = 0$ is a feasible solution of the dual problem $[D_1]$, which implies that $[D_1]$ needs to be unbounded. Then there should exist a feasible solution of the dual problem whose objective function value is n.

Henceforth, by defining $\hat{\mathcal{A}}_\ell^{\neq k} = \{(u_q, v_q), (v_q, u_q) : (u_q, v_q) \in E_\ell, 1 \leq q \leq m_\ell, q \neq k\}$, we propose the following constraints:

$$\pi_u^k - \pi_v^k \leq 1 + M(1 - x_{uv}) \quad \forall\, (u,v) \in \hat{\mathcal{A}}_\ell^{\neq k}, \tag{2.11a}$$

$$n x_{u_k v_k} \leq \pi_{u_k}^k \leq n - 1 + x_{u_k v_k}, \tag{2.11b}$$

$$\pi_{v_k}^k = 0, \tag{2.11c}$$

where $x_{uv} \in \{0,1\}$ is the binary variable indicating whether edge $(u,v) \in E_\ell$ is chosen by the leader, and M is a sufficiently large constant parameter, e.g., $M = n$. If $x_{u_k v_k} = 1$, then constraints (2.11b) ensure that a feasible solution with the objective function value of n exists for problem $[D_1]$. Thus, $(u_k, v_k) \in L$ and the edge is not contained in any cycle of $G[L]$. Otherwise, if $x_{u_k v_k} = 0$, then the edge $(u_k, v_k) \notin L$ and the relationship between $[P_1]$ and $[D_1]$ is not enforced.

To ensure that the second condition for bilevel feasibility of (L, F) is satisfied, see (ii) above, for any $(i_t, j_t) \in E_f$ the follower needs to verify whether there exists a path connecting i_t and j_t in graph $\mathcal{G}[\mathcal{A}_f^{<t}]$. We also formulate this question as the shortest path problem in directed graph $\mathcal{G}[\mathcal{A}_f^{<t} \cup (i_t, j_t)]$ with source vertex i_t and sink vertex j_t. The arc weights of 1 are imposed for all arcs in $\mathcal{A}_f^{<t}$ and the weight of arc (i_t, j_t) is set to n. The corresponding shortest path problem and its dual problem are given by:

$$[P_2]\ \min \sum_{(i,j)\in\mathcal{A}_f^t} x_{ij}^t + n x_{i_t j_t}^t$$

$$\text{s.t.}\quad A_f^{<t} x^t = \begin{cases} 1, & \text{for vertex } i_t \\ 0, & \text{for other vertices}, \\ -1, & \text{for vertex } j_t \end{cases}$$

$$x_{ij}^t \geq 0 \quad \forall\, (i_p, j_p) \in \mathcal{A}_f^{<t} \cup (i_t, j_t).$$

$$[D_2]\ \max\ \pi_{i_t}^t - \pi_{j_t}^t$$

$$\text{s.t.}\quad \pi_{i_p}^t - \pi_{j_p}^t \leq 1 \quad \forall\, (i_p, j_p) \in \mathcal{A}_f^{<t},$$

$$\pi_{i_t}^t - \pi_{j_t}^t \leq n.$$

Matrix $A_f^{<t}$ is the node-arc matrix of graph $\mathcal{G}[\mathcal{A}_f^{<t} \cup (i_t, j_t)]$. If i_t and j_t are disconnected in $\mathcal{G}[\mathcal{A}_f^{<t}]$, then the optimal objective function value of problem $[P_2]$ is n. Otherwise, the optimal value is at most $n-1$. Let $\hat{\mathcal{A}}_f^{<t} = \{(u_q, v_q), (v_q, u_q) : \ 1 \leq q \leq m_\ell\} \cup \{(i_p, j_p), (j_p, i_p) : 1 \leq p < t\}$ and let $\hat{A}_f^{<t}$ be the node-arc matrix of graph $\mathcal{G}[\hat{\mathcal{A}}_f^{<t} \cup \{(i_t, j_t)\}]$. By applying the strong duality property of LPs, we formulate the following set of constraints to indicate whether the follower chooses (i_t, j_t):

$$
\hat{A}_f^{<t} x^t = \begin{cases} 1, & \text{for vertex } i_t \\ 0, & \text{for other vertices }, \\ -1, & \text{for vertex } \ j_t \end{cases} \tag{2.12a}
$$

$$
x^t_{u_q v_q} + x^t_{v_q u_q} \leq x_{u_q v_q} \qquad \forall\, (u_q, v_q) \in E_\ell, \tag{2.12b}
$$

$$
x^t_{i_p j_p} + x^t_{j_p i_p} \leq y_{i_p j_p} \qquad \forall\, p < t, \tag{2.12c}
$$

$$
x^t_{ij} \geq 0 \qquad \forall\, (i,j) \in \hat{\mathcal{A}}_f^{<t} \cup (i_t, j_t), \tag{2.12d}
$$

$$
\pi^t_{u_q} - \pi^t_{v_q} \leq 1 + M(1 - x_{u_q v_q}) \ \ \forall\, (u_q, v_q) \in E_\ell, \tag{2.12e}
$$

$$
\pi^t_{v_q} - \pi^t_{u_q} \leq 1 + M(1 - x_{u_q v_q}) \ \ \forall\, (u_q, v_q) \in E_\ell, \tag{2.12f}
$$

$$
\pi^t_{i_p} - \pi^t_{j_p} \leq 1 + M(1 - y_{i_p j_p}) \quad \forall\, p \leq t, \tag{2.12g}
$$

$$
\pi^t_{j_p} - \pi^t_{i_p} \leq 1 + M(1 - y_{i_p j_p}) \quad \forall\, p \leq t, \tag{2.12h}
$$

$$
n y_{i_t j_t} \leq \pi^t_{i_t} \leq n - 1 + y_{i_t j_t}, \quad \pi^t_{j_t} = 0, \tag{2.12i}
$$

$$
\sum_{(i,j) \in \hat{\mathcal{A}}_f^t} x^t_{ij} + n x^t_{i_t j_t} = \pi^t_{i_t}, \tag{2.12j}
$$

where $y_{i_p j_p} \in \{0, 1\}$ denotes whether edge $(i_p, j_p) \in E_f$ is chosen by the follower, and M is a sufficiently large constant parameter, e.g., $M = n$. If the value of $\pi^t_{i_t}$ achieves n, then constraint (2.12i) requires the follower to pick edge (i_t, j_t), i.e., $y_{i_t j_t} = 1$. Otherwise, $y_{i_t j_t} = 0$.

Summarizing the discussion above, the bilevel spanning tree problem $\text{BST}_{(S,S)}$ can be formulated as the following MIP:

$$
\min_{x,y} \sum_{(u_q, v_q) \in E_\ell} c_{u_q v_q} x_{u_q v_q} + \sum_{(i_p, j_p) \in E_f} c_{i_p j_p} y_{i_p j_p} \tag{2.13a}
$$

$$
\text{s.t.} \ \ (2.11a) - (2.11c) \quad \forall\, k \in \{1, \ldots, m_\ell\}, \tag{2.13b}
$$

$$(2.12a) - (2.12j) \quad \forall\, t \in \{1, \ldots, n-1\}, \tag{2.13c}$$

$$\sum_{(u_q,v_q)\in E_\ell} x_{u_q v_q} + \sum_{(i_p,j_p)\in E_f} y_{i_p j_p} = n-1, \tag{2.13d}$$

$$x_{ij} + y_{ij} \leq 1 \quad \forall\, (i,j) \in E_\ell \cap E_f, \tag{2.13e}$$

$$x_{u_q v_q} \in \{0,1\} \quad \forall\, (u_q, v_q) \in E_\ell, \tag{2.13f}$$

$$y_{i_p j_p} \in \{0,1\} \quad \forall\, (i_p, j_p) \in E_f, \tag{2.13g}$$

where constraint (2.13d) ensures that there are $n-1$ edges in the resulting solution. Based on the discussion and derivations above, we state the following result.

Theorem 2.3. *Any optimal solution of (2.13) is also an optimal solution of the* $\text{BST}_{(S,S)}$ *problem.*

In formulation (2.13), we have $O(m_\ell + n)$ binary variables. Constraints (2.11) contain $O(n)$ continuous variables and $O(m_\ell)$ constraints for each $(u_k, v_k) \in E_\ell$, where $k \in \{1, 2, \ldots, m_\ell\}$. In (2.12) there are $O(m_\ell + n)$ continuous variables and $O(m_\ell + n)$ linear constraints for each $(i_t, j_t) \in E_f$, where $t \in \{1, 2, \ldots, n-1\}$. Therefore, formulation (2.13) consists of $O(m_\ell + n)$ binary variables, $O(nm_\ell + n^2)$ continuous variables, and $O(m_\ell^2 + n^2)$ constraints. Note that the total number of constraints and variables in (2.13) is polynomially bounded by the problem size.

2.5.3.2 Preprocessing

In this subsection, we describe an iterative preprocessing procedure to reduce the size of the MIP model (2.13). Assume that the input of the $\text{BST}_{(S,S)}$ problem is given by graph $G = (V, E)$, where $E_\ell \subseteq E$ and $E_f = E$. Then recall from the result of Corollary 2.1 (along with the discussion at the end of Section 2.5.1 and at the beginning of Section 2.5.3.1) that we can first set E_f to be any follower's reaction in $\mathcal{R}^o_{(S,S)}(\emptyset)$. After this simplification, we have $E_f \in \mathcal{R}^o_{(S,S)}(\emptyset)$ and $|E_f| = |V| - 1$. We refer to this simplification procedure as the *first preprocessing step*.

The *second preprocessing step* is based on the observation that for any edge $(i,j) \in E_\ell$, if c_{ij} is greater than the weight of any edge in E_f, then (i,j) can not be in the leader's optimal decision. Otherwise, we can construct a solution by removing (i, j) from the leader's decision to obtain a better objective function value.

Algorithm 5 Preprocessing for the $\text{BST}_{(S,S)}$ problem

Input $G = (V, E), E_\ell, c, d$
Choose $E_f \in \mathcal{R}^o_{(S,S)}(\emptyset)$
Sort E_f in the nondecreasing order as $(i_1, j_1) \preceq (i_2, j_2) \preceq \cdots \preceq (i_{n-1}, j_{n-1})$
$k \leftarrow 0$; $\mathcal{E}^0_\ell \leftarrow E_\ell$; $\mathcal{E}^0_f \leftarrow E_f$
while $k = 0$ or $\mathcal{E}^k_\ell \neq \mathcal{E}^{k-1}_\ell$ **do**
  $\bar{c} \leftarrow \max_{(i,j)\in\mathcal{E}^k_f} c_{ij}$
  $\mathcal{E}^{k+1}_\ell \leftarrow \{(i, j) \in \mathcal{E}^k_\ell : \ c_{ij} < \bar{c}\}$
  $\mathcal{E}^{k+1}_f \leftarrow \emptyset$
  for $(i_t, j_t) \in \mathcal{E}^k_f$ **do**
    $\hat{\mathcal{A}}^{<t}_f \leftarrow \{(u, v), (v, u) : \ (u, v) \in \mathcal{E}^{k+1}_\ell\} \cup \{(i_p, j_p), (j_p, i_p) : \ 1 \leq p < t\}$
    if there exists a path connecting i_t and j_t in graph $\mathcal{G}[\hat{\mathcal{A}}^{<t}_f] = (V, \hat{\mathcal{A}}^{<t}_f)$ **then**
      $\mathcal{E}^{k+1}_f \leftarrow \mathcal{E}^{k+1}_f \cup (i_t, j_t)$
    end if
  end for
  $k \leftarrow k + 1$
end while
$E_\ell \leftarrow \mathcal{E}^k_\ell$; set $y_{ij} = 1$ in formulation (2.13) for all $(i, j) \in E_f \setminus \mathcal{E}^k_f$

In the *third preprocessing step*, we identify edges in E_f that are always contained in the follower's reaction solution for any leader's decision. Recall from Section 2.5.3.1 that:

(*i*) for the follower's edges set we assume $E_f = \{(i_1, j_1), (i_2, j_2), \ldots, (i_{n-1}, j_{n-1})\}$ and $(i_1, j_1) \preceq (i_2, j_2) \preceq \cdots \preceq (i_{n-1}, j_{n-1})$;

(*ii*) edge (i_t, j_t), $t \in \{1, 2, \ldots, n-1\}$, is chosen by the follower if and only if i_t and j_t are disconnected in directed graph $\mathcal{G}[\mathcal{A}^{<t}_f]$;

(*iii*) $\hat{\mathcal{A}}^{<t}_f = \{(u_q, v_q), (v_q, u_q) : \ 1 \leq q \leq m_\ell\} \cup \{(i_p, j_p), (j_p, i_p) : 1 \leq p < t\}$ and thus, $\mathcal{A}^{<t}_f \subseteq \hat{\mathcal{A}}^{<t}_f$.

In view of the above, we can verify whether there exists a path connecting i_t and j_t in graph $\mathcal{G}[\hat{\mathcal{A}}^{<t}_f]$. If this is not the case, then i_t and j_t are also disconnected in graph $\mathcal{G}[\mathcal{A}^{<t}_f]$ regardless of the leader's decision. Therefore, we can set $y_{i_t j_t} = 1$ in (2.13).

Naturally, we can repeat the second and the third steps in an iterative manner, as formalized in the pseudo-code of Algorithm 5. Specifically, at iteration 0, we initialize $\mathcal{E}^0_\ell$ as E_ℓ and $\mathcal{E}^0_f$ as a follower's reaction in $\mathcal{R}^o_{(S,S)}(\emptyset)$. Set $\mathcal{E}^k_\ell$ denotes edges controlled by the leader after k iterations of the outlined procedure. Similarly, $E_f \setminus \mathcal{E}_f^k$ denotes the follower's edges that are fixed to be selected in the follower's reaction solution for any leader's decision. The algorithm stops whenever $\mathcal{E}_\ell^k$ coincides with $\mathcal{E}_\ell^{k-1}$ and thus no additional leader's edges are removed.

2.5.3.3 Computational Experiments

Test instances. We test our MIP reformulation (2.13) using the graphs obtained from the existing library of test instances for the Steiner tree problem [64]. Note that the latter is a well-studied class of combinatorial optimization problems, and shares clear similarities to the minimum spanning tree problem. Hence, test instances for the Steiner tree problem (after some appropriate modifications as described in details below) provide a suitable computational platform for our study. Specifically, in our experiments we use the following graph classes from [64]:

- sparse with random weights set **B** (50 - 100 nodes, 18 instances)
- sparse with Euclidian weights set **P6E** (100 - 200 nodes, 15 instances)
- sparse with random weights set **P6Z** (100 - 200 nodes, 15 instances)
- sparse with random weights set **C** (500 nodes, 12 instances)
- sparse with random weights set **MC** (400 nodes, 3 instances)

In our experiments, the number of edges controlled by the leader, $|E_\ell|$, is assumed to be a fraction, ρ, of the total number of edges $|E|$. That is, $|E_\ell| = \lceil \rho |E| \rceil$, where $\rho \in \{0.1, 0.2, 0.3\}$. For all of the above instances we use two approaches for constructing the leader's edge set:

- in the first approach, referred to as *random*, we consequently choose one of the edges from E with equal probability and add it to the leader's edge set, E_ℓ, until $|E_\ell| = \lceil \rho |E| \rceil$.
- in the second approach, referred to as *random with a degree constraint*, similar to the previous approach we randomly select one of the edges to be added to E_ℓ; however, we consider as candidates only edges in E that are incident to at least one vertex with its degree no less than the third quartile of all vertex degrees in the graph.

The rationale behind the second approach is based on the motivating network design application outlined in Section 2.1. Recall that the leader (e.g., the central government) prefers to construct only the most important connections (edges) in the graph. Naturally, the latter consideration can be modeled by assuming that for such connections at least one of its end-vertices is "highly influential," that is a vertex with a reasonably high degree (i.e., the number of the vertex connections). This concept is also known as the *degree centrality* in the network analysis literature [88].

Finally, for each $(i, j) \in E$ the follower's and leader's edge weights (costs) are set to $c_{ij} = w_{ij}r_{ij}$ and $d_{ij} = w_{ij} - c_{ij}$, respectively, where w_{ij} is an original edge weight from the corresponding data set in [64] and r_{ij} is generated following the uniform distribution over $[0, 1]$, i.e., $r_{ij} \sim U(0, 1)$.

Computational setup. Our numerical experiments are conducted using CPLEX 12.80 [51] on a Windows 7 PC with a 3.6 GHz CPU and 32 GB of RAM. We set the time limit to 3,600 seconds. For each class of test instance considered (i.e., one of the above graph classes along with the specific value of ρ and one of the two construction methods of the leader's edge set), we report the solver's average performance over 10 randomly generated instances. In particular, the average solution times (which include the times used by the preprocessing procedure) are reported in seconds.

Furthermore, to explore the quality of the MIP formulation we also report the integrality gap (IG). It is computed as IG(%) $= \frac{z^* - z_{LP}}{z^*} \times 100$, where z^* and z_{LP} are the optimal objective function values of the MIP model and its LP relaxation, respectively. The average integrality gap is calculated across the instances solved to optimality within the time limit. If the solver is not able to solve some instances for a particular size within the time limit, then we show the average optimality gap (OG) that is reported by the solver. The number of instances for which the solver can not find an optimal solution within the time limit are indicated as the superscripts over the optimality gap values.

Results and discussion. The first set of our computational experiments focuses on evaluating the effect of the preprocessing procedure. In particular, we consider dataset **B** and

$\rho = 0.3$. The computational results are summarized in Tables 1 and 2. The number of edges in E_ℓ after preprocessing is given in the column denoted by "new $|E_\ell|$". The number of edges in E_f that are fixed (by the preprocessing procedure) to belong to the follower's reaction set for any leader's decision is reported in the column denoted by "FixE_f". The number of iterations of the preprocessing procedure is reported in the column denoted by "Iter".

As expected, compared to the input of $E_f = E$, the preprocessing procedure drastically reduces the number of edges that need to be considered by the follower. For example, for instance b01 in Table 1, before preprocessing the cardinalities of the edge sets controlled by the leader and the follower are 19 and 63, respectively. After the preprocessing procedure, $|E_\ell|$ is reduced on average to 18.3; furthermore, on average 30.8 edges are fixed to be in the follower's reaction set. Thus, the number of the follower's edges that need to be considered is on average $|V| - 1 - 30.8 = 18.2$, compared to the initial $|E_f| = 63$. As for the MIP model size, the original formulation contains 68 binary variables, 4,331 continuous variables and 7,221 constraints. After preprocessing the model size reduces to about 37 binary variables, 2,342 continuous variables and 2,681 constraints on average. Consequently, the running times reported in Tables 1 and 2 indicate significant improvements after preprocessing for both construction methods of the leader's edge set.

Another observation from Tables 1 and 2 is that with the increase of the graph density and the number of edges controlled by the leader, the positive effects of the preprocessing procedure decrease, which is quite intuitive given the key ideas outlined in Section 2.5.3.2. Nevertheless, the benefits of the preprocessing procedure are considerable. Thus, we apply preprocessing procedure in the remaining set of our computational experiments for all test instances.

We next investigate the performance of the MIP model (2.13) for test sets **B**, **P6E** and **P6Z** and different sizes of the leader's edge set (i.e., the value of ρ); see the results in Tables 3–5. In general, the solver can easily handle these rather small-sized sparse graphs with reasonable running times. The average integrality gaps of instances with $\rho = 0.1$ are typically under 5%, which suggests that the LP relaxation of the MIP formulation can provide a sufficiently tight lower bound for the BST problem. With the increase in ρ, it is not surprising that the lower bound becomes weaker with longer runtime. This degradation is due to the growth of the number of binary variables and constraints in the MIP model when the leader is allowed to control more edges.

By comparing the construction methods for E_ℓ, Tables 3–5 show that the integrality gaps between these two methods are reasonably close for each class of the test instances and the same value of ρ. It demonstrates that the tightness of the LP relaxation associated with the MIP model (2.13) depends on the number of leader's edges rather than on the construction method of E_ℓ. However, we observe that in most cases, the performance on instances with E_ℓ randomly constructed is slightly better than that on the random instances generated with a degree constraint. We attribute it to the fact that if the edges in E_ℓ that are incident to the same vertex have similar weights, then it could be more difficult for the branch-and-bound solver to perform pruning. Thus, if we generate the leader's edges using a degree constraint, then for a high-degree vertex, it is possible that E_ℓ contains several edges of similar weights incident with this vertex, which causes the increase of the solver's runtime.

Tables 6 and 7 show the performance for test sets **MC** and **C**, which consist of medium size graphs. We note that in these tables, the symbol "–" means that the MIP model size is too large for the solver to perform initialization. Two observations are consistent with our earlier experiments. Namely, the LP relaxation quality is reasonably good even for larger instances in these test sets and the problems become more difficult as the cardinality of the leader's edge set increases. In test set **C**, all instances have 500 nodes, but different graph densities. By comparing the results for these graphs, as one would expect the problem becomes more difficult as the edge density increases.

We conclude this section by summarizing our main observations.

(*i*) The preprocessing procedure is useful in reducing the size of the MIP model, especially for large-scale sparse graphs, and hence leads to significant running time improvements.

(*ii*) The MIP formulation (2.13) has a reasonably good LP relaxation quality, particularly when the leader's edge set is reasonably small.

(*iii*) The performance of the MIP model degrades with the increase of either the number of nodes, the number of leader's edges or the graph density. Although, the MIP model can be effectively handled by a standard solver for moderately sized sparse graphs, large-scale graphs are still intractable and more memory intensive with the current MIP model. We believe that more sophisticated preprocessing procedures and advanced branch-and-cut strategies using the underlying network structures are two interesting directions of the future research.

2.6 Concluding Remarks

In this chapter, we consider a class of bilevel spanning tree problems with two types of objective functions, namely, the sum and bottleneck. We show that whenever one of the objective functions (either at the lower or upper level) involves the bottleneck-type function, then the problem admits a polynomial-time solution algorithm. In particular, our algorithms exploit the structural properties that are enforced by the presence of the bottleneck objective. The obtained results hold for both the optimistic and pessimistic cases. Furthermore, the min-max version of the problem is also polynomially solvable regardless of the objective function type. These results are particularly interesting if one recalls that for many other classes of polynomially solvable single-level optimization problems, their bilevel extensions are often NP-hard.

Table 1: Computational results for test set **B**, where $\rho = 0.3$ and E_ℓ is randomly constructed. The number of edges in E_ℓ after preprocessing is given in the column denoted by "new E_ℓ". The number of edges in E_f that are fixed (by the preprocessing procedure) to belong to the follower's reaction solution for any leader's decision is reported in the column denoted by "FixE_f". The number of iterations of the preprocessing procedure is reported in the column "Iter".

				with preprocessing					no preprocessing	
Ins.	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert E_\ell\rvert$	new $\lvert E_\ell\rvert$	FixE_f	IG(%)	Time	Iter	IG(%)	Time
b01	50	63	19	18.3	30.8	5.15	0.12	1.2	6.02	0.18
b02	50	63	19	15.3	33.8	2.14	0.09	2.8	5.77	0.17
b03	50	63	19	16.4	32.6	2.87	0.09	2.3	6.01	0.15
b04	50	100	30	28.2	21.7	10.03	0.51	1.9	14.72	1.02
b05	50	100	30	28.7	20.9	11.84	0.93	1.5	16.39	1.62
b06	50	100	30	29.8	20	21.29	1.42	1	22.35	2.62
b07	75	94	29	25.9	48.2	4.83	0.23	2.5	6.71	0.47
b08	75	94	29	24	50	2.1	0.18	3.2	7.63	0.52
b09	75	94	29	27.7	46.3	5.07	0.22	1.7	7.75	0.44
b10	75	150	45	44.6	30.2	14.43	7.07	1.1	14.81	16.5
b11	75	150	45	44.3	29.8	16.68	7.73	1.3	17.91	15.4
b12	75	150	45	44	30.7	12.24	12.22	1.2	14.32	31.48
b13	100	125	38	35.5	63.5	5.75	0.59	2	8.03	1.68
b14	100	125	38	34.7	64.4	5.66	0.39	2.4	10.73	1.09
b15	100	125	38	35.2	63.9	5.68	0.58	2	8.91	1.42
b16	100	200	60	58.9	40.9	15.41	71.61	1.1	17.84	171.83
b17	100	200	60	59.4	41.1	19.8	123.6	1	20.52	300.86
b18	100	200	60	58.5	41.2	17.35	156.07	1	18.02	417.07

Table 2: Computational results for test set **B**, where $\rho = 0.3$ and E_ℓ is randomly constructed with a degree constraint. The number of edges in E_ℓ after preprocessing is given in the column denoted by "new E_ℓ". The number of edges in E_f that are fixed (by the preprocessing procedure) to belong to the follower's reaction solution for any leader's decision is reported in the column denoted by "FixE_f". The number of iterations of the preprocessing procedure is reported in the column "Iter".

				with preprocessing					no preprocessing	
Ins.	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert E_\ell\rvert$	new $\lvert E_\ell\rvert$	FixE_f	IG(%)	Time	Iter	IG(%)	Time
b01	50	63	19	16.5	32.7	4.11	0.1	1.8	7.14	0.2
b02	50	63	19	16.2	32.8	2.46	0.07	2.6	5.71	0.23
b03	50	63	19	16.5	32.6	4.09	0.11	2.1	6.84	0.21
b04	50	100	30	28.8	21.2	11.45	0.44	1.5	13.77	0.7
b05	50	100	30	28.5	21.1	12.85	1.25	1.2	17.89	2.25
b06	50	100	30	29.6	20.5	19.4	1.51	1	20.31	2.78
b07	75	94	29	25	49.1	3.69	0.26	2.7	7.24	0.63
b08	75	94	29	23.6	50.4	2.95	0.18	3.2	7.43	0.46
b09	75	94	29	28.1	46	4.79	0.24	1.6	6.46	0.49
b10	75	150	45	44.5	31.4	12.84	14.94	1.1	12.84	25.05
b11	75	150	45	44.8	30.3	12.63	7.85	1.2	12.63	15.99
b12	75	150	45	44.2	30.8	12.08	10.03	1.4	13.35	19.74
b13	100	125	38	33.5	65.5	3.79	0.46	3	7.85	1.4
b14	100	125	38	35.4	63.6	6.41	0.49	2.1	9.57	1.73
b15	100	125	38	35	64	5.51	0.46	1.9	8.83	1.57
b16	100	200	60	58.6	41.9	14.29	117.65	1.4	16.63	339.18
b17	100	200	60	59.2	43.2	18.05	151.17	1	19.31	373.69
b18	100	200	60	58.1	42.5	16.55	293.44	1	17.66	911.22

Table 3: Computational results for test set **B** with two construction methods of E_ℓ.

			Random						Random with a degree constraint					
			$\rho = 0.1$		$\rho = 0.2$		$\rho = 0.3$		$\rho = 0.1$		$\rho = 0.2$		$\rho = 0.3$	
Ins.	$\lvert V\rvert$	$\lvert E\rvert$	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time
b01	50	63	0.34	0.02	1.27	0.04	5.15	0.12	0.32	0.02	0.94	0.04	4.11	0.1
b02	50	63	0.39	0.01	0.91	0.08	2.14	0.09	0.26	0.02	1.68	0.04	2.46	0.07
b03	50	63	0.06	0.03	1.61	0.04	2.87	0.09	0	0.02	1.27	0.03	4.09	0.11
b04	50	100	1.24	0.05	4.63	0.15	10.03	0.51	1.89	0.05	6.29	0.18	11.45	0.44
b05	50	100	1.85	0.06	7.71	0.23	11.84	0.93	3.28	0.06	6.9	0.27	12.85	1.25
b06	50	100	2.25	0.05	12.68	0.27	21.29	1.42	2.31	0.05	15.03	0.26	19.4	1.51
b07	75	94	0.4	0.02	0.93	0.06	4.83	0.23	0.04	0.01	2.27	0.09	3.69	0.26
b08	75	94	0.01	0.03	1.09	0.08	2.1	0.18	0.31	0.04	1.14	0.09	2.95	0.18
b09	75	94	0.65	0.04	1.86	0.08	5.07	0.22	1.09	0.05	1.7	0.09	4.79	0.24
b10	75	150	2.98	0.15	8.3	0.59	14.43	7.07	2.49	0.09	7.84	0.77	12.84	14.94
b11	75	150	4.22	0.15	10.95	0.82	16.68	7.73	3.98	0.11	11.11	1.03	12.63	7.85
b12	75	150	2.86	0.12	8.16	0.92	12.24	12.22	2.96	0.13	8.6	0.95	12.08	10.03
b13	100	125	1.06	0.07	0.93	0.1	5.75	0.59	0.04	0.03	1.85	0.13	3.79	0.46
b14	100	125	0.34	0.07	2.28	0.15	5.66	0.39	0.96	0.06	2.27	0.2	6.41	0.49
b15	100	125	0.42	0.06	2.77	0.21	5.68	0.58	0.98	0.05	3.12	0.21	5.51	0.46
b16	100	200	3.85	0.22	11.24	2.13	15.41	71.61	4.61	0.25	8.61	3.02	14.29	117.65
b17	100	200	5.03	0.28	10.42	4.63	19.8	123.6	6.32	0.28	13.91	4.86	18.05	151.17
b18	100	200	4.47	0.24	12.93	3.69	17.35	156.07	5.05	0.23	12.71	3.62	16.55	293.44

Table 4: Computational results for test set **P6E** with two construction methods of E_ℓ.

			Random						Random with a degree constraint					
			$\rho = 0.1$		$\rho = 0.2$		$\rho = 0.3$		$\rho = 0.1$		$\rho = 0.2$		$\rho = 0.3$	
Ins.	$\lvert V\rvert$	$\lvert E\rvert$	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time
p619	100	180	3.05	0.12	9.32	1.1	16.08	3.25	2.94	0.15	9.11	0.94	15.27	6.89
p620	100	180	2.33	0.11	8.82	0.75	18.26	4.35	3.48	0.1	7.84	0.73	17.41	8.07
p621	100	180	3.62	0.13	9.95	1.2	16.11	6.76	2.91	0.17	8.54	1.57	15.89	12.08
p622	100	180	1.3	0.14	4.78	0.57	9.87	4.98	1.89	0.14	5.37	0.69	10.12	4.28
p623	100	180	1.77	0.13	6.68	0.96	11.54	7.74	1.94	0.14	7.15	1.14	10.52	12.25
p624	100	180	1.94	0.13	6.72	0.83	9.9	3.14	1.39	0.11	4.32	0.67	12.3	4.75
p625	100	180	1.56	0.15	6.33	0.61	12.96	6.85	2.06	0.13	6.09	0.76	13.31	4.77
p626	100	180	2.07	0.1	7.18	0.91	12.38	3.69	2.18	0.14	8.87	0.88	15.42	4.21
p627	100	180	1.45	0.14	5.97	0.76	10.05	5.28	1.3	0.14	5.48	0.6	9.28	6.04
p628	100	180	0.96	0.15	4.42	0.72	8.78	4.02	1.26	0.14	4.62	0.79	8.35	6.71
p629	100	180	1.62	0.11	6.11	0.81	13.22	8.38	1.93	0.16	5.74	1.08	10.15	4.67
p630	200	370	1.78	0.97	6.15	17.54	9.62	160.02	2.17	0.78	5.48	18.45	10.56	316.67
p631	200	370	1.92	0.73	5.34	11.95	10.1	161.64	1.64	0.77	5.45	14.28	9.87	213.9
p632	200	370	1.73	0.69	5.18	13.67	9.25	160.41	1.95	0.7	6.08	10.21	11.05	207.7
p633	200	370	2.01	0.85	5.36	13.33	10.53	444.63	2.01	0.89	6.59	27.2	10.51	345.38

Table 5: Computational results for test set **P6Z** with two construction methods of E_ℓ.

			Random						Random with a degree constraint					
			$\rho = 0.1$		$\rho = 0.2$		$\rho = 0.3$		$\rho = 0.1$		$\rho = 0.2$		$\rho = 0.3$	
Ins.	$\|V\|$	$\|E\|$	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time	IG(%)	Time
p602	100	180	0.82	0.16	4.46	0.62	7.07	3.26	1	0.14	3.88	0.5	7.64	3.72
p603	100	180	1.87	0.11	5.68	0.63	14.61	5.85	2.42	0.12	7.91	0.91	12.9	4.88
p604	100	180	1.5	0.17	5.53	1.13	8.18	6.59	1.81	0.12	4.84	0.93	8.76	10
p605	100	180	1.95	0.11	6.24	0.78	11.01	6.16	2.9	0.14	5.69	1.07	15.14	5.63
p606	100	180	1.54	0.15	6.28	0.83	11.92	4.24	1.88	0.15	5.7	0.96	11.52	5.17
p607	100	180	1.83	0.14	6.39	0.96	11.47	4.09	2.93	0.13	6.39	0.94	11	4.72
p608	100	180	2	0.13	6.25	0.8	11.84	7.53	2.77	0.16	6.6	0.96	11.84	6.94
p609	100	180	0.78	0.1	6.55	0.77	10.27	5.19	1.86	0.14	5.37	0.87	9.94	3.69
p610	100	180	1.64	0.13	6.05	0.73	12.07	4.61	1.96	0.13	6.43	1.04	13.97	7.31
p611	100	180	2.71	0.13	8.27	0.88	12.97	5.95	1.03	0.13	7.49	1.26	15.29	6.77
p612	100	180	1.22	0.17	5.66	1.14	10.91	4.87	1.86	0.14	6.7	1.07	13.67	8.36
p613	200	370	1.89	0.72	6.61	10.26	11.87	128.37	2.48	0.76	7.23	13.66	11.49	206.07
p614	200	370	2.94	0.79	8.36	16.63	12.67	246.14	3.11	0.69	7.8	15.83	11.55	243.39
p615	200	370	3.35	0.81	8.94	10.33	14.1	197.91	2.79	0.94	8.75	19.36	13.94	254.95
p616	200	370	2.57	0.83	6.83	14.13	13.28	176.45	3.21	1.06	6.72	15.8	12.43	247.41

Table 6: Computational results for test set **MC** and **C**, where E_ℓ is randomly constructed.

			$\rho = 0.1$			$\rho = 0.2$			$\rho = 0.3$		
Ins.	$\lvert V\rvert$	$\lvert E\rvert$	IG(%)	Time	OG(%)	IG(%)	Time	OG(%)	IG(%)	Time	OG(%)
Test set **MC**											
mc7	400	760	5.36	24.05		15.61	1447.49				18.4^{10}
mc8	400	760	4.24	8.58		11.26	399.1				9.19^{10}
mc11	400	760	8.08	9.52		16.05	445.94				9.361^{10}
Test set **C**											
c01	500	625	1.66	2.87		3.8	105.74		5.87	1798.09	0.84^{2}
c02	500	625	1.21	2.48		3.72	123.65		6.28	1843.43	1.24^{2}
c03	500	625	1.28	3.45		3.88	116.02		5.99	1797.5	1.42^{4}
c04	500	625	1.42	3.24		3.94	229.34		5.51	1745.78	1.47^{9}
c05	500	625	1.2	2.52		3.66	68.66		5.59	2009.19	0.76^{6}
c06	500	1000	5.89	422.32				15.8^{10}			38.27^{10}
c07	500	1000	5.37	275.7				14.71^{10}			38.7^{10}
c08	500	1000	5.92	510.57				13.87^{10}			37.24^{10}
c09	500	1000	6.29	534.84				15.73^{10}			38.42^{10}
c10	500	1000	6.58	480.06				15.06^{10}			38.73^{10}
c11	500	2500			40.35^{10}	–	–	–	–	–	–
c12	500	2500			39.91^{10}	–	–	–	–	–	–

In column "OG(%)", the superscripts over the optimality gap values denote the number of instances not solved to optimality within the time limit.

"–" means the model size is too large so that the solver can not initialize the MIP formulation.

Table 7: Computational results for test set **MC** and **C**, where E_ℓ is randomly constructed with a degree constraint.

			$\rho = 0.1$			$\rho = 0.2$			$\rho = 0.3$		
Ins.	$\lvert V\rvert$	$\lvert E\rvert$	IG(%)	Time	OG(%)	IG(%)	Time	OG(%)	IG(%)	Time	OG(%)
Test set **MC**											
mc7	400	760	5.68	18.43		16.74	2270.93				18.09^{10}
mc8	400	760	5.11	7.44		9.51	469.18				16.77^{10}
mc11	400	760	8.18	11.1		16.48	491.38				9.68^{10}
Test set **C**											
c01	500	625	1.54	3.32		3.39	137.91		6.36	1810.64	1.31^{6}
c02	500	625	1.17	2.67		3.91	87.3		6.82	1923.73	0.94^{7}
c03	500	625	1.11	2.57		3.81	132.65		4.99	1095.91	1.71^{7}
c04	500	625	1.44	3.95		5.1	354.32		6.32	791.61	2.46^{9}
c05	500	625	1.5	3.07		3.92	242.13		5.46	3262.58	1.26^{8}
c06	500	1000	6.4	839	0.65^{1}			18.47^{10}			38.94^{10}
c07	500	1000	6	688.86				15.82^{10}			39.02^{10}
c08	500	1000	6.01	894.6				16.13^{10}			37.14^{10}
c09	500	1000	6.86	959.19				19.87^{10}			38.44^{10}
c10	500	1000	7.48	1112.95	0.23^{2}			17.36^{10}			39.84^{10}
c11	500	2500			35.36^{10}	–	–	–	–	–	–
c12	500	2500			39.22^{10}	–	–	–	–	–	–

In column “OG(%)”, the superscripts over the optimality gap values denote the number of instances not solved to optimality within the time limit.

“–” means the model size is too large so that the solver can not initialize the MIP formulation.

3.0 Mixed Integer Bilevel Optimization with k-optimal Follower: A Hierarchy of Bounds

3.1 Motivation

In this chapter, we focus on a broad class of mixed integer bilevel linear optimization problems (MIBLP) in which the follower's decision variables are all binary. In particular, the considered class of problems is formally stated as:

$$
\begin{aligned}
\eta^* = \max_{x,y} \;& \alpha^1 x + \alpha^2 y \\
\text{s.t.} \;& x \in \mathcal{X}, \\
& y \in \arg\max_{\bar{y}} \{\beta \bar{y} : \; Ax + G\bar{y} \leq d, \bar{y} \in \{0,1\}^n\},
\end{aligned} \tag{BP}
$$

where $\mathcal{X} = \mathcal{P} \cap (\mathbb{Q}_+^{p_1} \times \mathbb{Z}_+^{p_2})$ for a given polyhedron $\mathcal{P}$; $A \in \mathbb{Z}^{m \times (p_1+p_2)}, G \in \mathbb{Z}^{m \times n}$, $d \in \mathbb{Z}^m$; and $\alpha^1 \in \mathbb{Q}^{(p_1+p_2)}$, $\alpha^2 \in \mathbb{Q}^n$, and $\beta \in \mathbb{Q}_+^n$ are given row vectors. Let

$$\mathcal{S} = \{(x,y) \in \mathcal{X} \times \{0,1\}^n : \; Ax + Gy \leq d\}$$

be the feasible region of the relaxation obtained by dropping the optimality conditions on the follower's decision. We refer to x and y as the upper-level (leader's) variables and the lower-level (follower's) variables, respectively. Given $x \in \mathcal{X}$, we refer to

$$\mathcal{S}(x) = \{y \in \{0,1\}^n : \; (x,y) \in \mathcal{S}\} \quad \text{and} \quad \mathcal{R}(x) = \{y \in \mathcal{S}(x) : \; \beta y \geq \beta \bar{y} \quad \forall \bar{y} \in \mathcal{S}(x)\},$$

as the follower's feasible region and the follower's (rational) reaction set, respectively. The feasible region $\mathcal{S}(x)$ represents the options available to the follower, given a fixed solution x chosen by the leader, while the reaction set $\mathcal{R}(x)$ is the subset of the feasible region containing the feasible solutions that maximize the follower's objection function.

Note that (BP) specifies the optimistic case of the bilevel problem (BP). That is, there is an implicit assumption that the follower selects the most favorable solution for the leader from their reaction set $\mathcal{R}(x)$, i.e., if (x,y) is optimal for (BP), then we must have $y \in \mathcal{R}(x)$ and $\alpha^2 y \geq \alpha^2 \bar{y}$ for all $\bar{y} \in \mathcal{R}(x)$.

If we relax the optimality requirement on the follower's solution, the resulting *single-level relaxation* of (BP) is the mixed integer linear optimization problem (MILP)

$$\eta^{\text{SLR}} = \max_{(x,y)\in\mathcal{S}} \{\alpha^1 x + \alpha^2 y\}, \tag{SLR}$$

with feasible region $\mathcal{S}$. Problem (SLR) is also known as the *high-point problem* in the related literature; see, e.g., [85].

Observe that a feasible solution for (BP) can be obtained by fixing the leader's decision to an optimal solution obtained from (SLR). Specifically, assume (x^0, y^0) is optimal for (SLR). Then if $\hat{y}^0 \in \mathcal{R}(x^0)$ is an optimal (optimistic) follower's decision for the lower-level problem corresponding to x^0, then clearly, $(x^0, \hat{y}^0)$ is a feasible solution for (BP). Denote by $\hat{\eta}^{\text{SLR}} := \alpha^1 x_0 + \alpha^2 \hat{y}_0$ the leader's objective function value corresponding to $(x_0, \hat{y}_0)$. Thus, we have that:

$$\hat{\eta}^{\text{SLR}} \leq \eta^* \leq \eta^{\text{SLR}}.$$

The majority of branch-and-bound and branch-and-cut approaches for MIBLPs in the literature, see, e.g., [77, 85, 104, 109], solve either (SLR) or the linear programming relaxation of (SLR) to obtain the initial lower and upper bounds. We refer to the bounds obtained by this single-level relaxation-based approach as the SLR-based bounds.

Throughout the chapter, we make the following assumptions, which are standard in the bilevel optimization literature:

Assumption 3.1. *All entries in A, G and d are integers; $\beta \geq 0$.*

Assumption 3.2. *$\mathcal{P}$ is compact.*

As for Assumption 3.1, the components of A, G, and d can always be scaled to be integral as long as they are rational; this assumption is often exploited in the literature on MILPs, see, e.g., [37]. As for the second part of Assumption 3.1, this is without loss of generality, since whenever $\beta_j < 0$ for some $j \in [n] := \{1, 2, \ldots, n\}$, we can simply replace $\bar{y}_j$ by $1 - \bar{y}_j$, and then reduce the problem into an equivalent one with this assumption satisfied. As for Assumptions 3.2, it is common in the bilevel optimization literature to assume the compactness of the leader's feasible region, see, e.g., [36, 100, 104] and the references therein.

We also make a technical assumption involving the set of upper-level variables that we refer to as the *linking variables*, those with at least one non-zero coefficient in the lower-level constraint matrix. The linking variables are formally defined as those with indices in the set

$$L = \left\{ i \in \{1, \ldots, p_1 + p_2\} : A^i \neq 0 \right\},$$

where A^i denotes the i^{th} column of A. We make the following assumption to ensure that optimal solutions of (BP) exist [108]:

Assumption 3.3. *All linking variables are integer variables.*

For any positive integer n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. We denote by $e^j \in \mathbb{R}^n$ the j^{th} unit vector; by $\mathbf{0}$ the vector with all components equal to 0 and $\mathbf{1}$ the vector with all components equal to 1; and finally by M^i and m^i the i^{th} column and row of matrix M, respectively. All vectors are column vectors by default, with the exception of vectors used exclusively as objective function vectors (and the rows of matrices), which are taken to be row vectors for notational simplicity.

Related Work. In general, solving MIBLPs is quite challenging. The initial bounds used within branch-and-bound and branch-and-cut frameworks are of critical importance for the overall performance of this type of exact methods. Unfortunately, for general classes of bilevel problems there are no theoretical guarantees on the quality of the SLR-based bounds. Moreover, the computational experiments available in the literature also indicate that (SLR) typically yields relatively poor bounds; see, e.g., [23]. Consequently, exact solution of large-scale bilevel optimization problems remains an intractable task, in particular, if one compares available bilevel solvers to the state-of-the-art commercial solvers for MILPs, e.g., CPLEX [51]. The latter is capable of handling millions of variables and constraints for many broad classes of MILPs. On the other hand, the most recent version of MibS can typically solve medium-sized problems with only up to several hundred integer decision variables at the lower level.

There are only a few studies, mostly focused on special classes of bilevel problem, that use bounding methods that are not SLR-based. In particular, the studies in [23, 105] describe rather effective continuous relaxation based bounds that provide a lower bound for the mixed integer max-min optimization problems, where the objective functions of the decision-makers have the same functional form but have opposite signs. Specifically, these approaches relax the integrality constraints of the follower and then reformulate the resulting bilevel linear optimization problem as a MILP through optimality conditions. In [29], the knapsack interdiction problem is studied and the concept of the critical items in the follower's knapsack problem is explored to further improve the bounds of the continuous relaxation. However, the aforementioned types of bounding methods exploit specific structure of the considered classes of bilevel problems and are not applicable for more general MIBLPs.

Overview. The remainder of the chapter is organized as follows. Section 3.2 provides a formal description of our proposed framework and summarizes our contributions. Section 3.3 develops the hierarchy of lower and upper bounds for the considered class of MIBLPs. Then, in Section 3.4, we describe two single-level MILP reformulations for our bilevel problems. Section 3.5 considers a class of bilevel problems for which the follower's local optimality implies global optimality and hence, the follower can be viewed as rational, despite using a solution methodology typically employed as a heuristic. Finally, extensively computational experiments are conducted to illustrate the effectiveness of our proposed framework in Section 3.6.

3.2 Bilevel Optimization With k-optimal Follower

The main idea in the remainder of the chapter is to consider an optimality-based relaxation of (BP) in which the follower's response is required only to be a locally optimal solution to the parametric follower's problem. Formally, given a positive integer k, in response to the leader's decision x, the follower chooses a solution in the k-optimal reaction set, defined as follows.

Definition 3.1. *The* k-swap neighborhood *of* $y \in \{0,1\}^n$ *is the set*

$$\mathcal{N}_k(y) = \{\bar{y} \in \{0,1\}^n : \|\bar{y} - y\|_1 \leq k\} \qquad (\mathcal{N}_k)$$

of all vectors within Hamming distance k of y, where the Hamming distance [49] between two binary vectors y and $\bar{y}$ is the number of positions at which these vectors are different, i.e., $\|\bar{y} - y\|_1$.

Definition 3.2. *The* k-optimal reaction set *with respect to $x \in \mathcal{X}$ is the set*

$$\mathcal{R}_k(x) = \{y \in \mathcal{S}(x) : \beta y \geq \beta \bar{y} \quad \forall\, \bar{y} \in \mathcal{N}_k(y) \cap \mathcal{S}(x)\} \tag{$\mathcal{R}_k$}$$

of all k-optimal solutions to the lower-level problem with respect to $x \in \mathcal{X}$.

The concept of a k-optimal solution in Definition 3.2 is commonly used in the literature on combinatorial optimization in methods that exploit local optimality as a way of generating heuristic solutions; see examples in the context of the traveling salesman [1, 25, 48], routing and scheduling [17, 91, 94], as well as many other combinatorial optimization problems [56, 59, 82, 95].

The mixed integer bilevel linear optimization problem with k-optimal follower is then formally stated as follows:

$$\begin{aligned} \eta_k^* = \max_{x,y}\ & \alpha^1 x + \alpha^2 y \\ \text{s.t.}\ \ & x \in \mathcal{X}, \\ & y \in \mathcal{R}_k(x). \end{aligned} \tag{BP$_k$}$$

We say that (x, y) is feasible for (BP$_k$) if $x \in \mathcal{X}$ and $y \in \mathcal{R}_k(x)$. Note that, as with (BP), the formulation of (BP$_k$) implicitly assumes the optimistic case, although the approach can also be generalized to the pessimistic case (there is a brief discussion on this issue in Section 3.7).

Observe that (x, y) is a feasible solution for (BP$_0$) if and only if $(x, y) \in \mathcal{S}$. Hence, $\mathcal{R}_0(x) = \mathcal{S}(x)$ and (BP$_0$) is equivalent to (SLR). Furthermore, $\mathcal{R}_n(x) = \mathcal{R}(x)$, so that (BP$_n$) is equivalent to (BP). We show in Section 3.3 that for any other $k \in \{0, \ldots, n\}$, the optimal objective function value of (BP$_k$) provides an upper bound on the optimal objective function value of (BP); furthermore, the optimal objective function value of (BP$_k$) is monotonic in k. Optimal solutions to (BP$_k$) can also be used to derive a hierarchy of monotonically increasing lower bounds for (BP).

To solve (BP_k), in Section 3.4 we propose two single-level MILP formulations solvable by standard MILP solvers. The first one follows a disjunctive-based approach with additional logical variables. The second formulation extends the previous idea and exploits the inherent structure of (BP_k) through the lens of mixing-set inequalities [9, 46, 130]. Our extensive numerical study with these MILP formulations indicates that the developed bounds are substantially better than the SLR-based bounds. Furthermore, the bounds converge to the optimal objective function value of (BP) for rather small values of k, and the required computational efforts is small. This observation suggests that the bounds provided by (BP_k) have tremendous potential for boosting the performance of exact solvers, especially for bilevel problems with low quality single-level relaxation bounds, which are common in practice.

Aside from the obvious usefulness of the bounds that can be derived from (BP_k), another important reason for studying (BP_k) is that it provides an exact reformulation for classes of MIBLPs for which k-optimal solutions (ideally, for some small fixed k) are also globally optimal for the lower-level problem. In Section 3.5, we exploit this idea by showing that 2-optimality of the lower-level problem implies global optimality for a general class of bilevel matroid problems. Specialized approaches for solving the bilevel minimum spanning tree problem (BMST) are then developed and tested in Section 3.6.3.

Finally, the proposed modeling framework (BP_k) provides a natural connection between theoretical exact formulations for hierarchical decision-making problems and the practical considerations arising in many real-life applications. In standard exact formulations of bilevel optimization problems, it is typically assumed that the follower is completely rational and their computational resources are sufficient to solve the lower-level problem to global optimality for any leader's decision. In many practical settings, it is clear that this is an unrealistic assumption. The follower often faces a situation in which either their computational resources are limited or they simply lack the knowledge to develop an efficient approach to obtaining the exact solution of their lower-level problem (this may be the case, in particular, when the follower's problem is NP-hard). Furthermore, in practice it is often the case that the follower only seeks a high-quality sub-optimal solution within reasonable time. To address such "inexact" followers, Smith et al. [101] and Zare et al. [127] study mxied integer bilevel optimization problems, where the reaction solutions by the follower are computed using a finite set of heuristic algorithms. In our framework, we do not specify the follower's strategies and algorithms, but instead quantify their possible reaction by exploiting the concept of locally optimal solutions. Thus, the proposed (BP_k) problems can be viewed as a more general modeling framework for hierarchical decision-making problems than the classical MIBLPs.

3.2.1 Characterization of k-optimal Reaction Set

Although the definition of the k-optimal reaction set given is already straightforward, we would ideally like a characterization that can be used to formulate (BP_k) as a mathematical optimization problem (preferably an MILP). To develop such a characterization, we first define the notion of an *improving k-swap.*

Definition 3.3. *A vector $w \in \{-1, 0, 1\}^n$ represents an* improving k-swap *if $\|w\|_1 = k$ and $\beta w > 0$. The set of all improving k-swaps is denoted by $\mathcal{T}_k$.*

In the above definition, the members of $\mathcal{T}_k$ represent ways of flipping the values of k variables in a given solution to get a new solution with improved objective function value. Note that membership in $\mathcal{T}_k$ only considers the number of flips and their effect on the objective function value, not the effect on feasibility of the lower-level problem, since the effect of the k-swap on feasibility would vary depending on the solution.

To illustrate, let $x \in \mathcal{X}$, $y \in \mathcal{S}(x)$, and $w \in \mathcal{T}_k$ be given. Applying the k-swap to y, we get $y + w$, which is an improved solution if and only if $y + w \in \mathcal{S}(x)$. Note that $y + w$ may be infeasible either because $G(y + w) \nleq d - Ax$ or because $y + w \notin \{0, 1\}^n$, e.g., $y_i = w_i = 1$.

The following necessary and sufficient conditions characterize membership in $\mathcal{R}_k(x)$ for $x \in \mathcal{X}$. Informally, the result says that y is k-optimal if and only if no improving j-swap is feasible for $j \in [k]$.

Proposition 3.1. *Let $(x, y) \in \mathcal{S}$. Then $y \in \mathcal{R}_k(x)$ if and only if $y + w \notin \mathcal{S}(x)$ for all $w \in \mathcal{T}^k$, where $\mathcal{T}^k = \cup_{j\in[k]}\mathcal{T}_j$.*

Proof. Let $k \in [n]$ and $x \in \mathcal{X}$ be given. There are two parts to the proof.

"$\Leftarrow$" We prove the contrapositive. Let $y \notin \mathcal{R}_k(x)$ given. Then there exists $\bar{y} \in \mathcal{N}_k(y) \cap \mathcal{S}(x)$ such that $\beta\bar{y} > \beta y$. Let $\bar{w} = \bar{y} - y$ and $j = \|w\|_1$. We have $j \in [k]$, $w \in \mathcal{T}_j$, and

$y + w \in \mathcal{S}(x)$, so the contrapositive is proven.

"$\Rightarrow$" We again prove the contrapositive. We therefore have $j \in [k]$ and $w \in \mathcal{T}_j$ such that $\bar{y} = y + w \in \mathcal{S}(x)$. Then $\bar{y} \in \mathcal{N}_k(y) \cap \mathcal{S}(x)$ and $\beta\bar{y} > \beta y$, so the contrapositive is proven.

■

We next illustrate Definition 3.3 and Proposition 3.1 as follows.

Example 3.1. *Consider the bilevel knapsack problem*

$$\begin{aligned}
\max_{x,y} \quad & -\beta y \\
s.t. \quad & \sum_{j=1}^{n} x_j \le b, x \in \{0,1\}^n, \\
& y \in \arg\max_{\bar{y} \in \{0,1\}^n} \{\beta\bar{y} : \ a\bar{y} \le C, x_j + \bar{y}_j \le 1 \ \forall j \in [n]\},
\end{aligned}$$

where $n = 6$, vector $\beta = (70, 40, 39, 37, 17, 15)$, weight vector $a = (28, 25, 20, 18, 13, 10)$, the leader's and the follower's knapsack capacities are given by $b = 1$ and $C = 30$, respectively. We note that

$$\min_x \max_y \ \beta y = -\max_x \Big(-\max_y \ \beta y \Big)$$

and thus, the considered example is essentially an instance of the knapsack interdiction problem [32].

For a fixed leader's decision $x_0 = (1, 0, 0, 0, 0, 0)^\top$, the lower-level problem becomes a knapsack problem given by

$$\begin{aligned}
\max_{y \in \{0,1\}^6} \quad & 70y_1 + 40y_2 + 39y_3 + 37y_4 + 17y_5 + 15y_6 \\
s.t. \quad & \qquad\quad 25y_2 + 20y_3 + 18y_4 + 13y_5 + 10y_6 \le 30, \\
& y_1 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad = 0.
\end{aligned}$$

First, consider the case of $k = 1$. From Definition 3.3, we have $\mathcal{T}_1 = \{w \in \{-1, 0, 1\}^n : \beta w > 0, \|w\|_1 = 1\}$. Since $\beta_j > 0$ for each j, it follows that

$$\mathcal{T}_1 = \{e^j : \ j \in [n]\}.$$

Furthermore, we have y is a 1-optimal solution for the follower if and only if it is maximal (i.e., $ay \leq C$ and $ay + a_j > C$ for any j such that $y_j = 0$). Therefore, it can be verified that in this example we have that

$$\mathcal{R}_1(x_0) = \{\{2\}, \{3, 6\}, \{4, 6\}, \{5, 6\}\}.$$

In the above, for simplicity we use subsets of the selected items in the follower's knapsack to describe $\mathcal{R}_1(x_0)$ instead of the corresponding binary vector y, e.g., set $\{2\}$ denotes the follower's solution $y = (0, 1, 0, 0, 0, 0)^\top$.

Next, it is also easy to verify that

$$\mathcal{T}_2 = \{e^i - e^j : \ i, j \in [n], \beta_i > \beta_j\} \cup \{e^i + e^j : i, j \in [n], i \neq j\}$$

and the 2-optimal reaction set of the follower is given by $\mathcal{R}_2(x_0) = \{\{2\}, \{3, 6\}\} \subseteq \mathcal{R}_1(x_0)$. Then we can compute

$$\begin{aligned}
\mathcal{T}_3 = &\{e^i + e^j + e^k : i, j, k \in [n], i \neq j \neq k\} \cup \\
&\{e^i + e^j - e^k : i, j, k \in [n], i \neq j \neq k, \beta_i + \beta_j > \beta_k\} \cup \\
&\{e^i - e^j - e^k : i, j, k \in [n], i \neq j \neq k, \beta_i > \beta_j + \beta_k\}.
\end{aligned}$$

Finally, we have that $\mathcal{R}_3(x_0) = \cdots = \mathcal{R}_n(x_0) = \{\{3, 6\}\}$. It implies that in this instance the k-optimal reaction sets of follower are monotone decreasing and converge to $\mathcal{R}(x_0)$ for $k = 3$.

3.2.2 Complexity of (BP_k)

The next question of interest is to explore the theoretical computational complexity of computing (BP_k) for a fixed value of k. It is easy to guess that since the problem of simply finding a feasible solution to a 0-1 integer optimization problem is already NP-hard, then (BP_k) should also be NP-hard. In fact, for any pure 0-1 integer optimization problem, one can easily construct a bilevel optimization problem with a lower-level problem in which a solution is optimal if and only if it is 1-optimal.

Theorem 3.1. *(BP_k) is NP-hard for any fixed integer $k \geq 1$.*

Proof. We show that pure binary integer optimization can be reduced to an instance of (BP_k). Let $\mathcal{X} \subseteq \{0,1\}^n$ be the feasible region of a pure binary integer optimization problem with objective function vector $\alpha \in \mathbb{R}^n$. Let an instance of (BP_k) be defined as follows. The set $\mathcal{X}$ is as given. We let $A = I_n$, $G = -I_n$, $d = 0$, $\beta_i = -1$ for $i \in [n]$, $\alpha^1 = \alpha$ and $\alpha^2 = 0$. Then the second-level problem is trivially solvable, since $\mathcal{R}_k(x) = \mathcal{R}(x) = \{x\}$ for all $x \in \mathcal{X} \subseteq \{0,1\}^n$. The solution to this instance of (BP_k) also solves

$$\max_{x \in \mathcal{X}} \alpha x$$

■

Remark 3.1. *It is evident that the proof did not depend at all on the lower-level problem. As long as $\alpha^2 = 0$, the solution to (BP_k) will be the same as that of the binary integer optimization problem. A stronger result also holds—that the decision version of (BP_k) is NP-complete for fixed k—as we show later.*

Remark 3.2. *The MIBLP is known to be hard for complexity class Σ_2^p in general [54, 75]. However, in Section 3.4, we demonstrate a polynomial-time procedure to reduce (BP_k) to a single-level linear MILP of polynomial size for a fixed value of k. Thus, for any fixed k, the decision version of (BP_k) is in class NP.*

3.3 Hierarchy of Bounds

In this section, we formally describe the hierarchy of upper and lower bounds associated with (BP_k). Some basic properties of these bounds are also established.

3.3.1 Upper Bounds

We first show that (BP_k) provides a natural hierarchy of upper bounds for (BP). In particular, these upper bounds can be shown to be progressively tighter with increasing k. Formally:

Theorem 3.2. $\eta^{SLR} = \eta_0^* \geq \eta_1^* \geq \eta_2^* \geq \cdots \geq \eta_n^* = \eta^*$.

Proof. Recall that η_k^* is computed as

$$\eta_k^* = \max \{\alpha^1 x + \alpha^2 y : x \in \mathcal{X}, y \in \mathcal{R}_k(x)\}.$$

To prove $\eta_k^* \geq \eta_{k+1}^*$ for $k = 0, 1, \ldots, n-1$, it is sufficient to observe from Proposition 3.1 that $\mathcal{R}_k(x) \supseteq \mathcal{R}_{k+1}(x)$. If $k = 0$, then $\mathcal{N}_0(y) = \{y\}$ and $\mathcal{R}_0(x) = \mathcal{S}(x)$, which implies that $\eta^{\mathrm{SLR}} = \eta_0^*$. If $k = n$, then $\mathcal{N}_n(y) = \{0,1\}^n$ and $\mathcal{R}_n(x) = \mathcal{R}(x)$, which implies that $\eta_n^* = \eta^*$. ■

Our computational study in Section 3.6 indicates that the upper bound η_k^* is substantially tighter than the SLR-based bound, η_0^*, even for $k = 1$, and the optimal objective function value of (BP_k) converges to that of (BP) rather fast. The following example is provided as an illustration.

Example 1 (continued). *Observe that in Example 3.1, $\eta_0^* = 0$ with optimal solution $(x^0, y^0) = (\mathbf{0}, \mathbf{0})$. For $k = 1$, we can verify that the optimal leader's decision for* (BP_1) *is $\{6\}$ and the follower's 1-optimal solution is $\{5\}$, resulting in $\eta_1^* = -17$. For $k = 2$, the leader selects $\{1\}$, with the follower's 2-optimum given by $\{2\}$; thus, $\eta_2^* = -40$. For $k = 3$, the leader's optimal decision is $\{1\}$ and the follower chooses $\{3, 6\}$ with $\eta_3^* = -54$. We can further verify that $\eta_4^* = \cdots = \eta_6^* = -54$ with the leader's and the follower's optimal decisions given by $\{1\}$ and $\{3, 6\}$, respectively.*

In the above example in order to obtain an optimal solution for (BP) it is sufficient to solve (BP_3). Thus, a natural question is to establish when an optimal solution for (BP_k) is also optimal for (BP). The following results provide possible answers for this question.

Proposition 3.2. *Given an optimal solution (x^k, y^k) for (BP_k), if $y^k \in \mathcal{R}(x^k)$ (i.e., y^k is also a globally optimal solution for the lower-level problem with respect to the leader's decision x^k), then (x^k, y^k) is also optimal for (BP) and $\eta^*_k = \eta^*_{k+1} = \cdots = \eta^*_n = \eta^*$.*

Proof. Since $y^k \in \mathcal{R}(x^k)$, we have that (x^k, y^k) is a bilevel feasible solution. It follows that $\eta^* \geq \alpha x^k + \beta y^k = \eta^*_k$ as (x^k, y^k) is optimal for (BP_k). Therefore, based on Theorem 3.2, we have $\eta^*_k = \eta^*_{k+1} = \cdots = \eta^*_n$. ■

Note that Proposition 3.2 provides a practical approach to verify whether $\eta^*_k = \eta^*$. Next, we discuss a more general condition.

Proposition 3.3. *If any k-optimal solution of the lower-level problem is also a globally optimal solution for the follower, then $\eta^*_k = \eta^*_{k+1} = \cdots = \eta^*_n$.*

Proof. It follows directly from the observation that $\mathcal{R}_k(x) = \mathcal{R}_{k+1}(x) = \cdots = \mathcal{R}_n(x)$. ■

For some single-level combinatorial optimization problems, locally optimal solutions are also globally optimal for reasonably small values of k, e.g., the minimum spanning tree problem with $k = 2$. Hence, Proposition 3.3 provides us with one possible approach for treating bilevel generalizations of such problems. In Section 3.5, we use Proposition 3.3 in the context of a general class of bilevel matroid problems.

On the other hand, it also may occur that the upper bounds provided by (BP_k), $k \geq 1$, does not improve the SLR-based bound, η^*_0. The next example illustrates this situation.

Example 3.2. *Consider an instance of the bilevel knapsack problem:*

$$\begin{aligned}
&\max_{x,y} \alpha(x + y) \\
&\; s.t. \; \sum_{j=1}^{n} x_j \leq 1, x_1 = 0, x \in \{0,1\}^n, \\
&\qquad y \in \arg \max_{\bar{y} \in \{0,1\}^n} \{\sum_{j=1}^{n} \beta \bar{y} : ax + a\bar{y} \leq C, x_j + \bar{y}_j \leq 1 \; \forall j \in [n]\},
\end{aligned}$$

where $n = 4$, *vector* $\alpha = (15, 7, 6, 5)$, *vector* $\beta = (10, 9, 7, 4)$, *weight vector* $a = (10, 6, 4, 3)$, *and* $C = 10$. *We can verify that* (SLR), (BP_1) *and* (BP_2) *lead to the same optimal objective function value of* 15 *with the leader's decision* $\emptyset$ *and the follower's decision* $\{1\}$. *That is,* $\eta_0^* = \eta_1^* = \eta_2^* = 15$. *On the other hand,* $\eta^* = \eta_3^* = 13$ *for* (BP_3) *with the leader's decision* $\emptyset$ *and the follower's decision* $\{2, 3\}$.

Finally, this example can be generalized via the following analytical result.

Proposition 3.4. *If* $\alpha^2 w > 0$ *for any* $w \in \mathcal{T}^k$ *and* $1 \leq k \leq n$, *then* $\eta_0^* = \eta_1^* = \cdots = \eta_k^*$.

Proof. From Theorem 3.2, we have $\eta_0^* \geq \cdots \geq \eta_{k-1}^* \geq \eta_k^*$. Thus, we only need to prove that $\eta_0^* \leq \eta_k^*$. Suppose (x^0, y^0) is an optimal solution of (BP_0). Then it suffices to show that (x^0, y^0) is also a feasible solution for (BP_k), that is $y^0 \in \mathcal{R}_k(x^0)$. We first note that $y^0 \in \mathcal{S}(x^0)$ and $\alpha^2 y^0 \geq \alpha^2 y$ for any $y \in \mathcal{S}(x^0)$.

Suppose $y^0 \notin \mathcal{R}_k(x^0)$, then based on Proposition 3.1, there exists $w \in \mathcal{T}^k$ such that $y^0 + w \in \{0, 1\}^n$ and $y^0 + w \in \mathcal{S}(x^0)$. Since $\alpha^2 w > 0$, then $\alpha^2(y + w) > \alpha^2 y$, which contradicts with the assumption that (x^0, y^0) is optimal for (BP_0). Hence, $y \in \mathcal{R}_k(x^0)$ and the result follows. ■

Corollary 3.1. *If* $\alpha_j^2 > 0$ *for all* $j \in [n]$ *such that* $\beta_j > 0$, *then* $\eta_0^* = \eta_1^*$.

3.3.2 Lower Bounds

As outlined in Section 3.2, (BP_k) can also be exploited to construct lower bounds for (BP). Formally, let (x^k, y^k) be an optimal solution for (BP_k). Denote by $\hat{y}^k$ the follower's optimal solution that corresponds to the leader's decision x^k, i.e., $\hat{y}^k \in \mathcal{R}(x^k)$. Clearly, a pair $(x^k, \hat{y}^k)$ forms a bilevel feasible solution for (BP). Then we define:

$$\hat{\eta}_k = \alpha^1 x^k + \alpha^2 \hat{y}^k,$$

which provides a valid lower bound for η^*.

However, it can be verified (see an example below) that $\hat{\eta}_k$ is not necessarily monotonic in k. To present a hierarchy of monotonically increasing lower bounds for (BP), we need to slightly modify the definition of $\hat{\eta}_k$ as follows:

Definition 3.4. *Given an integer $k, 1 \le k \le n$, the modified $\hat{\eta}_k$, denoted as $\hat{\eta}'_k$, is given by*

$$\hat{\eta}'_k = \max_{t=0,1,\ldots,k} \{\alpha^1 x^t + \alpha^2 \hat{y}^t\}.$$

Then we obtain the following hierarchy of lower bounds:

Theorem 3.3. *$\hat{\eta}_k \le \hat{\eta}'_k$ for $1 \le k \le n$, and $\hat{\eta}^{SLR} = \hat{\eta}_0 \le \hat{\eta}'_1 \le \hat{\eta}'_2 \le \cdots \le \hat{\eta}'_n = \eta^*$.*

Proof. Note that $\hat{\eta}_k = \alpha^1 x^k + \alpha^2 \hat{y}^k \le \eta^*$. Thus, based on Definition 3.4 we have $\hat{\eta}_k \le \hat{\eta}'_k = \max_{t=0,1\ldots,k}\{\hat{\eta}_k\} \le \eta^*$. Also, it directly follows from Definition 3.4 that $\hat{\eta}'_k \le \hat{\eta}'_{k+1}$ for $0 \le k \le n-1$. If $k = n$, then $\mathcal{R}_n(x) = \mathcal{R}(x)$, which yields that $\eta^* = \eta^*_n = \hat{\eta}_n \le \hat{\eta}'_n \le \eta^*$, and the result follows. ∎

We next illustrate the considered lower bounds in the following example.

Example 3.3. *Consider an instance of the knapsack interdiction problem:*

$$\begin{aligned} \max_{x \in \{0,1\}^6} \Big(- \max_{y \in \{0,1\}^6} \;& 11y_1 + 2y_2 + 7y_3 + 8y_4 + 3y_5 + 10y_6 \Big) \\ s.t. \;& \sum_{j=1}^{6} x_j \le 1,\; x_j + y_j \le 1 \;\forall j \in [6], \\ & 14y_1 + 12y_2 + 6y_3 + 5y_4 + 3y_5 + 2y_6 \le 14. \end{aligned}$$

Then we compute that $\eta^*_0 = 0$ with the leader's decision $\mathbf{0}$, which leads to $\hat{\eta}_0 = \hat{\eta}'_0 = -25$ with follower's decision $\{4,5,6\}$. For $k = 1$, the leader's optimal decision set is $\{6\}$ with $\eta^*_1 = -2$. We can compute $\hat{\eta}_1 = \hat{\eta}'_1 = -18$ with follower's decision $\{3,4,5\}$. For $k = 2$, we see that $\mathbf{0}$ is optimal for the leader with $\eta^*_2 = -11 < \eta^*_1$, while its corresponding lower bound $\hat{\eta}_2 = -25 < \hat{\eta}_1$ with the follower's decision $\{3,4,6\}$. On the contrary, our modified lower bound $\hat{\eta}'_2 = \max_{k=0,1,2}\{\hat{\eta}_k\} = -18 \ge \hat{\eta}'_1$. We can further verify that $\hat{'}\eta_3 = \cdots = \hat{'}\eta_6 = \eta^* = -18$.

3.4 Extended Formulations

In this section, we present two extended formulations based on the concepts presented so far. In Section 3.4.1, we begin by describing a formulation of (BP_1), i.e., the case when $k = 1$, using a disjunctive-based approach. We then generalize this formulation to general k in Section 3.4.2. In doing so, we show that (BP_k) is polynomially reducible to a single-level MILP for any fixed k. Additionally, several preprocessing procedures are considered to reduce the number of variables and constraints in the proposed MILP formulations. Finally, by looking carefully into the structure of (BP_k), we identify, somewhat surprisingly, a mixing-set substructure within (BP_k), which is exploited to develop a tighter extended MILP formulation in Section 3.4.3.

3.4.1 Formulation for (BP_1)

For $k = 1$, Definitions 3.1 and 3.2 imply that the the follower's 1-optimal reaction set is defined as:

$$\mathcal{R}_1(x) = \{y \in \mathcal{S}(x) : \beta y \geq \beta \bar{y} \quad \forall\, \bar{y} \in \mathcal{N}_1(y) \cap \mathcal{S}(x)\},$$

where $\mathcal{N}_1(y) = \{\bar{y} \in \{0,1\}^n : ||y - \bar{y}||_1 \leq 1\}$. We have $\mathcal{T}_1 = \{e^j : \ \beta_j > 0\}$. Then Proposition 3.1 can be used to provide a simplified condition to determine whether $y \in \mathcal{R}_1(x)$ as follows:

Proposition 3.5. *Let $(x, y) \in \mathcal{S}$. Then $y \in \mathcal{R}_1(x)$ if and only if either $y_j = 1$, or $y_j = 0$ and $y + e^j \notin \mathcal{S}(x)$ for $j \in J^+_\beta$, where $J^+_\beta = \{j \in [n] : \beta_j > 0\}$.*

Observe that for any $y \in \{0,1\}^n$, if $y \notin \mathcal{S}(x)$, then based on Assumption 3.1, there must exist some row $i \in [m]$ such that

$$a^i x + g^i y \geq d_i + 1,$$

which (recalling that a^i and g^i are the i^{th} rows of matrices A and G, respectively) yields that

$$\Big\{y \in \{0,1\}^n : y \notin \mathcal{S}(x)\Big\} = \bigcup_{i=1}^{m} \Big\{y \in \{0,1\}^n : g^i y \geq d_i + 1 - a^i x\Big\}.$$

To develop a formulation that reflects the conditions in Proposition 3.5, we introduce binary variables z_{ij} for $i \in [m]$, $j \in J^+_\beta$, such that when $z_{ij} = 1$, we must have $a^i x + g^i(y + e^j) \geq d_i + 1$, i.e., $z_{ij} = 1 \Rightarrow y + e^j \notin \mathcal{S}(x)$. Using this set of auxiliary binary variables, we reformulate (BP_1) as a linear MILP in the following form:

$$\eta_1^* = \max_{x,y,z} \alpha^1 x + \alpha^2 y \tag{BP$_1$-DF-a}$$

$$\text{s.t.} \quad (x, y) \in \mathcal{S}, \tag{BP$_1$-DF-b}$$

$$a^i x + g^i y + z_{ij}(\mu_i - h_{ij}) \geq \mu_i \quad \forall i \in [m], j \in J_\beta^+, \tag{BP$_1$-DF-c}$$

$$\sum_{i=1}^{m} z_{ij} + y_j \geq 1 \quad \forall j \in J_\beta^+, \tag{BP$_1$-DF-d}$$

$$z_{ij} \in \{0, 1\} \quad \forall i \in [m], j \in J_\beta^+, \tag{BP$_1$-DF-e}$$

where $h_{ij} = d_i + 1 - g_j^i$, and μ_i is a sufficiently small constant parameter (below we provide some additional discussion on its appropriate values).

To demonstrate the correctness of the obtained formulation, we make the following observations. First, note that constraint (BP_1-DF-c) ensures that if $z_{ij} = 1$, then we must have $a^i x + g^i(y + e^j) \geq d_i + 1$; if $z_{ij} = 0$, then constraint (BP_1-DF-c) is always satisfied for appropriate values of μ_i. Constraint (BP_1-DF-d) guarantees that if $y_j = 0$ for some $j \in J_\beta^+$, then there exists $i \in [m]$ such that $z_{ij} = 1$. This, in turn, ensures that $y + e^j \notin \mathcal{S}(x)$. On the other hand, since $y_j = 1$ already implies $y + e^j \notin \mathcal{S}(x)$, we do not need to ensure violation of any constraints in this case.

For a constraint (BP_1-DF-c) to be redundant when $z_{ij} = 0$, it is sufficient to have

$$\mu_i \leq \min_{(x,y) \in \mathcal{S}} a^i x + g^i y.$$

Based on Assumption 3.2, there exist vectors l and u such that for any $x \in \mathcal{X}$ we have $l \leq x \leq u$, which leads to a straightforward choice of μ_i as

$$\mu_i = \sum_{j: a_j^i > 0} a_j^i l_j + \sum_{j: a_j^i < 0} a_j^i u_j + \sum_{j: g_j^i < 0} g_j^i.$$

If all entries in A and G are non-negative, then μ_i can be trivially set to 0. Note that the chosen value of μ_i may influence the quality of the above MILP reformulation (BP_1-DF). A tighter μ_i can be achieved for bilevel problems with identifiable structures; this issue is further addressed in Section 3.6.

Next, we discuss how to reduce the number of variables and constraints in the formulation (BP_1-DF) through some simple preprocessing steps.

Proposition 3.6. *(i) If $g^i_j \leq 0$ for some $i \in [m]$ and $j \in J^+_\beta$, then $z_{ij} = 0$ for any feasible solution of (BP_1-DF).*

(ii) If $\alpha^2_j > 0$ and $g^i_j \geq 0$ for some $j \in J^+_\beta$ and all $i \in [m]$, then removing variables z_{ij} and the associated constraints for all $i \in [m]$ in (BP_1-DF) does not change its optimal objective function value.

Proof. (i) Suppose (x, y, z) is a feasible solution of (BP_1-DF). Since $(x, y) \in \mathcal{S}$, then $a^i x + g^i y \leq d_i$ for all $i \in [m]$. If $z_{ij} = 1$ for some $i \in [m]$ and $j \in J^+_\beta$ such that $g^i_j \leq 0$, then from constraint (BP_1-DF-c), we have

$$a^i x + g^i y \geq d_i + 1 - g^i_j \geq d_i + 1,$$

where the second inequality follows from the assumption that $g^i_j \leq 0$. Thus, we obtain a contradiction and it immediately follows that z_{ij} should be equal to 0 in any feasible solution.

(ii) We refer to (BP'_1-DF) as the problem obtained by removing z_{ij_0} for some $j_0 \in J^+_\beta$ and all $i \in [m]$ such that $\alpha^2_{j_0} > 0$ and $g^i_{j_0} \geq 0$. Denote by η'_1 the resulting optimal objective function value of (BP'_1-DF). It is clear that $\eta'_1 \geq \eta^*_1$.

Assume (x', y', z') is an optimal solution of (BP'_1-DF). Next, it is sufficient to show that $y' \in \mathcal{R}_1(x')$ as the latter implies that $\eta'_1 \leq \eta^*_1$. By Proposition 3.5 (recall that z_{ij} is not removed from (BP_1-DF) for any $j \neq j_0$) we have that $y' + e^j \notin \mathcal{S}(x')$ for any $j \in J^+_\beta$ and $j \neq j_0$. Therefore, we only need to show $y' + e^{j_0} \notin \mathcal{S}(x')$.

If $y'_{j_0} = 1$, then the statement holds trivially. If $y'_{j_0} = 0$, then suppose $y' + e^{j_0} \in \mathcal{S}(x)$. Consider $j \in J^+_\beta$ such that $j \neq j_0$ and $y'_j = 0$. Since $y' + e^j \notin \mathcal{S}(x')$, assume that $a^i x + g^i(y' + e^j) \geq d_i + 1$ for some $i \in [m]$ (i.e., $z_{ij} = 1$ in constraint (BP_1-DF-c)). It immediately follows that $a^i x + g^i(y' + e^j + e^{j_0}) \geq d_i + 1$ as $g^i_{j_0} \geq 0$. Therefore, we have $y' + e^{j_0} + e^j \notin \mathcal{S}(x')$ for any $j \in J^+_\beta$. Hence, based on Proposition 3.5, $y' + e^{j_0}$ is a feasible solution for (BP_1-DF). However, $\eta^*_1 \geq \alpha^1 x' + \alpha^2(y' + e^{j_0}) = \alpha^1 x' + \alpha^2 y' + \alpha^2_{j_0} > \alpha^1 x' + \alpha^2 y' = \eta'_1$, which contradicts with the fact that $\eta^*_1 \leq \eta'_1$. Thus, $y' + e^{j_0} \notin \mathcal{S}(x')$ and the result follows.

Note that (i) simply follows from the fact that given any $y \in \mathcal{S}(x)$, if $y_j = 0$, then $a^i x + g^i(y + e^j) \leq d$, as $g^i_j \leq 0$.

■

Remark 3.3. *Note that if z_{ij} can be fixed to zero, then the corresponding constraint in* (BP$_1$-DF-c) *can also be removed.*

Remark 3.4. *Observe that the statement in Proposition 3.6 (ii) is consistent with Corollary 3.1. In particular, if $\alpha^2_j > 0$ for all $j \in J^+_\beta$ (as in Corollary 3.1), then in fact, we do not need to consider the signs of g^i_j and all z_{ij} variables can be removed, as* (BP$_1$-DF) *coincides with* (SLR).

Remark 3.5. *The proposed MILP formulation can be further strengthened as follows. Assume there exist $j_1, j_2 \in [n]$ such that $g^i_{j_1} \leq g^i_{j_2}$ for all $i \in [m]$. Clearly, if $y + e^{j_1} \notin \mathcal{S}(x)$ for some $y \in \mathcal{S}(x)$ and $y_{j_1} = 0$, then $y + e^{j_2} \notin \mathcal{S}(x)$. Thus, we can replace constraint* (BP$_1$-DF-d) *for j_2 with*

$$\sum_{i=1}^{m} z_{ij_2} + y_{j_1} y_{j_2} \geq y_{j_1}. \tag{3.1}$$

Constraint (3.1) takes the value of y_{j_1} into consideration: if $y_{j_1} = 0$, then constraints (BP$_1$-DF-c) *and* (BP$_1$-DF-d) *for j_1 ensure $y + e^{j_1} \notin \mathcal{S}(x)$. Hence, the values of z_{ij_2} are not required to be considered, as $y + e^{j_2} \notin \mathcal{S}(x)$ is already implied by $y + e^{j_1} \notin \mathcal{S}(x)$; otherwise, if $y_{j_1} = 1$, constraint (3.1) reduces to* (BP$_1$-DF-d). *We can linearize the nonlinear item $y_{j_1} y_{j_2}$ through McCormick envelopes [50] by introducing additional binary variables.*

Next, we discuss how to generalize the MILP formulation (BP$_1$-DF) for (BP$_1$) to (BP$_k$).

3.4.2 Formulation for (BP$_k$)

Similar to the MILP formulation (BP$_1$-DF), we introduce binary variables z_{iw} for $i \in [m]$ and $w \in \mathcal{T}^k$ to verify the condition that $y + w \notin \mathcal{S}(x)$ (recall Proposition 3.1). We then reformulate (BP$_k$) as:

$$\eta^*_k = \max_{x,y,z} \alpha^1 x + \alpha^2 y \tag{BP$_k$-DF-a}$$

$$\text{s.t.} \quad (x, y) \in \mathcal{S}, \tag{BP$_k$-DF-b}$$

$$a^i x + g^i y + z_{iw}(\mu_i - h_{iw}) \geq \mu_i \quad \forall i \in [m], w \in \mathcal{T}^k, \tag{BP$_k$-DF-c}$$

$$\sum_{i=1}^{m} z_{iw} + w^\top y + \|w^-\|_1 \geq 1 \quad \forall w \in \mathcal{T}^k, \tag{BP$_k$-DF-d}$$

$$z_{iw} \in \{0,1\} \quad \forall i \in [m],\ w \in \mathcal{T}^k, \tag{BP$_k$-DF-e}$$

where μ_i is a sufficiently small constant chosen as in Section 3.4.1; $h_{iw} = d_i + 1 - g^i w$ for all $i \in [m]$ and $w \in \mathcal{T}^k$; and $\|w^-\|_1$ is the number of entires of w with negative values.

If $z_{iw} = 1$, then constraint (BP$_k$-DF-c) implies that $a^i x + g^i(y + w) \geq d_i + 1$; on the contrary, if $z_{iw} = 0$, then the associated constraint (BP$_k$-DF-c) is redundant. As for constraints (BP$_k$-DF-d), first note that $w^\top y + \|w^-\|_1 \geq 0$ for all $y \in \{0,1\}^n$ and $w^\top y + \|w^-\|_1 = 0$ if and only if $y + w \in \{0,1\}^n$. Hence, $w^\top y + \|w^-\|_1 > 0$ implies $y + w \notin \mathcal{S}(x)$. When $w^\top y + \|w^-\|_1 = 0$, we ensure $y + w \notin \mathcal{S}(x)$ by again employing Proposition 3.1, ensuring that there exists at least one $i \in [m]$ such that $z_{iw} = 1$, which further implies that $y + w \notin \mathcal{S}(x)$ by constraint (BP$_k$-DF-c). Hence, the overall combination of constraints (BP$_k$-DF-c) and (BP$_k$-DF-d) ensure $y + w \notin \mathcal{S}(x)$.

For a fixed k, the cardinality of $\mathcal{T}^k$ is $O(n^k)$ and the number of variables and constraints in (BP$_k$-DF) is $O(mn^k)$. Therefore, the above reformulation is of polynomial size for any fixed k. This is an interesting observation from the theoretical perspective in two respects. First, for a fixed value of k, we can say that (BP$_k$) is polynomially reducible to an MILP. It immediately follows that (BP$_k$) is in class NP; recall our discussions in Remark 3.2. Furthermore, if for some fixed k, any follower's k-optimal solution is globally optimal, then based on Proposition 3.3, we have (BP) $\equiv$ (BP$_k$). Thus, we conclude that the decision version of such a class of bilevel problems is in NP-complete and the problem itself is NP-hard. For example, in Section 3.5 we show that (BP) $\equiv$ (BP$_2$) for a general class of bilevel matroid problems.

From the practical implementation perspective, the magnitude of $O(n^k)$ is still substantial, even for small values of k, when n is large. Thus, we now consider two essential questions: how to reduce the size of (BP$_k$-DF) through some simple preprocessing to remove irrelevant members of $\mathcal{T}^k$ and how to efficiently enumerate a relevant subset of $\mathcal{T}^k$ in an efficient manner.

We first discuss procedures for fixing some variable values and removing redundant variables and constraints from (BP$_k$-DF).

Proposition 3.7.(*i*) *If $g^i w \leq 0$ for some $w \in \mathcal{T}^k$ and $i \in [m]$, then $z_{iw} = 0$ for any feasible solution of* (BP$_k$-DF).

(*ii*) *If $h_{iw^0} = h_{iw^1}$ for some $i \in [m]$ and $w^0, w^1 \in \mathcal{T}^k$, then there exists an optimal solution to* (BP$_k$-DF) *in which $z_{iw^1} = z_{iw^0}$.*

Proof. (*i*) Suppose (x, y, z) is a feasible solution of (BP$_k$-DF). Since $(x, y) \in \mathcal{S}$, then $a^i x + g^i y \leq d_i$. If $z_{iw} = 1$, then from constraint (BP$_k$-DF-c), we have

$$a^i x + g^i y \geq d_i + 1 - g^i w \geq d_i + 1,$$

where the second inequality follows from the assumption that $g^i w \leq 0$. Thus, we obtain a contradiction and it immediately follows that z_{iw} should be equal to 0 in any feasible solution.

(*ii*) Suppose (x^*, y^*, z^*) is an optimal solution of (BP$_k$-DF), and $z^*_{iw^0} \neq z^*_{iw^1}$. Without loss of generality, assume $z^*_{iw^0} = 1$ and $z^*_{iw^1} = 0$. Define z' as follows: $z'_{iw^1} = 1$ and $z'_{iw} = z^*_{iw}$ for all $i \in [m]$ and $w \in \mathcal{T}^k \setminus \{w^1\}$. Observe that constraints (BP$_k$-DF-d) are trivially satisfied for (x^*, y^*, z'). Since $z^*_{iw^0} = 1$ and $h_{iw^0} = h_{iw^1}$, then $a^i x^* + g^i y^* \geq h_{iw^1}$. Thus, we can verify that constraints (BP$_k$-DF-c) also hold for (x^*, y^*, z'). Consequently, it follows that (x^*, y^*, z') is also optimal for (BP$_k$) and $z'_{iw^0} = z'_{iw^1}$.

■

Note that when we either fix $z_{iw^1} = 0$ or set it equal to z_{iw^0} for some $i \in [m]$ and $w^0, w^1 \in \mathcal{T}^k$, the corresponding constraint (BP$_k$-DF-c) can also be removed. Our computational study in Section 3.6 indicates that the preprocessing procedures in Proposition 3.7 can significantly decrease the number of variables and constraints (BP$_k$-DF-c). Also, observe that the computational efforts required to evaluate conditions (*i*) and (*ii*) in Proposition 3.7 are of order $O(nm|\mathcal{T}^k|)$, and $O(m|\mathcal{T}^k|\log(|\mathcal{T}^k|))$, respectively. Thus, the cardinality of $\mathcal{T}^k$ is a primary driver of the efficiency of the proposed approach. In view of this, we next discuss how to prune the components in $\mathcal{T}^k$ in order to effectively reduce the formulation size.

Proposition 3.8. *(i)* *Given $w^0 \in \mathcal{T}^{k-1}$ and $\ell \in [n]$ such that $w^0_\ell = 0$, let $w^1 = w^0 + e^\ell \in \mathcal{T}^k$. If $g^i_\ell \geq 0$ for all $i \in [m]$, then removing variables z_{iw^1} and the associated constraints for all $i \in [m]$ in (BP$_k$-DF) does not change its optimal objective function value.*

(ii) *Given $w^0 \in \mathcal{T}^k$ and $\ell \in [n]$ such that $w^0_\ell = 1$, let $w^1 = w^0 - e^\ell \in \mathcal{T}^{k-1}$. If $g^i_\ell \leq 0$ for all $i \in [m]$, then removing variables z_{iw^1} and the associated constraints for all $i \in [m]$ in (BP$_k$-DF) does not change its optimal objective function value.*

(iii) *If $\alpha^2 w > 0$ for some $w \in \mathcal{T}^k$, and $g^i w \geq 0$ for all $i \in [m]$, then removing variables z_{iw} and the associated constraints for all $i \in [m]$ in (BP$_k$-DF) does not change its optimal objective function value.*

Proof. *(i)* Denote by (BP$'_k$-DF) and η'_k the problem and its optimal objective function value, respectively, where z_{iw^1} and associated constraints are removed from (BP$_k$-DF). It is clear that $\eta'_k \geq \eta^*_k$. It suffices to show that given any optimal solution (x', y', z') in (BP$'_k$-DF), there exists z^* such that (x', y', z^*) is a feasible solution of (BP$_k$-DF), which leads to $\eta'_k \leq \eta^*_k$

If $y' + w^1 \notin \{0,1\}^n$, then let $z^*_{iw^1} = 0$ for all $i \in [m]$, and $z^*_{iw} = z'_{iw}$ for all $w \in \mathcal{T}^k \setminus \{w^1\}$. Clearly, (x', y', z^*) is feasible for (BP$_k$-DF). Otherwise, if $y' + w^1 \in \{0,1\}^n$, then based on constraint (BP$'_k$-DF-d) for w^0, we have $\sum_{i=1}^m z'_{iw^0} \geq 1$. Since $g^i_\ell \geq 0$, we have $h_{iw^1} = d_i + 1 - g^i w^1 \leq h_{iw^0}$ for all $i \in [m]$. Let $z^*_{iw^1} = z'_{iw^0}$, and $z^*_{iw} = z'_{iw}$ for all $w \in \mathcal{T}^k \setminus \{w^1\}$, then we can verify that (x', y', z^*) is feasible for (BP$_k$-DF). This observation completes the proof.

(ii) The proof is similar to *(i)* above, and omitted for brevity.

(iii) Denote by (BP$'_k$-DF) and η'_k the corresponding problem and its optimal objective function value, respectively where z_{iw} and associated constraints are removed from $\mathcal{T}^k$ in (BP$_k$-DF). It is clear that $\eta'_k \geq \eta^*_k$. Suppose (x', y', z') is an optimal solution of (BP$'_k$). We next focus on proving that $y' \in \mathcal{R}_k(x)$, which implies that there exists z^* such that (x', y', z^*) is feasible for (BP$_k$-DF) and $\eta'_k \leq \eta^*_k$. Note that to verify $y' \in \mathcal{R}_k(x)$, we only need to show that $y' + w \notin \mathcal{S}(x)$.

If $y' + w \notin \{0,1\}^n$, then $y' + w \notin \mathcal{S}(x)$ trivially. Otherwise, we proceed by contradiction, so suppose $y' + w \in \mathcal{S}(x)$. Since $g^i w \geq 0$, then we can verify that $(x', y' + w)$ is also feasible for (BP$'_k$-DF). Moreover, $\alpha^1 x' + \alpha^2 (y' + w) > \alpha^1 x' + \alpha^2 y'$, which contradicts our initial

Algorithm 6 Algorithm for constructing $\mathcal{T}^k$

Input $A, G, d, \alpha^1, \alpha^2, \beta$
1: $\mathcal{T}^k \leftarrow \emptyset$
2: $J_G^+ \leftarrow \{j \in [n] : g_j^i \geq 0 \; \forall i \in [m]\}$
3: $J_G^- \leftarrow \{j \in [n] : g_j^i \leq 0 \; \forall i \in [m]\}$
4: **for** all $w \in \{-1, 0, 1\}^n$ such that $\|w\|_1 \leq k$ **do**
5:     **if** $\beta w \leq 0$ **then**
6:         Discard w and go to Line 4
7:     **end if**
8:     $\beta_0 \leftarrow \min\{\beta_j : \; w_j = 1 \text{ and } j \in J_G^+\}$
9:     **if** $\beta w \geq \beta_0$ **then**
10:         Discard w and go to Line 4   *// based on Proposition 3.8*(i)
11:     **end if**
12:     **if** $\exists j \in J_G^-$ such that $w_j = -1$ **then**
13:         Discard w and go to Line 4   *// based on Proposition 3.8*(ii)
14:     **end if**
15:     **if** $\alpha^2 w > 0$ and $g^i w > 0$ for all $i \in [m]$ **then**
16:         Discard w and go to Line 4   *// based on Proposition 3.8*(iii)
17:     **end if**
18:     $\mathcal{T}^k \leftarrow \mathcal{T}^k \cup \{w\}$
19: **end for**
Return $\mathcal{T}^k$

assumption that (x', y', z') is an optimal solution for (BP'_k-DF). Thus, $y' + w \notin \mathcal{S}(x)$ and the result follows.

■

We outline the pseudocode of the procedure to construct $\mathcal{T}^k$ in Algorithm 6. In lines 8-11, we first find the component $\ell \in [n]$, for which the corresponding value of β is smallest among $j \in J_G^+$ for which $w_j = 1$, where $J_G^+ := \{j \in [n] : g_j^i \geq 0 \; \forall i \in [m]\}$. Then w is discarded based on Proposition 3.8(i). Following Proposition 3.8(ii), we discard w in lines 12-14 if there exists $\ell \in J_G^- = \{j \in [n] : g_j^i \leq 0 \; \forall i \in [m]\}$ such that $w_\ell = -1$. We finally evaluate the conditions of Proposition 3.8(iii) in lines 15-17 to determine whether to add w into $\mathcal{T}^k$.

3.4.3 Strengthened Formulation for (BP_k)

Next, we provide a strengthened formulation based on the inherent structural properties of (BP_k). Though the results here are derived from first principles, we note that the properties we exploit and the resulting formulations are closely related to the mixing-set inequalities, which have been studied in a number of contexts [9, 46, 130].

To derive the stronger formulation, we first re-write constraints (BP$_k$-DF-c) as

$$a^i x + g^i y \geq \mu_i - (\mu_i - h_{iw}) z_{iw} \ \forall i \in [m], w \in \mathcal{T}^k. \tag{3.2}$$

This rewriting is to emphasize that the combined effect of the additional binary variables introduced is to essentially set the right-hand side of a single constraint to one of a number of different possible values. This is illustrated even more clearly by combining the inequalities (3.2) associated with $i \in [m]$ into the single inequality

$$a^i x + g^i y \geq \max_{w \in \mathcal{T}^k} \{\mu_i - (\mu_i - h_{iw}) z_{iw}\} \ \forall i \in [m]. \tag{3.3}$$

Of course, this inequality involves a non-linear function, but there is a way to combine the inequalities in a different way that yields a strong linear formulation that we describe next.

The next property that we use is that after the preprocessing procedure described in Proposition 3.7, the values of h_{iw} are distinct in the remaining constraints (BP$_k$-DF-c) for each $i \in [m]$. Denote the number of distinct values of h_{iw} for $i \in [m]$ as ℓ_i. We assume without loss of generality that $h_{iw^1} < h_{iw^2} < \cdots < h_{iw^{\ell_i}}$ for $w^1, w^2, \ldots, w^{\ell_i}$ in $\mathcal{T}^k$.

The strengthened formulation for (BP$_k$) is then given by:

$$\tilde{\eta}_k^* = \max_{x,y,z} \alpha^1 x + \alpha^2 y \tag{BP$_k$-Mix-a}$$

$$\text{s.t.} \quad (x, y) \in \mathcal{S}, \tag{BP$_k$-Mix-b}$$

$$a^i x + g^i y + \sum_{j=1}^{\ell_i} (h_{iw^{j-1}} - h_{iw^j}) z_{iw^j} \geq \mu_i \quad \forall i \in [m], \tag{BP$_k$-Mix-c}$$

$$z_{iw^j} \geq z_{iw^{j+1}} \quad \forall j \in [\ell_i], i \in [m], \tag{BP$_k$-Mix-d}$$

$$\sum_{i=1}^{m} z_{iw} + w^\top y + \|w^-\|_1 \geq 1 \quad \forall w \in \mathcal{T}^k, \tag{BP$_k$-Mix-e}$$

$$z_{iw^j} \in \{0, 1\} \quad \forall j \in [\ell_i], i \in [m], \tag{BP$_k$-Mix-f}$$

where $h_{iw^0} = \mu_i$ for all $i \in [m]$. The underlying concept is the one illustrated earlier, that the value of the (variable) right-hand side is, in fact, controlled by the largest h_{iw} whose corresponding variable z_{iw} is equal to one (as illustrated in (3.3)), so that the whole set of constraints involving the original constraint $i \in [m]$ can be collapsed into a single constraint that dominates the set of original ones.

We refer the readers for more technical discussion about the extended formulation of the mixing-set inequality in [78]. Note that the z variables here ultimately play the same role as in the original formulation, but we enforce more structure on them with the addition of the precedence constraints (BP$_k$-Mix-d) for reasons that will become clear in the proof of the next theorem.

Theorem 3.4. (BP$_k$-Mix) *and* (BP$_k$-DF) *have the same optimal objective function values, that is,* $\eta_k^* = \tilde{\eta}_k^*$.

Proof. Let (x, y, z) be a feasible solution for (BP$_k$-DF). We show that there exists a feasible solution to (BP$_k$-Mix) with the same objective function value. For $i \in [m]$, let $j_0 = \max\{j \in [\ell_i] : z_{iw^j} = 1\}$. Let $z'_{iw^j} = 1$ for all $j = 1, \ldots, j_0$ and $z'_{iw^j} = 0$ for $j = j_0 + 1, \ldots, \ell_i$. Recalling (3.3), we have $a^i x + g^i y - h_{iw^{j_0}} \geq 0$. Then

$$\begin{aligned} a^i x + g^i y + \sum_{j=1}^{\ell_i} (h_{iw^{j-1}} - h_{iw^j}) z'_{iw^j} &= a^i x + g^i y + \sum_{j=1}^{j_0} (h_{iw^{j-1}} - h_{iw^j}) \\ &= a^i x + g^i y + h_{iw^0} - h_{iw^{j_0}} \\ &= a^i x + g^i y + \mu_i - h_{iw^{j_0}} \\ &\geq \mu_i. \end{aligned}$$

Therefore, (x, y, z') satisfies constraints (BP$_k$-Mix-c) and is feasible for (BP$_k$-Mix). It also has the same objective function value (BP$_k$-Mix) as (x, y, z) has in (BP$_k$-DF). It immediately yields that $\eta_k^* \leq \tilde{\eta}_k^*$.

On the other hand, let (x', y', z') be a feasible solution of (BP$_k$-Mix). For $i \in [m]$, let $j_0 = \max\{j \in [\ell_i] : z'_{iw^j} = 1\}$. Based on constraint (BP$_k$-Mix-c), we have

$$\begin{aligned} a^i x' + g^i y' + \sum_{j=1}^{\ell_i} (h_{iw^{j-1}} - h_{iw^j}) z'_{iw^j} &= a^i x' + g^i y' + h_{iw^0} - h_{iw^{j_0}} \\ &= a^i x' + g^i y' + \mu_i - h_{iw^{j_0}} \\ &\geq \mu_i \end{aligned}$$

which results in $a^i x' + g^i y' \geq h_{iw^{j_0}}$. Observe that constraints (BP$_k$-DF-c) are trivially satisfied for i and $j \geq j_0$ because $z'_{iw^j} = 0$ for $j \geq j_0$. If $j < j_0$, then $z'_{iw^j} = 1$, and we have

$a^i x' + g^i y' \geq h_{iw^{j_0}} > h_{iw^j}$, which implies that (x', y', z') satisfies constraints (BP$_k$-DF-c). Therefore, (x', y', z') is itself feasible for (BP$_k$-DF) and $\tilde{\eta}_k^* \leq \eta_k^*$. ■

When we remove the integrality constraints of decision variables in a MILP, the resulting linear optimization problem (LP) is referred to as the LP relaxation of the original MILP. It is known that the tightness of the LP relaxations for MILPs is a critical factor affecting the overall performance of the solver. We then show that the MILP formulation (BP$_k$-Mix) is stronger than (BP$_k$-DF).

Proposition 3.9. *The LP relaxation of* (BP$_k$-Mix) *is at least as strong as that of* (BP$_k$-DF).

Proof. Let (x, y, z) be any feasible solution for the LP relaxation of (BP$_k$-Mix). It suffices to show that (x, y, z) is also feasible for the LP relaxation of (BP$_k$-DF). Since constraints (BP$_k$-DF-d) are also included in (BP$_k$-Mix), we need to show that (x, y, z) satisfies constraints (BP$_k$-DF-c). Based on constraints (BP$_k$-Mix-c) and (BP$_k$-Mix-d), we have, for any $i \in [m]$, and $j_0 \in [\ell_i]$,

$$
\begin{aligned}
a^i x + g^i y + (\mu_i - h_{iw^{j_0}}) z_{iw^{j_0}} &= a^i x + g^i y + \sum_{j=1}^{j_0} (h_{iw^{j-1}} - h_{iw^j}) z_{iw^{j_0}} \\
&\geq a^i x + g^i y + \sum_{j=1}^{j_0} (h_{iw^{j-1}} - h_{iw^j}) z_{iw^j} \\
&\geq \mu_i,
\end{aligned}
$$

where the first inequality follows from $z_{iw^j} \geq z_{iw^{j+1}}$ (recall constraint (BP$_k$-Mix-d)) and $h_{iw^{j-1}} - h_{iw^j} < 0$; the second inequality follows from our initial assumption that (x, y, z) satisfies constraint (BP$_k$-Mix-c). ■

We next illustrate Theorem 4 and Proposition 3.9 with the following example.

Example 3.4. *Consider an instance of the bilevel problem:*

$$
\min_{x \in \mathbb{R}} \max_{y} \{y_1 + y_2 : \; 2y_1 + 3y_2 \leq 4, \; y \in \{0, 1\}^2\}.
$$

If $k = 1$, then $\mathcal{T}_1 = \{e^1, e^2\}$. Let z_1, z_2 be the binary variables in (BP$_k$-DF) *that correspond to e^1 and e^2, respectively. We set $\mu = 0$ for constraint* (BP$_k$-DF-c)*. Then the feasible region for the LP relaxation of* (BP$_k$-DF) *with $k = 1$ is given as:*

$$\mathcal{Q}^1_{DF} = \left\{ (x,y,z) \in \mathbb{R} \times [0,1]^4 : \begin{array}{l} 2y_1 + 3y_2 \leq 4, \\ 2y_1 + 3y_2 \geq 3z_1, \\ 2y_1 + 3y_2 \geq 2z_2, \\ y_1 + z_1 \geq 1, \\ y_2 + z_2 \geq 1 \end{array} \right\}.$$

The feasible region for the LP relaxation of (BP$_k$-Mix) *with* $k = 1$ *is given as:*

$$\mathcal{Q}^1_{Mix} = \left\{ (x,y,z) \in \mathbb{R} \times [0,1]^4 : \begin{array}{l} 2y_1 + 3y_2 \leq 4, \\ 2y_1 + 3y_2 \geq 2z_2 + z_1, \\ y_1 + z_1 \geq 1, \\ y_2 + z_2 \geq 1, \\ z_2 \geq z_1 \end{array} \right\}.$$

It is easy to verify that $\mathcal{Q}^1_{Mix} \subseteq \mathcal{Q}^1_{DF}$. *Meantime, note that* $(0, 1, \frac{2}{3}, 1, \frac{1}{3})^\top \in \mathcal{Q}^1_{DF}$. *However, solution* $(0, 1, \frac{2}{3}, 1, \frac{1}{3})^\top$ *is not feasible in* $\mathcal{Q}^1_{Mix}$. *It immediately follows that* $\mathcal{Q}^1_{Mix} \subset \mathcal{Q}^1_{DF}$.

Remark 3.6. *Based on the results of the mixing-set in [9, 46], we can show that for any* $i \in [m]$, *the inequalities*

$$a^i x + g^i y + \sum_{j=1}^{\ell} (h_{iw^{t_{j-1}}} - h_{iw^{t_j}}) z_{iw^{t_j}} \geq \mu_i \quad \forall \{w^{t_1}, w^{t_2}, \ldots, w^{t_\ell}\} \subseteq \mathcal{T}^k,$$

where $h_{iw^{t_1}} < h_{iw^{t_2}} < \cdots < h_{iw^{t_\ell}}$ *are valid for* (BP$_k$-DF). *The above inequalities are called star inequalities, which can be further shown to describe the convex hull of a mixing-set polytope; see [9, 46]. In our strengthened formulation* (BP$_k$-Mix), *we note that constraint* (BP$_k$-Mix-c) *is a special class of star inequalities. We exploit the inherent structure in* (BP$_k$) *to simplify the star inequalities and provide the stronger formulation* (BP$_k$-Mix).

3.5 Bilevel Matroid Optimization

Next, we focus on exploring in more detail a special case of (BP_k). Specifically, by exploiting Proposition 3.3 we show that (BP) $\equiv$ (BP_k) for a general class of bilevel matroid problems whenever $k \geq 2$.

Definition 3.5 ([70])**.** *Let $[n] = \{1, \ldots, n\}$ be a finite set, and let $\mathcal{F}$ be a set of subsets of $[n]$. We say that $M = ([n], \mathcal{F})$ is a matroid if the following conditions are satisfied:*

(i) $\emptyset \in \mathcal{F}$;

(ii) $S \in \mathcal{F}$ *and* $S' \subseteq S$ *implies* $S' \subseteq \mathcal{F}$;

(iii) *for any* $S, S' \in \mathcal{F}$, *if* $|S| > |S'|$, *then there exists* $j \in S \setminus S'$ *such that* $S' \cup \{j\} \in \mathcal{F}$.

Elements of $\mathcal{F}$ are called independent sets, and the remaining sets of $[n]$ are called dependent sets.

Denote the set that contains the characteristic vectors of all independent sets of a matroid $M = ([n], \mathcal{F})$ as:

$$\mathcal{I} = \{y^S \in \{0,1\}^n : \ S \in \mathcal{F}\},$$

where y^S is the characteristic vector of set S such that $y^S_j = 1$ for $j \in S$, and $y^S_j = 0$, otherwise. The basic properties of $\mathcal{I}$ are shown as follows.

Lemma 3.1. *If $\mathcal{I}$ is the set of characteristic vectors of the independent sets of a matroid, then $\mathcal{I}$ satisfies the following statements:*

(i) $0 \in \mathcal{I}$;

(ii) *given* $y, y' \in \{0,1\}^n$ *and* $y \leq y'$, *if* $y' \in \mathcal{I}$, *then* $y \in \mathcal{I}$;

(iii) *if* $y, y' \in \mathcal{I}$, *and* $\|y\|_1 < \|y'\|_1$, *then there exists* $j \in \{i \in [n] : \ y_i = 0, y'_i = 1\}$ *such that* $y + e^j \in \mathcal{I}$.

Proof. It directly follows from Definition 3.5. ■

Denote by $\mathcal{F}_B$ and $\mathcal{I}_B$ the set of all maximal independent sets in $\mathcal{F}$ and the set of its corresponding characteristic vectors, respectively. The most fundamental matroid optimization problems are maximum-weight independent set problem $\max\{\beta y : \ y \in \mathcal{I}\}$ and minimum-weight basis problem $\min\{\beta y : \ y \in \mathcal{I}_B\}$.

It is well known that these two matroid optimization problems are polynomially solvable by a greedy algorithm [69, 70], which iteratively selects an element with the largest/smallest weight among the remaining elements. In particular, observe that the obtained greedy solutions are 2-optimal. We next provide a self-contained proof to show that if the lower-level feasible region contains the matroid structure, (BP) is equivalent to (BP_k) whenever $k \geq 2$.

Lemma 3.2. *Suppose $\mathcal{S}(x)$ is the characteristic vector set of all independent sets of a matroid for any $x \in \mathcal{X}$.*

(i) *If $y, y' \in \mathcal{R}_k(x)$ for some $k \geq 1$ and $y' \leq y$, then $\beta y' = \beta y$.*

(ii) *Given $y, y' \in \mathcal{S}(x)$ and $y' \leq y$, if $y' \in \mathcal{R}_k(x)$ for some $k \geq 1$, then $y \in \mathcal{R}_k(x)$.*

Proof. (i) Since $\beta \geq 0$ and $y' \leq y$, we have $\beta y' \leq \beta y$. Suppose $\beta y' < \beta y$. Then there exists at least one j such that $y'_j = 0, y_j = 1$ and $\beta_j > 0$. By Lemma 3.1(ii), we have $y' + e^j \in \mathcal{S}(x)$. Then we have a contradiction by observing that $y' + e^j \in \mathcal{N}_k(y)$ for $k \geq 1$ and $\beta(y' + e^j) > \beta y'$.

(ii) According to the definition of $\mathcal{R}_k(x)$, we need to show that $\beta y \geq \beta \bar{y}$ for any $\bar{y} \in \mathcal{N}_k(y) \cap \mathcal{S}(x)$. Let $w = y - \bar{y}$, then $w \in \{-1, 0, 1\}^n$ and $\|w\|_1 \leq k$.
Construct $w' \in \{-1, 0, 1\}^n$ such that $w'_j = 0$ if $w_j = 1$ and $y'_j = 0$; otherwise $w'_j = w_j$. Thus, $\|w'\|_1 \leq \|w\|_1 \leq k$ and $y' - w' \in \{0, 1\}^n$. Observe that $y' - w' \leq y - w = \bar{y}$. Thus, based on Lemma 3.1(ii), we have $y' - w' \in \mathcal{S}(x)$. Following our condition that $y' \in \mathcal{R}_k(x)$, we have $\beta y' \geq \beta(y' - w')$ as $y' - w' \in \mathcal{N}_k(y') \cap \mathcal{S}(x)$, which implies that $\beta y \geq \beta(y - w') \geq \beta(y - w)$ and the result follows.

■

Theorem 3.5.(i) *If $\mathcal{S}(x)$ is the characteristic vector set of all independent sets of a matroid for any $x \in \mathcal{X}$, then (BP) $\equiv$ (BP_k) for any integer $k \geq 2$.*

(ii) *If the follower solves a minimization problem, and $\mathcal{S}(x)$ is the characteristic vector set of all maximal independent sets of a matroid, then (BP) $\equiv$ (BP_k) for any integer $k \geq 2$.*

Proof. For brevity, we only provide the proof for (i); the proof for (ii) can be derived in a similar manner. By Proposition 3.3, it is sufficient to show that for any leader's decision x the corresponding 2-optimal follower's response is also a globally optimal solution for the lower-level problem. That is, $\mathcal{R}_2(x) = \mathcal{R}(x)$. Since $\mathcal{R}(x) \subseteq \mathcal{R}_2(x)$, we only need to verify that $\mathcal{R}_2(x) \subseteq \mathcal{R}(x)$ for any $x \in \mathcal{X}$.

Based on Lemmas 3.2(i) and 3.2(ii), it suffices to focus on the maximal independent set in $\mathcal{R}_2(x)$ and $\mathcal{R}(x)$. Let y' and y^* be the characteristic vector of any maximal independent set in $\mathcal{R}_2(x)$ and $\mathcal{R}(x)$, respectively. It is clear that $\|y'\|_1 = \|y^*\|_1$ based on Lemma 3.1(iii). We next show that $\beta y' = \beta y^*$, which implies that $y' \in \mathcal{R}(x)$.

Without loss of generality, assume $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$; also, suppose $y' = \sum_{j=1}^{t} e^{i'_j}$ and $y^* = \sum_{j=1}^{t} e^{i^*_j}$ such that $i'_1 < i'_2 < \cdots < i'_t$ and $i^*_1 < i^*_2 < \cdots < i^*_t$. Suppose there exists $j \in [t]$ such that $\beta_{i'_j} \neq \beta_{i^*_j}$, let $\ell = \min\{j \in [t] : \beta_{i'_j} \neq \beta_{i^*_j}\}$. Then we need to consider two cases: $\beta_{i'_\ell} < \beta_{i^*_\ell}$, and $\beta_{i'_\ell} > \beta_{i^*_\ell}$.

We first discuss the case that $\beta_{i'_\ell} < \beta_{i^*_\ell}$. Since $\sum_{j=1}^{\ell} e^{i^*_j} \in \mathcal{I}$, $\sum_{j=1}^{\ell-1} e^{i'_j} \in \mathcal{I}$ and $\| \sum_{j=1}^{\ell} e^{i^*_j}\|_1 > \| \sum_{j=1}^{\ell-1} e^{i'_j}\|_1$, based on Lemma 3.1($iii$), there exists $j^* \in \{1, \ldots, \ell\}$ such that $\sum_{j=1}^{\ell-1} e^{i'_j} + e^{i^*_{j^*}} \in \mathcal{I}$. Observe that $\| \sum_{j=1}^{\ell-1} e^{i'_j} + e^{i^*_{j'}}\|_1 < \|y'\|_1$. Based on Lemma 3.1($iii$), there exists j' such that $y'_{i'_{j'}} = 1$, $i'_{j'} \geq i'_\ell$, and $\bar{y} := y' - e^{i'_{j'}} + e^{i^*_{j^*}} \in \mathcal{I}$. Note that $\bar{y} \in \mathcal{N}_2(y')$ and $\beta\bar{y} > \beta y'$ as $\beta_{i^*_{j^*}} \geq \beta_{i^*_\ell} > \beta_{i'_\ell} \geq \beta_{i'_{j'}}$, which contradicts with the assumption that $y' \in \mathcal{R}_2(x)$. Thus, the first case (i.e., $\beta_{i'_\ell} < \beta_{i^*_\ell}$) is considered.

The proof for the case of $\beta_{i'_\ell} > \beta_{i^*_\ell}$ is similar and omitted for brevity. Therefore, we have that $\beta_{i'_j} = \beta_{i^*_j}$ for all $j \in [t]$ and the result immediately follows.

■

To illustrate Theorem 3.5, we next outline several bilevel problems such that their lower-level problems are reducible to a matroid optimization problem. Specifically, we focus on problems with

$$\mathcal{S} = \{(x, y) \in \mathcal{X} \times \{0,1\}^n : \; y \in \mathcal{I}(x)\},$$

where $\mathcal{I}(x)$ is some characteristic vector set of a matroid for any $x \in \mathcal{X}$. We also consider two special cases of the above set given by:

- $\mathcal{S} = \{(x, y) \in \mathcal{X} \times \{0,1\}^n : \; y \in \mathcal{I}, \; x + y \leq 1\}$, and $\mathcal{I}$ is the characteristic vector set of a matroid; and
- $\mathcal{S} = \{(x, y) \in \mathcal{X} \times \{0,1\}^n : \; x + y \in \mathcal{I}\}$, and $\mathcal{I}$ is the characteristic vector set of a matroid.

There are a number of single-level combinatorial optimization problems that contain the matroid structure; we refer the reader to [70, 90, 113] and the references therein. In the literature, there are several bilevel generalizations of these problems with the following feasible sets at the lower level:

(i) The knapsack set with a cardinality constraint: Given a positive budget C, let

$$\mathcal{I} = \{y \in \{0,1\}^n : \sum_{j=1}^{n} y_j \leq C\},$$

which is the characteristic vector set of the uniform matroid; see, e.g., [113]. The bilevel knapsack problem has been extensively studied in recent years; see [22, 23, 29, 32].

(ii) The knapsack problem with multiple disjoint cardinality constraints: Given a partition of set $[n]$, $\{N_i\}_{i=1}^{r}$ and budgets C_i for each class i, let

$$\mathcal{I} = \{y \in \{0,1\}^n : \sum_{j \in N_i} y_j \leq C_i \ \forall i \in [r]\},$$

which is the characteristic vector set of the partition matroid; see, e.g., [4]. Some interesting results for the bilevel multidimensional knapsack problem are developed in [37].

(iii) The spanning tree set: Given an undirected graph $G = (N, E)$, let

$$\mathcal{I}_B = \{y \in \{0,1\}^n : \ G[y] \text{ is a spanning tree of graph } G\},$$

where $G[y] := G[E_y] = (N, E_y)$ is the subgraph induced by edges in $E_y = \{(i,j) \in E : y_{ij} = 1\}$. The spanning tree set contains all maximal independent sets of a tree matroid [4], and its bilevel versions are considered in [38, 99, 112].

(iv) Unit-time task scheduling problem: Given a set of unit-time tasks $N \in \{1, \ldots, n\}$ and their deadlines d_i for each task $i \in N$, let

$$\mathcal{I} = \{y \in \{0,1\}^n\} : \text{ there exists a schedule for tasks } N_y \text{ without delay}\},$$

where $N_y = \{i \in N : \ y_i = 1\}$. Several scheduling problems are shown to have matroid structures; see, e.g., [69, 70, 113]. Examples of bilevel scheduling problems can be found in [61, 77].

In Section 3.6.3, we provide a case study illustrating the use of (BP_k) to solve the bilevel minimum spanning tree problem (BMST) [38, 99, 112], in which the follower's optimization problem involves constructing a minimum spanning tree (MST) in a graph.

3.6 Computational Experiments

In this section, we report the results of our computational experiments with several classes of bilevel problems. We would like to point out that our main goal is not to solve general bilevel problems to optimality, but rather to evaluate (i) the quality of the proposed lower and upper bounds provided by (BP_k) as well as (ii) the performance of our MILP formulations. Therefore, we do not compare our approaches against specialized algorithms designed for solving particular classes of bilevel problems. Instead, the generic mixed integer bilevel solver MibS [104] and the SLR-based bounds are used as the main benchmarks.

This section is organized as follows. We first consider the knapsack interdiction problem (KIP) in Section 3.6.1. In Section 3.6.2, we consider the bilevel vertex cover problem. Section 3.6.3 illustrates our results developed in Section 3.5 by running the experiments on the bilevel minimum spanning tree problem. Finally, we note that our numerical experiments are performed using CPLEX 12.8 [51] on an Ubuntu 16.04 system with a 3.2GHz CPU and 19 GB of RAM.

3.6.1 Knapsack Interdiction Problem (KIP)

We consider the knapsack interdiction problem [32, 37] given as:

$$\min_{x\in\{0,1\}^n} \max_{y\in\{0,1\}^n} \sum_{j=1}^{n} \beta_j y_j \tag{KIP-a}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_j^i x_j \le h_i \quad \forall i \in [m_\ell], \tag{KIP-b}$$

$$\sum_{j=1}^{n} g_j^i y_j \le d_j \quad \forall i \in [m_f], \tag{KIP-c}$$

$$x_j + y_j \le 1 \quad \forall j \in [n], \tag{KIP-d}$$

where $\beta_j, a_j^i, g_j^i, h_i$ and d_i are positive integers for all i and j; parameters m_ℓ and m_f denote the number of knapsack constraints at the upper and lower levels, respectively. We refer to the knapsack interdiction problem with k-optimal follower as (KIP_k). Since the leader aims to minimize their objective function, we note that the optimal objective function values of

(KIP$_k$), η_k^*, provide a hierarchy of lower bounds for η^*. The objective function values $\hat{\eta}_k^*$ of bilevel feasible solutions that are constructed from (KIP$_k$), on the other hand, are natural upper bounds for η^*.

Experimental setup. To construct the test instances, we follow an approach similar to the one used in [23, 84]. The costs β_j as well as the weights a_j^i and g_j^i are generated randomly and independently using the discrete uniform distribution in interval $[0, 100]$. For each $n \in \{10, 20, 30, 40, 50\}$ and $r \in \{1, 2, \ldots, 10\}$, parameter d_i, $i \in [m_f]$, is set to $\lceil \frac{r}{11} \sum_{j=1}^{n} g_j^i \rceil$; parameter h_i, $i \in [m_\ell]$, is generated using the discrete uniform distribution in interval $[\frac{\sum_{i=1}^{m_f} d_i}{m_f} - 10, \frac{\sum_{i=1}^{m_f} d_i}{m_f} + 10]$. We construct 10 instances for each pair of n and r, and report the corresponding average performance.

In our computational experiments, we set the time limit for MibS and (KIP$_k$) to 600 seconds (10 minutes); all results are reported in seconds. Denote by η_M^* and $\hat{\eta}_M^*$ the best lower and upper bounds obtained by MibS at termination, respectively. Whenever CPLEX cannot solve (KIP$_k$) to optimality within the time limit, the best lower bound reported by CPLEX is referred to as η_k^*. Then the leader's feasible solution reported by CPLEX is used to derive the upper bound $\hat{\eta}_k^*$.

Next, we first discuss the sizes of the formulation (BP$_k$-DF) for (KIP$_k$) after the preprocessing steps based on Propositions 3.7 and 3.8 are applied. Then we first conduct the experiments on (KIP) instances with a single knapsack constraint at both levels, i.e., $m_\ell = m_f = 1$. Finally, we also explore the quality of our bounds for the instances of (KIP) with multiple knapsack constraints at the lower level.

Formulation size for (KIP$_k$). The average cardinality of $\mathcal{T}^k$ and the average number of constraints (BP$_k$-DF-c) for each n and m_f are shown in Table 8. Observe that despite the fact that the cardinality of $\mathcal{T}^k$ grows exponentially with respect to k, the number of constraints (BP$_k$-DF-c) after preprocessing has a rather moderate increase and is roughly a concave function with respect to m_f, n and k. Since the numbers of constraints (BP$_k$-DF-c) and variables z are equal, these results also indicate that we introduce a reasonably small number of additional binary variables z in our MILP reformulations for (KIP$_k$).

Table 8: The average cardinality of $\mathcal{T}^k$ and the average number of constraints (BP$_k$-DF-c) for the formulation (BP$_k$-DF) of (KIP$_k$) after the preprocessing procedure.

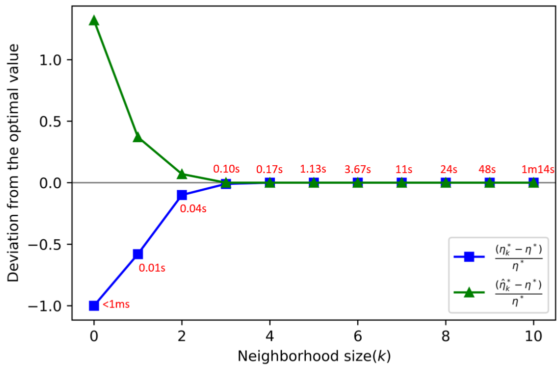
		$k=1$		$k=2$		$k=3$	
	n	$\lvert\mathcal{T}^k\rvert$	#(BP$_k$-DF-c)	$\lvert\mathcal{T}^k\rvert$	#(BP$_k$-DF-c)	$\lvert\mathcal{T}^k\rvert$	#(BP$_k$-DF-c)
$m_f = 1$	10	9	9	53	25	163	57
	20	20	18	208	61	1,324	137
	30	30	26	461	81	4,503	163
	40	40	33	812	88	10,694	174
	50	50	40	1,262	93	20,889	182
$m_f = 3$	10	10	28	54	137	174	357
	20	20	54	208	347	1,344	837
	30	30	78	460	467	4,520	986
	40	40	98	812	524	10,704	1,056
	50	50	118	1,262	549	20,856	1,087
$m_f = 5$	10	10	47	54	232	174	599
	20	20	90	208	576	1,348	1,404
	30	30	129	460	778	4,515	1,640
	40	40	165	811	868	10,702	1,750
	50	50	196	1,261	911	20,897	1,809

Results for (KIP) with a single knapsack constraint at both levels ($m_\ell = m_f = 1$). To evaluate MILP formulations (BP$_k$-DF) and (BP$_k$-Mix) for (KIP$_k$), we need first to select a tight value of μ for constraint (BP$_k$-DF-c), which corresponds to (KIP-c). Recall from our discussion in Section 3.4 that μ can be set to some lower bound for the term $\sum_{j=1}^n a_j^2 y_j$. For the knapsack interdiction problem, we separate the leader's decisions into two possible groups:

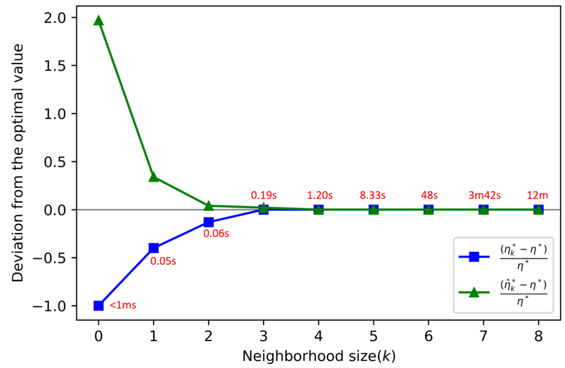
(i) if $\sum_{j=1}^n a_j^2(1-x_j) \leq C_F$ for some leader's feasible decision x, then the lower-level problem has a k-optimal solution $y_j = 1 - x_j$ for all $j \in [n]$ and any $k \geq 1$. Therefore, the value of μ for such leader's decision x can be trivially achieved by solving:

$$\mu^1 = \min_{x\in[0,1]^n} \{\sum_{j=1}^n a_j^2(1-x_j) : \sum_{j=1}^n a_j^1 x_j \leq C_L\}.$$

(ii) if $\sum_{j=1}^n a_j^2(1-x_j) > C_F$ for some leader's feasible decision x, then any follower's k-optimal solution y is maximal for $k \geq 1$, that is $\sum_{j=1}^n a_j^2 y_j \leq C_F$ and $\sum_{j=1}^n a_j^2 y_j + a_\ell^2 > C_F$ for



(a) $n = 15, r = 4$.



(b) $n = 20, r = 4$.

Figure 4: The average deviations from the optimal value of (KIP$_k$) for different k. The solution time of formulation (BP$_k$-DF) is shown in red.

any ℓ such that $y_\ell = 0$ and $x_\ell = 0$. Therefore, the value of μ for such leader's decision x can be set to:

$$\mu^2 = C_F - \max_{j \in [n]} \{a_j^2\}.$$

Therefore, we set $\mu = \min\{\lceil \mu^1 \rceil, \mu^2\}$ in our experiments.

Next, we evaluate the quality of our bounds, which are also depicted in Figure 4 for $n = 15$ and $n = 20$. The horizontal axis shows the value of k and the vertical axis indicates the deviation of the bounds from the true optimal objective function value η^* of the bilevel optimization problem. In Figure 4 we also indicate the solution time of the formulation (BP$_k$-DF) for each k, please see the labels in red.

One observation is that the bounds provided by (KIP$_k$) for $k \geq 1$ are substantially better than the SLR-based bounds (i.e., $k = 0$). For example, in Figure 4(a) for $n = 15, r = 4$, the gaps between the optimal objective function value and the SLR-based bounds are 197% and 100%, respectively. On the contrast, the bounds provided by (KIP$_1$) are only 40% and 34% away from the optimal values, respectively, and are computed within 0.02 seconds in total.

Consistent with Theorem 3.2, we can see from Figure 4 that the obtained lower bounds η_k^* improves monotonically with respect to k, but, of course, at increased computational expense.

Moreover, these lower bounds converge rapidly and the optimal value η^* is achieved at $k = 3$ for both cases, with the corresponding (KIP_3) solved to optimality within a second. As for $\hat{\eta}^*_k$, although these upper bounds are not guaranteed to be monotone with respect to k (recall our discussion in Section 3.3.2), it is interesting to observe that for almost all of the tested instances, $\hat{\eta}^*_k$ provides better upper bound with the increase of k.

For both cases in Figure 4, we observe that the required computational time is small for sufficiently small values of k. For $n = 15$ and $r = 4$, the solver can efficiently tackle the formulation (BP_k-DF) for (KIP_k) with $k \geq 10$. However, we observed memory limitations in some cases for $n = 20$ and $r = 4$, when $k \geq 9$. We attribute it to the fact that the formulation size grows considerably with the increase of k (recall our discussion in Section 3.4.2). Therefore, we next focus on examining the performance of (KIP_k) for $k \leq 3$.

Specifically, in our next set of experiments, we compare the performances of the formulations for (KIP_k) against (SLR) and the general bilevel solver, MibS [104]. The average performances for the considered solution approaches are presented in Tables 9 and 10. In particular, for MibS and (SLR) (i.e., $k = 0$), we report the average runtime in seconds (column "Time"). For (KIP_k), $k \in \{1, 2, 3\}$, the average runtime in seconds for formulations (BP_k-DF) and (BP_k-Mix) are shown in columns "Time" and "ExtTime", respectively. For each solution approach, the ratios between the achieved bounds and $\hat{\eta}^*_M$ (i.e., $\frac{\eta^*_k}{\hat{\eta}^*_M}$ and $\frac{\hat{\eta}^*_k}{\hat{\eta}^*_M}, k \in \{0, 1, 2, 3\}$) are reported in columns "ObjL" and "ObjU", respectively.

MibS succeeds in solving the small instances to optimality within the time limit up to $n = 30$. We can also observe that the instances with either sufficiently small r or large r were easier to solve, e.g., when $r \geq 6$ in Table 10. This is because when the leader has a large interdiction budget (i.e., r is large), then the feasible region of the follower is typically small, which leads to fewer bilevel feasible solutions. Similarly, a scarce budget (i.e., small r) results in a small number of feasible decisions for the leader, and also makes the overall problem easier.

From Tables 9 and 10, we observe that (SLR) provides rather poor bounds for all considered instances. On the contrast, it is usually possible to obtain a bound equal to the optimal value by solving (KIP_k) for some small k. For the easy instances in Table 10, $k = 1$ suffices. For the hard instances in Table 9, the improvements provided by (KIP_1) are less significant; nevertheless, the optimal values are attained by η^*_3 and $\hat{\eta}^*_3$ for more than half of the instances.

In Tables 9 and 10, we also highlight that the MILP formulations (BP_k-DF) and (BP_k-Mix) for all tested instances can be solved to optimality in under 20 seconds. We can also observe that there is no significant difference between these two formulations in terms of their running time performance in Tables 9 and 10.

Results for (KIP) with multiple knapsack constraints at the lower level ($m_f > 1$). Since the formulation sizes of (BP_k-DF) and (BP_k-Mix) depend only on the size of the lower-level problem, we use $m_\ell = 1$ in our experiments, i.e., there is a single constraint at the upper-level problem. For simplicity, we set $\mu_i = 0$ for all $i \in [m_f]$ in (BP_k-DF) and (BP_k-Mix).

We report the average performance of (KIP) instances with $m_f \in \{3, 5\}$ in Table 11. In particular, similar to the discussions for (KIP) with a single knapsack constraint, larger values of r correspond to easier instances. Similar to the previous set of experiments, the bounds provided by (KIP_k) dramatically improve with the increase of k at the expense of more computational efforts. We observe that the optimal value of (KIP) can be achieved by (KIP_3) in most of our test instances.

For $k = 1$, the quality of achieved bounds notably outperforms those provided by (SLR). The formulations (BP_k-DF) and (BP_k-Mix) for (KIP_1) have fairly fast and stable solution times across all the instances, which implies that (KIP_1) is very scalable. Therefore, using (KIP_1) instead of (SLR) as the initial relaxation problem could be a promising approach for speeding up the performance of general branch-and-cut solvers.

Furthermore, Table 11 shows that (BP_k-Mix) significantly outperforms (BP_k-DF) (in contrast to our previous set of experiments for problems with a single constraint). This observation can be justified by a more complex structure of the lower-level problem for test instances with multiple constraints and highlights our theoretical results in Proposition 3.9. In particular, (BP_k-Mix) provides a speedup of at least one order of magnitude for $k \geq 2$ in Table 11; see e.g., the results for $m_f = 3$, $n = 40$ and $r = 2$. On the other hand, we recognize that our proposed formulations (BP_k-DF) and (BP_k-Mix) for (KIP_k) with $k \geq 2$ are sensitive to the size of instances and the value of parameter r. We attribute it to the fact the cardinality of $\mathcal{T}^k$ remains in order of $O(n^k)$ despite our preprocessing; see Table 8.

Table 9: Results for the instances of (KIP) with $r \in \{1, 2, \dots, 5\}$. For MibS and (SLR) (i.e., $k = 0$), we report the average runtime in seconds (column "Time"). For (KIP_k), $k \in \{1, 2, 3\}$, the average runtime in seconds for formulations (BP_k-DF) and (BP_k-Mix) are shown in columns "Time" and "ExtTime," respectively. For each solution approach, the ratios between the achieved bounds and the best upper bound, $\hat{\eta}^*_M$, reported by MibS (i.e., $\frac{\eta^*_k}{\hat{\eta}^*_M}$ and $\frac{\hat{\eta}^*_k}{\hat{\eta}^*_M}, k \in \{M, 0, 1, 2, 3\}$) are shown in the columns "ObjL" and "ObjU," respectively.

		MibS [104]		$k=0$			$k=1$				$k=2$				$k=3$			
n	r	ObjL	Time	ObjL	ObjU	Time	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime
10	1	1	< 0.01	0	1.75	< 0.01	0.35	1.75	< 0.01	< 0.01	0.99	1.01	< 0.01	< 0.01	1	1	< 0.01	< 0.01
10	2	1	< 0.01	0	1.79	< 0.01	0.23	1.4	0.01	0.01	0.78	1.17	0.01	0.01	0.99	1	0.01	0.01
10	3	1	< 0.01	0	1.92	< 0.01	0.42	1.3	0.01	0.01	0.94	1.11	0.03	0.04	1	1	0.05	0.08
10	4	1	< 0.01	0	2.71	< 0.01	0.34	1.27	0.01	0.01	0.94	1.04	0.03	0.02	1	1.01	0.04	0.07
10	5	1	< 0.01	0	8.32	< 0.01	0.49	1.44	0.01	0.01	0.95	1.06	0.02	0.02	1	1	0.04	0.04
20	1	1	< 0.01	0	1.58	< 0.01	0.14	1.39	0.01	0.01	0.75	1.1	0.03	0.04	0.97	1	0.06	0.06
20	2	1	1.5	0	1.8	< 0.01	0.14	1.28	0.01	0.01	0.71	1.16	0.06	0.09	0.99	1.03	0.15	0.15
20	3	1	9.6	0	1.98	< 0.01	0.19	1.27	0.02	0.01	0.78	1.11	0.05	0.08	0.98	1.02	0.14	0.21
20	4	1	15	0	2.6	< 0.01	0.34	1.46	0.02	0.01	0.84	1.13	0.05	0.09	0.99	1	0.12	0.14
20	5	1	7.7	0	3.43	< 0.01	0.63	1.35	0.03	0.02	0.89	1.03	0.05	0.07	1	1.01	0.12	0.15
30	1	1	2.6	0	1.53	< 0.01	0.06	1.38	0.01	0.01	0.7	1.09	0.09	0.11	0.99	1	0.33	0.29
30	2	1	445.3	0	1.82	< 0.01	0.1	1.35	0.02	0.01	0.73	1.12	0.1	0.11	0.97	1.01	0.58	0.75
30	3	1	529.9	0	1.93	< 0.01	0.19	1.3	0.02	0.01	0.73	1.1	0.1	0.1	0.97	1.02	0.55	0.77
30	4	1	600	0	2.54	< 0.01	0.31	1.5	0.03	0.01	0.8	1.16	0.1	0.14	0.99	1.01	0.45	0.56
30	5	1	402.4	0	3.71	< 0.01	0.66	1.49	0.05	0.02	0.99	1.02	0.13	0.1	1	1	0.38	0.25
40	1	1	178	0	1.59	< 0.01	0.07	1.33	0.02	0.01	0.55	1.19	0.15	0.13	0.96	1.01	1.57	1.53
40	2	0.9	600	0	1.8	< 0.01	0.11	1.44	0.02	0.01	0.65	1.12	0.16	0.17	0.96	1.03	1.69	2.53
40	3	0.8	600	0	2	< 0.01	0.2	1.35	0.03	0.02	0.69	1.11	0.21	0.21	0.97	1.02	2.2	2.79
40	4	0.7	600	0	2.49	< 0.01	0.34	1.44	0.04	0.02	0.82	1.17	0.21	0.25	0.99	1	1.51	2.27
40	5	0.9	600	0	3.21	< 0.01	0.62	1.51	0.07	0.04	0.92	1.09	0.22	0.17	0.99	1	1.64	1.44
50	1	1	426.5	0	1.56	< 0.01	0.06	1.38	0.03	0.02	0.57	1.24	0.25	0.23	0.98	1.01	6.82	7.75
50	2	0.9	600	0	1.79	< 0.01	0.09	1.37	0.03	0.02	0.69	1.08	0.3	0.33	0.98	1.01	8.84	11.38
50	3	0.5	600	0	2.05	< 0.01	0.18	1.3	0.03	0.02	0.64	1.07	0.3	0.38	0.98	1.01	8.09	14.57
50	4	0.3	600	0	2.53	< 0.01	0.36	1.4	0.07	0.04	0.77	1.12	0.32	0.37	0.97	0.98	6.26	8.2
50	5	0.4	600	0	3.31	< 0.01	0.64	1.49	0.15	0.05	0.92	1.05	0.35	0.26	0.99	0.99	4.27	4.48

Table 10: Computational results for the instances of (KIP) with $r \in \{6, 7, \ldots, 10\}$. For MibS and (SLR) (i.e., $k = 0$), we report the average runtime in seconds (column "Time"). For (KIP$_k$), $k \in \{1, 2, 3\}$, the average runtime in seconds for formulations (BP$_k$-DF) and (BP$_k$-Mix) are shown in columns "Time" and "ExtTime," respectively. For each solution approach, the ratios between the achieved bounds and the best upper bound, $\hat{\eta}^*_M$, reported by MibS (i.e., $\frac{\eta^*_k}{\hat{\eta}^*_M}$ and $\frac{\hat{\eta}^*_k}{\hat{\eta}^*_M}, k \in \{M, 0, 1, 2, 3\}$) are shown in the columns "ObjL" and "ObjU," respectively.

		Mibs [104]		$k=0$			$k=1$				$k=2$				$k=3$			
n	r	ObjL	Time	ObjL	ObjU	Time	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime
10	6	1	0.02	0.12	54.65	< 0.01	0.75	1.31	0.01	0.01	0.99	1	0.01	0.01	1	1	0.01	0.01
10	7	1	0.02	0.22	81.65	< 0.01	0.95	1.04	0.01	0.01	0.99	1	0.01	0.01	1	1	0.02	0.02
10	8	1	0.02	0.02	9.35	< 0.01	1	1	< 0.01	< 0.01	1	1	< 0.01	< 0.01	1	1	0.01	0.01
10	9	1	0.01	0.23	97.48	< 0.01	1	1	< 0.01	< 0.01	1	1	< 0.01	< 0.01	1	1	0.01	0.01
10	10	1	0.01	0.26	126.21	< 0.01	1	1	< 0.01	< 0.01	1	1	< 0.01	< 0.01	1	1	0.01	0.01
20	6	1	2.02	0.01	5.82	< 0.01	0.97	1.15	0.03	0.02	1	1	0.05	0.05	1	1	0.08	0.1
20	7	1	0.98	0.01	10.1	< 0.01	0.99	1.06	0.02	0.01	1	1	0.03	0.03	1	1	0.05	0.07
20	8	1	0.32	0.03	28	< 0.01	1	1	0.02	0.01	1	1	0.02	0.01	1	1	0.04	0.07
20	9	1	0.04	0.15	157.31	< 0.01	1	1	0.01	0.01	1	1	0.01	0.01	1	1	0.02	0.05
20	10	1	0.02	0.47	435.3	< 0.01	1	1	< 0.01	< 0.01	1	1	0.01	0.01	1	1	0.01	0.03
30	6	1	282.74	0.01	6.4	< 0.01	0.99	1.23	0.06	0.04	1	1	0.1	0.1	1	1	0.25	0.18
30	7	1	19.62	0.01	9.03	< 0.01	0.99	1.08	0.06	0.02	1	1	0.07	0.05	1	1	0.15	0.16
30	8	1	3.19	0.02	33.22	< 0.01	1	1	0.03	0.01	1	1	0.03	0.03	1	1	0.08	0.1
30	9	1	0.24	0.15	224.45	< 0.01	1	1	0.01	0.01	1	1	0.02	0.02	1	1	0.06	0.1
30	10	1	0.02	0.18	263.71	< 0.01	1	1	0.01	0.01	1	1	0.01	0.01	1	1	0.05	0.07
40	6	1	542.45	0	5.25	0.01	0.97	1.17	0.17	0.04	0.99	0.99	0.17	0.12	0.99	0.99	0.77	0.74
40	7	1	116.57	0.01	14.58	< 0.01	1	1	0.06	0.03	1	1	0.09	0.08	1	1	0.34	0.54
40	8	1	129.03	0.01	20.78	0.01	1	1	0.06	0.02	1	1	0.05	0.06	1	1	0.21	0.49
40	9	1	2.33	0.12	261.7	< 0.01	1	0.96	0.04	0.01	1	1	0.02	0.03	1	1	0.13	0.37
40	10	1	0.2	0.39	804.49	< 0.01	1	1	0.01	0.01	1	1	0.01	0.02	1	1	0.1	0.26
50	6	0.9	600	0	5.37	0.01	0.96	1.03	0.29	0.07	0.99	0.99	0.29	0.12	0.99	0.99	3.26	1.89
50	7	1	545.39	0.01	11.74	< 0.01	0.99	0.99	0.18	0.04	0.99	0.99	0.2	0.1	0.99	0.99	1.99	1.52
50	8	1	382.01	0.01	16.94	0.01	1	1	0.1	0.02	1	1	0.06	0.09	1	1	0.6	1.23
50	9	1	5.96	0.03	78.9	< 0.01	1	1	0.02	0.01	1	1	0.03	0.05	1	1	0.32	0.8
50	10	1	0.07	0.33	804.69	< 0.01	1	1	0.01	0.01	1	1	0.02	0.03	1	1	0.21	0.5

Table 11: Computational results for the instances of (KIP) for $m_f \in \{3, 5\}$. For MibS and (SLR) (i.e., $k = 0$), we report the average runtime in seconds (column "Time"). For (KIP$_k$), $k \in \{1, 2, 3\}$, the average runtime in seconds for formulations (BP$_k$-DF) and (BP$_k$-Mix) are shown in columns "Time" and "ExtTime," respectively. For each solution approach, the ratios between the achieved bounds and the best upper bound, $\hat{\eta}_M^*$, reported by MibS (i.e., $\frac{\eta_k^*}{\hat{\eta}_M^*}$ and $\frac{\hat{\eta}_k^*}{\hat{\eta}_M^*}, k \in \{M, 0, 1, 2, 3\}$) are shown in the columns "ObjL" and "ObjU," respectively.

			Mibs [104]		$k=0$			$k=1$				$k=2$				$k=3$			
	n	r	ObjL	Time	ObjL	ObjU	Time	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime
	20	3	1	146.59	0.01	2.27	< 0.01	0.21	1.54	0.08	0.04	0.62	1.3	1.12	0.79	0.95	1.02	8.23	5.79
	20	4	1	162.38	0	2.49	< 0.01	0.37	1.6	0.13	0.05	0.79	1.22	1.64	0.66	0.99	1	16.81	3.76
	20	5	1	41.21	0	2.62	0.01	0.41	1.37	0.12	0.04	0.79	1.21	2.34	0.58	1	1	15.31	4.36
	20	6	1	20.52	0	3.45	< 0.01	0.75	1.43	0.13	0.05	0.98	1.06	2.03	0.35	1	1	9.48	1.83
	20	7	1	0.62	0.01	9.31	< 0.01	1	1.08	0.05	0.02	1	1	0.15	0.09	1	1	0.42	0.32
	30	3	0.61	600	0	1.85	0.01	0.19	1.43	0.12	0.07	0.52	1.2	36.65	4.33	0.92	1.05	501.1	71.14
	30	4	0.71	600	0	2.51	0.01	0.3	1.66	0.14	0.08	0.64	1.31	68.35	3.98	0.99	1.01	338.76	50.08
$m_f = 3$	30	5	0.78	564.6	0	3.41	0.01	0.59	1.7	0.19	0.09	0.91	1.13	60.18	1.67	1	1	239.47	19.13
	30	6	0.94	461.67	0	5.32	0.01	0.88	1.25	0.28	0.08	1	1	19.48	0.42	1	1	72.01	4.51
	30	7	1	138.96	0.02	22.67	0.01	1	1	0.14	0.07	1	1	1.15	0.1	1	1	7.63	0.7
	40	3	0.39	600	0	1.89	0.01	0.18	1.38	0.16	0.1	0.5	1.24	241.89	13.47	0.91	1.01	600	327.48
	40	4	0.45	600	0	2.4	0.01	0.31	1.51	0.28	0.12	0.64	1.28	365.95	12.06	0.97	1.01	600	240.46
	40	5	0.53	600	0	2.82	0.01	0.47	1.49	0.32	0.12	0.78	1.17	436.04	7.09	0.99	1	600	116.86
	40	6	0.71	560.78	0	5.79	0.01	0.93	1.21	0.6	0.11	1	1	86.17	1.45	1	1	231.03	17.82
	40	7	0.83	544.32	0	8.33	0.01	1	1	0.36	0.1	1	1	7.95	0.19	1	1	83.18	2.7
	20	3	1	213.32	0	1.83	0.01	0.19	1.48	0.1	0.07	0.53	1.32	2.03	2.51	0.93	1.08	20.75	32.92
	20	4	1	144.56	0	2.31	0.01	0.31	1.46	0.11	0.07	0.68	1.32	7.13	4.65	0.99	1	80.87	39.33
	20	5	1	60.71	0	2.97	< 0.01	0.45	1.51	0.12	0.08	0.84	1.14	11.56	2.2	1	1.01	53.88	16.48
	20	6	1	7.62	0	4.71	< 0.01	0.85	1.46	0.14	0.08	1	1	1.65	0.4	1	1	6.42	2.72
	20	7	1	0.75	0	24.21	< 0.01	0.96	1.11	0.06	0.04	1	1	0.41	0.12	1	1	1.07	1.21
	30	3	0.55	600	0	1.72	0.01	0.19	1.49	0.13	0.1	0.48	1.27	88.75	22.5	0.91	1.02	600	287.14
	30	4	0.63	600	0	2.39	0.01	0.31	1.53	0.19	0.13	0.62	1.26	323.6	14.51	0.97	1.01	574.77	238.13
$m_f = 5$	30	5	0.72	600	0	2.96	0.01	0.51	1.52	0.36	0.12	0.86	1.2	471.29	13.9	0.99	1	583.49	95.97
	30	6	0.89	496.44	0	4.37	0.01	0.87	1.28	0.49	0.12	0.98	1.03	169.73	1.75	1	1	248.91	20.62
	30	7	0.99	210.67	0	8.88	0.01	0.97	1.02	0.32	0.11	1	1	11.59	0.37	1	1	70.4	5.06
	40	3	0.33	600	0	1.78	0.02	0.17	1.4	0.4	0.13	0.38	1.22	574.95	85.51	0.61	1.02	600	600
	40	4	0.33	600	0	2.24	0.01	0.34	1.65	0.81	0.13	0.61	1.27	600	83.92	0.85	0.99	600	541.18
	40	5	0.45	600	0	2.77	0.01	0.53	1.49	1.27	0.13	0.79	1.14	569.92	25.57	0.96	0.97	600	421.23
	40	6	0.6	600	0	4.44	0.01	0.91	1.34	1.27	0.12	0.99	0.99	173.04	2.26	0.99	0.99	580.04	65.21
	40	7	0.8	581.89	0	7.12	0.01	1	1	1.17	0.12	1	1	37.96	0.48	1	1	380.96	19.53

Hence, we have the exponential growth of the number of constraints in (BP_k-DF-d) with the increase of k, which leads to the deterioration of the overall performance.

3.6.2 Bilevel Vertex Cover (BVC)

Given a graph $G = (N, E)$, the vertex cover problem is to find a subset of vertices whose total weight is as small as possible such that each vertex in the graph is either in this subset or connected to at least one vertex in this subset [41]. We consider its bilevel extension with interdiction constraints, referred to as the bilevel vertex cover (BVC) problem [11]. In BVC, the leader first removes vertices from N subject to some budgetary constraint, and then the follower solves the vertex cover problem. Formally, the BVC problem is stated as:

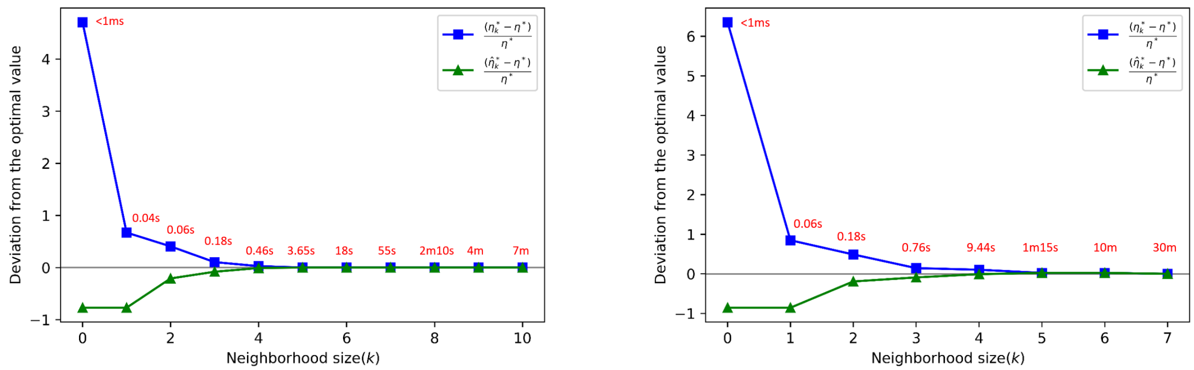
$$\begin{aligned} \max_{x,y} \; & \sum_{j=1}^{n} \alpha_j y_j \\ \text{s.t.} \; & \sum_{j=1}^{n} x_j \leq b, x \in \{0,1\}^n, \\ & y \in \arg\min_{\bar{y}} \{ \sum_{j=1}^{n} \beta_j \bar{y}_j : \; \bar{y} \in \mathcal{S}_{\text{VC}}(x) \}, \end{aligned} \tag{BVC}$$

where $n = |N|$, and

$$\mathcal{S}_{\text{VC}}(x) = \Big\{ y \in \{0,1\}^n : \; x_j + y_j \leq 1 \; \forall j \in N, \sum_{i \in N_j} y_i \geq 1 \; \forall j \in N \Big\},$$

where $N_j := \{i \in N : (i,j) \in E\} \cup \{j\}$ is the extended neighborhood of $j \in N$, that includes j itself. If $\alpha_j = \beta_j$ for all $j \in N$, then we say that the BVC problem is symmetric, and can be referred to as the vertex cover interdiction problem; otherwise, the BVC problem is asymmetric. Note that in the formulations (BP_k-DF) and (BP_k-Mix), a valid μ_j is the upper bound for the term $\sum_{i \in N_j} y_j$ for each $j \in N$. Thus, we trivially set $\mu_j = |N_j|$ in our experiments.

Experimental setup. Let (BVC_k) be the bilevel vertex cover problem with k-optimal follower. In the BVC problem, the leader maximizes her objective function, thus η_k^* and $\hat{\eta}_k^*$ provide valid upper and lower bounds for η^*, respectively. Denote by η_M^* and $\hat{\eta}_M^*$ the best



(a) $n = 15$, deg$= 8, b = 8$.

(b) $n = 20$, deg$= 10, b = 10$.

Figure 5: The average deviations from the optimal value of (BVC$_k$) for different k. The solution time of formulation (BP$_k$-DF) is shown in red.

upper and lower bounds reported by MibS at termination, respectively. In our experiments, we use $\hat{\eta}^*_M$ as the benchmark.

We randomly construct graphs with SNAP [71], in which the degree of each vertex is no less than $\lceil \frac{n}{2} \rceil$ with respect to the vertex size $n \in \{10, 15, 20, \ldots, 45, 50\}$. We refer to the minimum vertex degree in the graph as "deg". For each pair of the considered classes (i.e., the number of vertices n along with the specific value of deg and b), we report the average performance over 10 randomly constructed instances. The time limit for MibS is set to 30 minutes in order to avoid the out-of-memory error. The time limit for (BVC$_k$) is also set to 30 minutes.

Results and discussions. In Figure 5, we first depict the deviation of our proposed bounds from the true optimal objective function value η^* of (BVC) for different values of k. We evaluate the average performance of 10 randomly constructed symmetric (BVC) instances for each $n \in \{15, 20\}$. The solution time is indicated as the labels in red. Similar to Figure 4 for (KIP), we observe that the bounds provided by (BVC$_k$), $k \geq 1$, significantly outperform the quality of SLR-based bounds. Furthermore, our bounds converge rapidly to the optimal value η^* for relatively small values of k.

Table 12: Computational results for the instances of (BVC). For MibS and (SLR) (i.e., $k=0$), we report the average runtime in seconds (column “Time”). For (BVC_k), $k \in \{1,2,3\}$, the average runtime in seconds for formulations (BP_k-DF) and (BP_k-Mix) are shown in columns “Time” and “ExtTime,” respectively. For each solution approach, the ratios between the achieved bounds and $\hat{\eta}^*_M$ (i.e., $\frac{\hat{\eta}^*_k}{\hat{\eta}^*_M}$ and $\frac{\eta^*_k}{\hat{\eta}^*_M}$, $k \in \{M,0,1,2,3\}$) are reported in columns “ObjL” and “ObjU,” respectively.

			Mibs [104]		$k=0$			$k=1$				$k=2$				$k=3$			
n	deg	b	ObjU	Time	ObjL	ObjU	Time	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime	ObjL	ObjU	Time	ExtTime
Symmetric Objective																			
10	5	5	1	0.57	0.27	4.00	< 0.01	0.27	1.38	0.03	0.03	0.89	1.06	0.04	0.04	0.95	1.03	0.06	0.06
15	8	8	1	25.66	0.23	5.71	< 0.01	0.23	1.67	0.04	0.04	0.79	1.41	0.06	0.06	0.92	1.10	0.18	0.24
20	10	10	1	984.85	0.14	7.19	< 0.01	0.14	1.81	0.06	0.06	0.79	1.46	0.18	0.18	0.89	1.12	0.76	1.32
25	13	13	6.96	1800	0.14	9.47	< 0.01	0.14	2.30	0.06	0.06	0.79	1.66	0.35	0.38	1.08	1.37	2.35	4.52
30	15	15	10.75	1800	0.17	12.90	< 0.01	0.17	2.92	0.10	0.10	0.73	2.05	1.02	0.86	1.15	1.76	9.10	18.67
35	18	18	16.40	1800	0.16	18.49	< 0.01	0.16	3.77	0.25	0.17	0.97	2.56	2.32	2.20	1.17	2.03	43.49	71.31
40	20	20	20.33	1800	0.25	22.26	< 0.01	0.25	4.31	0.57	0.38	0.87	2.81	6.56	6.68	1.28	2.37	197.25	479.93
45	23	23	27.28	1800	0.18	29.08	< 0.01	0.18	5.31	0.75	0.57	0.98	3.39	12.91	14.10	1.70	2.83	737.55	1277.52
50	25	25	32.19	1800	0.17	33.86	< 0.01	0.17	5.86	1.13	0.98	0.83	3.94	31.58	36.76	1.63	2.94	1800	1800
Asymmetric Objective																			
10	5	5	1	0.54	0.50	3.12	< 0.01	0.50	1.07	0.03	0.03	0.78	1.04	0.04	0.04	0.98	1	0.07	0.06
15	8	8	1	24.75	0.37	3.59	< 0.01	0.37	1.11	0.04	0.04	0.53	1.06	0.07	0.07	0.69	1	0.13	0.16
20	10	10	1	925.88	0.46	4.35	< 0.01	0.46	1.19	0.05	0.05	0.62	1.06	0.13	0.13	0.66	1.02	0.61	0.80
25	13	13	4.30	1800	0.43	5.94	< 0.01	0.43	1.37	0.07	0.07	0.70	1.28	0.29	0.29	0.79	1.24	1.38	2.35
30	15	15	5.66	1800	0.54	6.78	< 0.01	0.54	1.53	0.1	0.09	0.79	1.40	0.68	0.58	0.89	1.29	4.98	10.27
35	18	18	7.27	1800	0.44	8.25	< 0.01	0.48	1.68	0.27	0.17	0.72	1.54	1.16	0.99	0.75	1.44	12.76	25.78
40	20	20	8.58	1800	0.55	9.39	< 0.01	0.55	1.84	0.55	0.34	0.68	1.63	2.98	2.80	0.81	1.47	72.02	112.00
45	23	23	9.79	1800	0.66	10.51	< 0.01	0.66	1.85	0.82	0.67	0.68	1.71	4.59	4.42	0.89	1.56	204.57	228.72
50	25	25	10.66	1800	0.61	11.26	< 0.01	0.61	1.97	1.07	0.93	0.62	1.86	8.18	8.10	0.85	1.70	684.02	656.31

Table 13: Computational results for the instances of (BVC$_k$) with different minimum vertex degrees for fixed $n = 20$ and $b = 10$. For MibS and (SLR) (i.e., $k = 0$), we report the average runtime in seconds (column "Time"). For (BVC$_k$), $k \in \{1, 2, 3\}$, the average runtime in seconds for formulations (BP$_k$-DF) and (BP$_k$-Mix) are shown in columns "Time" and "ExtTime," respectively. For each solution approach, the ratios between the achieved bounds and $\hat{\eta}^*_M$ (i.e., $\frac{\hat{\eta}^*_k}{\hat{\eta}^*_M}$ and $\frac{\eta^*_k}{\hat{\eta}^*_M}, k \in \{M, 0, 1, 2, 3\}$) are reported in columns "ObjL" and "ObjU," respectively.

			Mibs [104]	$k=0$			$k=1$				$k=2$				$k=3$			
n	deg	b	Time	ObjL	ObjU	Time	ObjL	ObjU	Time	Ext Time	ObjL	ObjU	Time	Ext Time	ObjL	ObjU	Time	Ext Time
Symmetric Objective																		
20	10	10	984.85	0.14	7.19	< 0.01	0.14	1.81	0.06	0.06	0.79	1.46	0.18	0.18	0.89	1.12	0.76	1.33
20	11	10	1245.49	0.15	7.09	< 0.01	0.15	1.72	0.04	0.04	0.70	1.32	0.16	0.17	0.87	1.11	0.66	1.09
20	12	10	1430.29	0.11	7.16	< 0.01	0.11	1.70	0.05	0.05	0.81	1.28	0.15	0.15	0.80	1.07	0.56	0.98
20	13	10	1543.51	0.12	7.57	< 0.01	0.12	1.67	0.05	0.05	0.72	1.32	0.11	0.11	0.94	1.10	0.48	0.89
20	14	10	1534.97	0.09	7.28	< 0.01	0.09	1.46	0.05	0.05	0.68	1.21	0.10	0.10	0.89	1.02	0.37	0.65
20	15	10	1529.98	0.08	7.53	< 0.01	0.08	1.35	0.04	0.04	0.74	1.18	0.09	0.09	0.98	1.01	0.26	0.46
Asymmetric Objective																		
20	10	10	925.88	0.46	4.35	< 0.01	0.46	1.19	0.05	0.05	0.62	1.06	0.13	0.13	0.66	1.02	0.60	0.79
20	11	10	1200.43	0.42	4.77	< 0.01	0.42	1.16	0.05	0.05	0.55	1.09	0.11	0.11	0.90	1.01	0.52	0.71
20	12	10	1383.42	0.43	5.18	< 0.01	0.43	1.23	0.05	0.05	0.70	1.19	0.10	0.10	0.95	1.04	0.40	0.62
20	13	10	1489.77	0.42	5.23	< 0.01	0.42	1.10	0.05	0.05	0.54	1.03	0.09	0.09	0.96	1	0.32	0.51
20	14	10	1463.03	0.38	5.81	< 0.01	0.38	1.15	0.05	0.05	0.55	1.06	0.1	0.10	0.92	1	0.27	0.42
20	15	10	1445.94	0.27	6.25	< 0.01	0.27	1.04	0.04	0.04	0.65	1.02	0.09	0.10	0.94	1.01	0.23	0.29

The numerical results for MibS and BVC$_k, k \in \{0, 1, 2, 3\}$, are reported for both symmetric and asymmetric objectives in Table 12. In particular, for MibS and (SLR) (i.e., $k = 0$), we report the average runtime in seconds (column "Time"). For BVC$_k, k \in \{1, 2, 3\}$, the average runtime in seconds for formulations (BP$_k$-DF) and (BP$_k$-Mix) are shown in columns "Time" and "ExtTime", respectively. For each solution approach, the ratios between the achieved bounds and $\hat{\eta}^*_M$ (i.e., $\frac{\hat{\eta}^*_k}{\hat{\eta}^*_M}$ and $\frac{\eta^*_k}{\hat{\eta}^*_M}, k \in \{M, 0, 1, 2, 3\}$) are reported in columns "ObjL" and "ObjU", respectively.

In Table 12, we observe that MibS can only handle the smallest instances within the computational limits. For instances with $n = 50$, deg$= b = 25$ and symmetric objectives, the average ratio between the best upper and lower bounds (i.e., $\frac{\eta^*_M}{\hat{\eta}^*_M}$) obtained by MibS at termination is 32.19, and the average ratio between the upper bound obtained by (SLR) and $\hat{\eta}^*_M$ (i.e., $\frac{\eta^*_0}{\hat{\eta}^*_M}$) is 33.86. The difference between these two ratios is less than 2, which suggests that there is no substantial progress achieved by MibS to reduce the optimality gap after extensive branching and cut generation.

On the other hand, all instances (except for $n = 50$, deg$= b = 25$ and symmetric objectives) are solved to optimality when using formulations (BP$_k$-DF) and (BP$_k$-Mix) for (BVC$_k$) with $k \in \{1, 2, 3\}$. The average ratio between the obtained upper bound and $\hat{\eta}^*_M$ is reduced from 33.86 to 5.86 for $k = 1$, and further to 2.94 for $k = 3$. As for the lower bound, the average ratio between the obtained lower bound of (BVC$_1$) and $\hat{\eta}^*_M$ does not improve over that of (SLR). It is not a surprising result as we can easily verify that (SLR) and (BVC$_1$) have identical optimal values of $x = 0$. However, when $k = 2$, this ratio increases from 0.17 to 0.83. Moreover, the ratio is further improved to 1.63 when $k = 3$, which implies that the lower bound obtained by (BVC$_3$) is better than the best lower bound reported by MibS. For the instances with asymmetric objectives, similar improvements can be observed for (BVC$_k$). Hence, we conclude that the bounds by (BVC$_k$) are superior to the SLR-based bounds, and the overall performances of (BVC$_k$) is better than MibS for larger instances.

With respect to the solution times for (BVC$_k$), one immediate observation is that the instances with asymmetric objectives are much easier for the solver. We also observe that the average computational times are very similar for the formulations (BP$_k$-DF) and (BP$_k$-Mix), for $k \in \{1, 2\}$. Also, the runtime of formulation (BP$_k$-DF) is relatively better than that of formulation (BP$_k$-Mix) for $k = 3$. Recall Proposition 3.9 that the LP relaxation of (BP$_k$-Mix) is stronger than that of (BP$_k$-DF). However, the strengthened constraints in (BP$_k$-Mix) are denser, which may result in more computational efforts required for solving its LP relaxation.

In our next set of experiments, we explore how the quality of bounds obtained by (BVC$_k$) depends on the minimum vertex degree in the graph. The corresponding results are presented in Table 13. Observe that the performance of MibS deteriorates when the minimum vertex degree increases. It is intuitive given that the graph density increases for larger values of the minimum vertex degree. On the other hand, we note that the quality of bounds obtained by BVC$_k, k \geq 1$, improves for larger minimum vertex degrees and with smaller computational times. These comparisons illustrate that our proposed bounding approach is capable of exploiting the problem inherent structure, which further supports our earlier results in Section 3.6.1.

3.6.3 Bilevel Minimum Spanning Tree (BMST)

In this section, we study the bilevel minimum spanning tree problem (BMST) to illustrate our results in Section 3.5. Two single-level formulations for BMST are developed based on (BP$_k$) for $k = 2$ in Section 3.6.3.1. The computational experiments are then conducted in Section 3.6.3.2.

In particular, we focus on the variant of the BMST problem considered in [99], which is described as follows: given an undirected graph $G = (N, E)$, the leader and the follower construct a spanning tree of graph G in a hierarchical manner. The leader first selects a subset of edges from among those in $E_L \subseteq E$. The follower then selects a set of edges from E that complete a spanning tree, according to their own objective function. Formally, the BMST problem can be stated as:

$$
\begin{aligned}
\eta^* = & \min_{x,y} \alpha(x + y) \\
\text{s.t. } & x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E_L, \\
& y \in \arg\min_{\bar{y}} \{\beta \bar{y} : \bar{y} \in \mathcal{S}_{\text{MST}}(x)\},
\end{aligned}
\tag{BMST}
$$

where we let $m_0 = |E_L|$, $m = |E|$, and

$$\mathcal{S}_{\text{MST}}(x) = \begin{Bmatrix} y \in \{0,1\}^m : x_{ij} + y_{ij} \leq 1 \quad \forall (i,j) \in E_L, \\ G[x+y] \text{ is a spanning tree of graph } \ G \end{Bmatrix},$$

where $G[x] := G[E_x] = (N, E_x)$ is the subgraph induced by edges in $E_x = \{(i,j) \in E : x_{ij} = 1\}$ for $x \in \{0,1\}^m$. We also define directed graph $\widetilde{G}[x] = (N, \mathcal{A}_{[x]})$, where $\mathcal{A}_{[x]} = \{(i,j),(j,i) : x_{ij} = 1\}$ for any $x \in \{0,1\}^m$.

We refer to the bilevel minimum spanning tree problem with a k-optimal follower as (BMST$_k$). Following our discussion in Section 3.5, the follower's feasible region $\mathcal{S}_{\text{MST}}(x)$ is the set of characteristic vectors of all maximal independent sets of a matroid. Therefore, by Theorem 3.5 we have (BMST) $\equiv$ (BMST$_k$) for $k \geq 2$. Next, we derive single-level MILP formulations for (BMST) based on (BMST$_2$) and its particular structure.

3.6.3.1 MILP Formulations

Based on Proposition 3.1, we have $\mathcal{T}^2 = \{w = (e_{i_0j_0} - e_{i_1j_1}) : \ \beta_{i_0j_0} < \beta_{i_1j_1}\}$. We then explore the optimality conditions for the lower-level problem of (BMST$_2$).

Proposition 3.10. *Let x be a given leader's decision. Then y is a follower's 2-optimal solution, i.e., $y \in \mathcal{R}_2(x)$ if and only if the following two conditions hold:*

(i) $y \in \mathcal{S}_{\text{MST}}(x)$;

(ii) *for any $w \in \mathcal{T}^2$, then $G[x+y+w]$ is not a spanning tree (i.e., either $x+y+w \notin \{0,1\}^m$ or $G[x+y+w]$ contains a cycle).*

Proof. The result follows directly from Proposition 3.1. ■

Thus, we can reformulate (BMST) as:

$$\begin{aligned} \eta^* = \min_{x,y} \ & \alpha(x+y) \\ \text{s.t. } & x_{ij} + y_{ij} \leq 1 \quad \forall (i,j) \in E_L, \\ & G[x+y] \text{ is a spanning tree of graph } \ G, \\ & G[x+y+w] \text{ is not a spanning tree } \quad \forall w = (e_{i_0j_0} - e_{i_1j_1}) \in \mathcal{T}^2, \\ & x \in \{0,1\}^{m_0}, y \in \{0,1\}^m. \end{aligned}$$

To formulate the condition that $G[x+y]$ is a spanning tree via a set of linear constraints, we apply the multi-commodity flow model [80]. Let $\mathcal{A} = \{(i,j),(j,i) : (i,j) \in E\}$ be the directed arcs that are constructed from E, i.e., each edge in E is cloned into two arcs with opposite directions. Let vertex u_0 in N be an arbitrary source node. Then we impose the following constraints:

$$\sum_{(i,j)\in E_L} x_{ij} + \sum_{(i,j)\in E} y_{ij} = n-1, \tag{3.4a}$$

$$Af^v = \begin{cases} 1, & \text{for vertex } u_0 \\ -1, & \text{for vertex } v \\ 0, & \text{otherwise} \end{cases} \quad \forall v \in N \setminus \{u_0\}, \tag{3.4b}$$

$$f^v_{ij} + f^v_{ji} \leq x_{ij} + y_{ij} \quad \forall (i,j) \in E, v \in N \setminus \{u_0\}, \tag{3.4c}$$

$$f^v_{ij} \geq 0 \quad \forall v \in N \setminus \{u_0\}, (i,j) \in \mathcal{A}, \tag{3.4d}$$

where A is the node-arc matrix of the directed graph $\widetilde{G} := (N, \mathcal{A})$.

To formulate the condition (ii) in Proposition 3.10, we first observe that if $x+y+w \notin \{0,1\}^n$, then it is clear that the condition holds; if $x+y+w \in \{0,1\}^n$ and $w = (e_{i_0j_0} - e_{i_1j_1}) \in \mathcal{T}^2$, then it implies that $w^\top y + ||w^-||_1 + |w|^\top x = 0$. Next, we use linear constraints to ensure that $G[x+y+w]$ is not a spanning tree.

Let $(i_0, j_0) \in E$ be such that $w = (e_{i_0j_0} - e_{i_1j_1}) \in \mathcal{T}^2$. Consider a shortest path problem from i_0 to j_0 in graph $\widetilde{G}[x+y+w]$, where the edge weight for (i_0, j_0) is set to n, and the weight for all other edges is set to 1. Observe that $G[x+y+w]$ is not a spanning tree if and only if the length of the shortest path from i_0 to j_0 in $\widetilde{G}[x+y+w]$ is strictly less than n. Therefore, to ensure condition (ii) in Proposition 3.10, we restrict the objective function of the shortest path problem to take values less than n, as follows:

$$Az^w = \begin{cases} 1, & \text{for vertex } i_0 \\ -1, & \text{for vertex } j_0 \\ 0, & \text{otherwise} \end{cases}, \tag{3.5a}$$

$$\sum_{(i,j)\in\mathcal{A}^w} z_{ij}^w + n z_{i_0 j_0}^w \leq n - 1 + w^\top y + ||w^-||_1 + |w|^\top x, \tag{3.5b}$$

$$0 \leq z_{ij}^w \leq \max\{0, x_{ij} + y_{ij} + w_{ij}\} \quad \forall (i,j) \in \mathcal{A}, \tag{3.5c}$$

Finally, we formalize the single-level MILP formulation for the BMST problem as:

$$\begin{aligned} \eta^* = \min_{x,y}\ & \alpha(x+y) \\ \text{s.t. }& (3.4), \\ & (3.5)\ \forall w \in \mathcal{T}^2, \\ & x_{ij} + y_{ij} \leq 1 \quad \forall (i,j) \in E_L, \\ & x \in \{0,1\}^{m_0}, y \in \{0,1\}^m. \end{aligned} \tag{BMST-1}$$

Although, the mixing-set structure is not evident in the above MILP formulation, the key idea behind the extended formulation derived in Section 3.4.3 can be similarly applied. Recall that additional variables and precedence constraints are introduced for the extended formulation (BP$_k$-Mix) of (BP$_k$). For the BMST problem, we can thus develop another MILP formulation that also employs the precedence conditions, as described next. Note that this technique is also exploited by Shi et al. [99] (see Section 5.3), but we highlight that this idea can be generalized to other variants of the bilevel minimum spanning tree problem. For the sake of completeness and to provide a self-contained narrative, we review the concepts in our context.

Proposition 3.11 ([99]). *Assume w.l.o.g. that $E = \{(i_k, j_k) : 1 \leq k \leq m\}$ is such that $\beta_{i_1 j_1} \leq \beta_{i_2 j_2} \leq \cdots \leq \beta_{i_m j_m}$ and let $y^{<\ell}$ be such that*

$$y_{i_k j_k}^{<\ell} = \begin{cases} y_{i_k j_k} & \textit{if } k < \ell \\ 0 & \textit{otherwise.} \end{cases}$$

for a given $y \in \{0,1\}^m$. Then for a given $x \in \mathcal{X}$, $y \in \mathcal{R}_2(x)$ if and only if

(i) $y \in \mathcal{S}_{\mathrm{MST}}(x)$; and

(ii) For $1 \leq \ell \leq m$, $y_{i_\ell j_\ell} = 1$ if and only if i_ℓ is disconnected from j_ℓ in $\widetilde{G}[x + y^{<\ell}]$.

In a fashion similar to that used in deriving constraints (3.5), we formulate condition (ii) in Proposition 3.11 by considering a shortest path problem in graphs $G[x + y^{<\ell} + e_{i_\ell j_\ell}]$ from vertex i_ℓ to vertex j_ℓ as follows, where the formulation on the right is the LP dual of the formulation on the left.

$$\min_{z^\ell} \sum_{(i,j)\in\mathcal{A}^\ell_{[x+y^{<\ell}]}} z^\ell_{ij} + nz^\ell_{i_\ell j_\ell} \qquad\qquad \max_{\pi^\ell} \pi^\ell_{i_\ell} - \pi^\ell_{j_\ell}$$

$$\text{s.t.}\;\; A_{[x+y^{<\ell}+e_{i_\ell j_\ell}]} z^\ell = \begin{cases} 1, & \text{for vertex } i_\ell \\ -1, & \text{for vertex } j_\ell \\ 0, & \text{otherwise} \end{cases}, \qquad \text{s.t.}\;\; \pi^\ell_i - \pi^\ell_j \leq 1 \quad \forall\, (i,j) \in \mathcal{A}_{[x+y^{<\ell}]},$$

$$z^\ell_{ij} \geq 0 \quad \forall\, (i,j) \in \mathcal{A}_{[x+y^{<\ell}+e_{i_\ell j_\ell}]}, \qquad\qquad \pi^\ell_{i_\ell} - \pi^\ell_{j_\ell} \leq n,$$

where $A_{[x+y^{<\ell}+e_{i_\ell j_\ell}]}$ is the node-arc matrix of graph $\widetilde{G}[x + y^{<\ell} + e_{i_\ell j_\ell}]$ and $\mathcal{A}_{[x+y^{<\ell}+e_{i_\ell j_\ell}]}$ is its associated set of arcs. Note that there does not exist a path from i_ℓ to j_ℓ in $\widetilde{G}[x+y^{<\ell}]$ if and only if the above shortest path problem has optimal objective function value of n. Therefore, we enforce the following constraints for condition (ii) in Proposition 3.11:

$$A^\ell z^\ell = \begin{cases} 1, & \text{for vertex } i_\ell \\ -1, & \text{for vertex } j_\ell \\ 0, & \text{otherwise} \end{cases}, \tag{3.6a}$$

$$z^\ell_{ij} + z^\ell_{ji} \leq x_{ij} + y_{ij} \quad \forall (i,j) \in \mathcal{A}^\ell, \tag{3.6b}$$

$$z^\ell_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{A}^\ell \cup (i_\ell, j_\ell), \tag{3.6c}$$

$$\pi^\ell_i - \pi^\ell_j \leq 1 + \mu(1 - y_{ij}) \quad \forall\, (i,j) \in \mathcal{A}^\ell, \tag{3.6d}$$

$$\pi^\ell_{i_\ell} - \pi^\ell_{j_\ell} \leq n - 1 + x_{i_\ell j_\ell} + y_{i_\ell j_\ell}, \tag{3.6e}$$

$$\pi^\ell_{i_\ell} - \pi^\ell_{j_\ell} \geq n - \mu(1 - y_{i_\ell j_\ell} + x_{i_\ell j_\ell}), \tag{3.6f}$$

$$z^\ell_{ij} + nz^\ell_{i_\ell j_\ell} = \pi^\ell_{i_\ell} - \pi^\ell_{j_\ell}, \tag{3.6g}$$

where A^ℓ is the node-arc incidence matrix of graph $\widetilde{G}^\ell = (N, \mathcal{A}^\ell)$ and $\mathcal{A}^\ell = \{(i,j),(j,i) : (i,j) \in E_L\} \cup \{(i_p, j_p), (j_p, i_p) : 1 \leq p < \ell\}$, and μ is sufficiently large, e.g., n.

Based on the above, we provide another MILP formulation for BMST as follows:

$$\begin{aligned}\eta^* = \min_{x,y}\ & \alpha(x+y) \\ \text{s.t. } & (3.4), \\ & (3.6)\ \forall \ell = 1,\ldots,m, \\ & x_{ij} + y_{ij} \leq 1 \quad \forall (i,j) \in E_L, \\ & x \in \{0,1\}^{m_0}, y \in \{0,1\}^m.\end{aligned} \tag{BMST-2}$$

All in all, the MILP formulations (BMST-1) and (BMST-2) are derived based on the local optimality conditions in Propositions 3.10 and 3.11, respectively. We note that (BMST-2) is similar to the formulation proposed in [99]. With respect to the latter, we need to point out the following observations.

(i) The only difference between (BMST-2) and the model in [99] is how to formulate condition (i) in Proposition 3.11 (i.e., $G[x+y]$ is a spanning tree of G). Since the considered version of (BMST) is to construct a spanning tree by joint actions of both decision-makers, the condition in Proposition 3.11 (i) is implied by Proposition 3.11 (ii) and a new condition that $G[x]$ does not contain a cycle. The formulation in [99] adopts the latter approach to reformulate (BMST), which also allows to reduce the number of constraints in comparison to the formulation in (BMST-2). However, we observe in our experiments (not reported here) that the formulation (BMST-2) is stronger than the one in [99], as the LP relaxation of (BMST-2) always provides a tighter lower bound.

(ii) The modeling approach discussed in this section can be applied to other variants of (BMST), e.g., the minimum edge blocker spanning tree problem [112], the minimum spanning tree interdiction problem [38], where the leader removes the edges in the graph to maximize the weight of follower's minimum spanning tree. Under this setting, the condition (i) in Propositions 3.10 and 3.11 is that $G[y]$ is a spanning tree of G. Thus, the constraints used in the formulation from [99] are not applicable, while the formulations (BMST-1) and (BMST-2) can be easily extended with slight modifications.

Finally, Shi et al. [99] discuss an efficient preprocessing procedure to substantially reduce the size of the formulation. We note that their procedure can also be applied to our formulations (BMST-1) and (BMST-2); we omit its discussion for brevity.

3.6.3.2 Computational Results for (BMST)

We now computationally compare the MILP formulations (BMST-1) and (BMST-2).

Experimental setup. We use the graphs from the test set **B** in [64]. The test set **B** contains 18 graphs with $50-100$ number of vertices. We generate our test instances similar to the procedure in [99]. Specifically, for each graph in **B**, we randomly generate the edges in E_L with a specific fractional value $\rho \in \{0.05, 0.1, 0.15\}$. Here, ρ denotes the ratio between the number of edges in E_L and in E, i.e., $\rho := \frac{|E_L|}{|E|}$.

The follower's and leader's edge weight is generated through $\beta_{ij} = w_{ij}r_{ij}$ and $\alpha_{ij} = w_{ij}(1 - r_{ij})$ for each $(i, j) \in E$, respectively; where w_{ij} is the original edge weight from graphs in **B** and proportion r_{ij} is uniformly generated from interval $[0, 1]$. The average performance is reported over 10 random instances for each pair of graph and ρ. We set the time limit to one hour.

Results and discussion. The computational results for the MILP formulations (BMST-1) and (BMST-2) are reported in Table 14. For each graph in **B**, the number of vertices and edges are denoted in the columns "$|N|$" and "$|E|$", respectively. For each formulation, we report the average solver's runtime in the column "Time". We also report the integrality gap in the column "IG (%)", which is computed by

$$\text{IG}(\%) := \frac{(\eta^* - \eta_{LP})}{\eta^*} \times 100,$$

where η^* and η_{LP} denote the optimal objective function value of the formulation and its LP relaxation, respectively.

As expected, the BMST instances become more difficult as we increase of the number edges controlled by the leader (i.e., larger value of ρ) and the graph density. However, as we can see in Table 14, the average integrality gaps of both MILP formulations are very close and rather small, typically, under 5% for all instances even with $\rho = 0.15$. Moreover, all of the tested instances can be solved to optimality within the time limits when using both formulations. In particular, we observe that the formulation (BMST-2) based on the precedence edge orders performs better than (BMST-1). It usually requires a few seconds for the

Table 14: Computational results for the instances of (BMST). For each graph in **B**, the number of vertices and edges are denoted in the columns "$|N|$" and "$|E|$", respectively. For each formulation, we report the average solver's runtime in the column "Time". Column "IG (%)" reports the integrality gap, which is computed by IG(%) $:= \frac{(\eta^* - \eta_{LP})}{\eta^*} \times 100$, where η^* and η_{LP} denote the optimal objective function value of the formulation and its LP relaxation, respectively.

			$\rho = 0.05$				$\rho = 0.1$				$\rho = 0.15$			
			(BMST-1)		(BMST-2)		(BMST-1)		(BMST-2)		(BMST-1)		(BMST-2)	
Ins.	$\|N\|$	$\|E\|$	IG (%)	Time	IG (%)	Time	IG (%)	Time	IG (%)	Time	IG (%)	Time	IG (%)	Time
b1	50	63	0	0.04	0	0.02	0	0.08	0	0.04	0	0.14	0	0.04
b2	50	63	0	0.03	0	0.02	0	0.04	0	0.03	0.52	0.23	0.52	0.06
b3	50	63	0	0.04	0	0.02	0	0.12	0	0.04	0.83	0.13	0.83	0.04
b4	50	100	0.16	0.07	0.16	0.03	0.33	0.22	0.33	0.06	1.87	0.62	1.86	0.10
b5	50	100	0	0.07	0	0.04	0.68	0.24	0.60	0.06	2.65	0.73	2.44	0.14
b6	50	100	0	0.09	0	0.04	0.72	0.28	0.53	0.08	2.52	0.67	2.42	0.12
b7	75	94	0	0.16	0	0.07	0	0.34	0	0.08	0.50	1.04	0.50	0.13
b8	75	94	0	0.07	0	0.05	0.06	0.30	0.06	0.08	0.15	0.85	0.12	0.14
b9	75	94	0	0.13	0	0.06	0	0.47	0	0.10	0.14	0.81	0.14	0.13
b10	75	150	0.16	0.30	0.16	0.10	0.73	0.96	0.70	0.18	2.43	7.38	2.39	0.45
b11	75	150	0.13	0.27	0.13	0.09	1.11	1.32	1.02	0.21	3.13	7.31	2.90	0.52
b12	75	150	0.06	0.26	0.06	0.09	0.71	1.35	0.68	0.19	2.50	6.64	2.48	0.48
b13	100	125	0.08	0.34	0.08	0.14	0.04	0.90	0.04	0.15	0.61	2.72	0.61	0.33
b14	100	125	0	0.26	0	0.11	0.13	0.91	0.13	0.16	0.25	1.78	0.25	0.20
b15	100	125	0	0.30	0	0.11	0	0.67	0	0.13	0.10	2.14	0.10	0.23
b16	100	200	0.30	1.05	0.26	0.21	1.27	4.98	1.23	0.50	3.92	201.64	3.73	1.77
b17	100	200	0.29	1.00	0.29	0.22	0.83	3.57	0.79	0.42	1.81	11.15	1.69	0.84
b18	100	200	0.05	0.43	0.05	0.14	1.39	3.11	1.32	0.38	5.12	139.41	5.05	2.22

solver to handle (BMST-2), while (BMST-1) needs more than 5 minutes for large graphs and ρ. These results suggest that our formulations (BMST-1) and (BMST-2) are extremely tight, and the MILP formulation (BMST-2) might be more effective for solving the BMST problem.

3.7 Concluding Remarks

We considered mixed integer bilevel linear optimization problems in which the follower's decision variables are all binary. In response to the leader's decision, the proposed framework assumes that the follower does not have sufficient computational capabilities to obtain globally optimal solutions but instead implements a locally optimal solution. To capture the local optimality requirement we use the concept of k-optimality, where k is some predetermined neighborhood size of a given 0–1 vector. That is, $k = 0$ implies that the follower's objective function is completely ignored (also known as the single-level relaxation of the original bilevel problem), while $k = n$ corresponds for the fully-rational follower, who solves the lower-level problem to global optimality.

Under the assumption that the follower is optimistic, our framework naturally provides a hierarchy of upper and lower bounds for the standard bilevel optimization problem, where the follower is fully rational. To compute these bound for any fixed k, we develop single-level formulations, which can be solved by off-the-shelf solvers. In our extensive computational study the proposed bounds converge to the optimal objective function values of bilevel problems for reasonably small values of k. Moreover, the proposed framework provides lower and upper bounds of substantially better quality than those based on the widely used single-level relaxation method. Hence, our framework can be embedded into exact solvers—in particular, those that rely on single-level relaxations.

Our framework can also be used for solving classes of bilevel problems, in which local optimality of a follower's decision (within some sufficiently "small" neighborhood) implies its global optimality for the lower-level problem. As an example, in this chapter we exploit this idea to reformulate a general class of bilevel matroid problems as equivalent linear MILPs.

One of our framework's limitations is that the sizes of the proposed MILP formulations with large k prevent us for solving large-scale instances. Thus, improving the scalability of our approaches (e.g., by designing more advanced exact and approximate solution methods) is another important direction for future research.

Finally, if the follower is pessimistic, then our framework results in a tri-level optimization problem, which can be formalized as:

$$\begin{aligned} \max_{x} \min_{y} \ & \alpha^1 x + \alpha^2 y \\ \text{s.t. } & (x, y) \in \mathcal{S}, \\ & y \in \mathcal{R}_k(x) = \{y \in \mathcal{S}(x) : \beta y \geq \beta \hat{y} \ \forall \hat{y} \in \mathcal{N}_k(y) \cap \mathcal{S}(x)\}. \end{aligned}$$

Note that the ideas behind our MILP reformulations can be directly extended to represent $\mathcal{R}_k(x)$ via linear constraints. It implies that the above tri-level optimization problem can be reduced to a bilevel max-min problem. To solve the latter, development of more advanced solution strategies provides an interesting topic for further research.

4.0 Sequence Independent Lifting for a Set of Submodular Maximization Problems

An extended abstract of this chapter appeared in the proceedings of the 21st Conference on Integer Programming and Combinatorial Optimization (IPCO 2020) [98][1].

4.1 Motivation

Given a ground set $N = \{1, \ldots, n\}$, we consider a submodular maximization problem in the form:

$$\max_{S \subseteq N} \{f(S) : \; S \in \mathcal{I}\}, \tag{4.1}$$

where $\mathcal{I}$ is a collection of subsets of N, and $f : 2^N \to \mathbb{R}$ is a real-valued, submodular set function. Let $\rho_j(S) = f(S \cup \{j\}) - f(S)$ for $S \subseteq N$ and $j \in N \setminus S$. Function f is said to be *submodular* if $\rho_j(S) \geq \rho_j(T)$ for all subsets $S \subseteq T \subseteq N$ and for any $j \in N \setminus T$. We note that this property is often referred to as the law of diminishing returns; see, e.g., [65].

In this chapter, we are primarily interested in solving a class of the submodular maximization problems via mixed-integer programming (MIP) approaches. Specifically, we focus on a class of submodular functions that are represented by a concave function composed with a affine function [2], i.e.,

$$f(S) = g(a(S) + b), \tag{4.2}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a concave function, $b \in \mathbb{R}$, vector $a \in \mathbb{R}^n$ consists of components that are either all nonnegative or all nonpositive, and $a(S) := \sum_{j \in S} a_j$. The same sign of the components of a is a necessary condition to ensure that f is submodular; see [2]. For convenience, in the remainder of the chapter we assume that $b = 0$ and $a \in \mathbb{R}^n_+$. The considered class of functions f is, in fact, very flexible and has been widely used in a number of contexts, in-

[1] Reprinted by permission from Springer Nature: Sequence independent lifting for the set of submodular maximization problem. Shi, X., Prokopyev, O. A., & Zeng B.. In International Conference on Integer Programming and Combinatorial Optimization, pp. 378-390. Springer, Cham, ©2020.

cluding utility maximization [2, 72, 96], data summarization [33, 34, 106, 111], combinatorial multi-armed bandits [26, 92, 124], and multi-class queuing systems [53, 102], etc.

Formally, let binary vector x be the incidence vector for subsets of N. Then submodular maximization problem (4.1) can be written as:

$$\max\{w : \ w \leq f(x), x \in \mathcal{X} \subseteq \{0,1\}^n\}, \tag{4.3}$$

where $\mathcal{X}$ is the feasible region of x with respect to $\mathcal{I}$. In this chapter, we study the corresponding mixed-integer submodular maximization set given by:

$$P = \{(w,x) \in \mathbb{R} \times \{0,1\}^n : \ w \leq g(a^T x), x \in \mathcal{X}\}. \tag{4.4}$$

In the remainder of the chapter if $\mathcal{X} = \{0,1\}^n$, then we refer to P as P_0. Note that we can remove the assumption on the signs of the components of a when $\mathcal{X} = \{0,1\}^n$ as x_j can be replaced by $1 - x_j$ for all j such that $a_j < 0$. Therefore, the valid inequalities developed for the convex hull of P_0 (i.e., $\text{conv}(P_0)$ with $\mathcal{X} = \{0,1\}^n$) and $a \in \mathbb{R}^n_+$ are also applicable for the convex hull of P (i.e., $\text{conv}(P)$) with general $\mathcal{X}$ and $a \in \mathbb{R}^n$.

The convex hull of P is a polyhedron since P is a union of a finite number of rays with the same directions. Nevertheless, problem (4.1), with f as in (4.2), is NP-hard; see, e.g., a discussion in [2]. To understand the mathematical structure of (4.4), Nemhauser and Wolsey [86] propose an approach with exponentially many submodular inequalities to formulate P as a mixed-integer linear program. However, its linear programming relaxation is quite weak and the traditional branch-and-bound methods are often ineffective [2]. Hence, the submodular maximization problem (4.1) remains challenging from the computational perspective.

To address this challenge, Ahmed and his colleagues obtained several pioneering results that provide a better understanding of the mathematical structure of P. When $\mathcal{X} = \{0,1\}^n$ and g is strictly increasing, concave and differentiable, Ahmed and Atamtürk [2] employ the lifting technique to derive the first set of strong lifted inequalities for $\text{conv}(P_0)$. Their study is mostly focused on a continuous relaxation of the lifting function as it allows for application of the Karush–Kuhn–Tucker (KKT) conditions to derive its subadditive approximation. The numerical results in [2] demonstrate that this approximate lifting is computationally effective. In a subsequent paper, Yu and Ahmed [122] adopt this approximation idea to study set

P, where $\mathcal{X}$ involves exactly one additional knapsack constraint. In particular, the authors consider its cardinality relaxation, i.e., the knapsack constraint is replaced by a cardinality constraint, and extend the subadditive approximate lifting developed in [2] to handle this somewhat more complicated case.

Inspired by the fundamental results in [2, 122], in this chapter, we revisit the sequence independent lifting and its multidimensional extensions for $\text{conv}(P)$. After developing a new class of subadditive functions, we recognize that the lifting function of $\text{conv}(P_0)$ given in [2] is naturally subadditive. We believe that this new finding is of a significant value for further systematic study of the submodular maximization problem and the associated set P.

Specifically, our technical results and contributions in this chapter can be summarized as follows. First, instead of assuming that g is strictly concave, increasing and differentiable as in [2, 122], we only assume that g is concave to ensure its submodularity. As mentioned earlier, in [2, 122] the increasing and differentiability properties are exploited for application of the KKT conditions in order to derive the closed-form subadditive approximation of the lifting function. Our approach does not rely on the KKT conditions. Clearly, with those key assumptions removed, a much broader class of concave functions that does not carry those properties can be considered.

Second, as an immediate consequence of the above generalization, piecewise linear functions g can be analyzed. In particular, it can be shown that the well-known mixed 0-1 knapsack set is a special case of set P as in (4.4). Consider the mixed 0-1 knapsack set K [83] given by:

$$K = \{(\pi, x) \in \mathbb{R} \times \{0,1\}^n : a^T x \leq b + \pi, \pi \geq 0, x \in \mathcal{X}\}.$$

Define variable $w = -\pi$, and concave function $g(z) = \min\{0, b - z\}$. Then set K can be represented in the form of (4.4). We believe that these connections between the mixed 0-1 knapsack sets and submodular sets are novel and rather important, given that a number of computationally effective results have been derived in the literature for the mixed 0-1 knapsack sets. Furthermore, it is known [83] that the mixed 0-1 knapsack set K can be viewed as a relaxation of the popular single-node flow set, given by

$$F = \{(x, y) \in \{0,1\}^n \times \mathbb{R}^n_+ : \sum_{j \in N^+} y_j - \sum_{j \in N^-} y_j \leq b, y_j \leq a_j x_j \; \forall j \in N, x \in \mathcal{X}\},$$

where $N = N^+ \cup N^-$. This result immediately implies that the valid inequalities for P are valid for the flow set F. Note that the connections between the submodularity and the flow models are also considered by Wolsey [115] and Atamtürk et al. [6].

Third, we develop several results to support lifting operations in the context of submodular optimization. In particular, new results on multidimensional sequence independent (SI) lifting are derived, and a new type of subadditive functions to ensure SI lifting is constructed and verified. The latter result generalizes the existing ones in the literature.

Finally, in addition to strengthening the existing polyhedral results for P_0 we also derive new interesting results when P has a more involved structure. Specifically, for P_0, i.e., when $\mathcal{X} = \{0,1\}^n$, we strengthen the results in [2, 122] and present two family of facets for $\text{conv}(P_0)$. Also, the connections with the mixed 0-1 knapsack and the single-node flow sets highlighted above allow us to unify the existing results in [83] by deriving valid inequalities for $\text{conv}(K)$ and $\text{conv}(F)$. Moreover, we consider a somewhat more involved set P, where $\mathcal{X}$ contains multiple disjoint cardinality constraints; such $\mathcal{X}$ is often encountered in the discrete optimization literature. For its convex hull, denoted by $\text{conv}(P_{MC})$, facet-defining and strong valid inequalities are derived based on the multidimensional lifting function and its strongest subadditive approximation, respectively. Again, these inequalities extend the results for the mixed 0-1 knapsack set and the single-node flow set with disjoint cardinality constraints.

An extended abstract of this chapter can be found in the Proceedings of the IPCO 2020 conference [98]. This full version includes more detailed proofs of our results, a new family of strong valid inequalities for $\text{conv}(P)$ when $\mathcal{X}$ contains disjoint cardinality constraints, and a new separation algorithm for its implementation. Furthemore, this chapter also reports substantially more extensive computational experiments.

The remainder of the chapter is organized as follows. In Section 4.2, we discuss basic properties of $\text{conv}(P)$ and provide a brief overview of the sequential and sequence independent lifting. Also, a new class of subadditive functions is described, which is then exploited in derivations of our subsequent results. In Section 4.3, we consider one-dimensional sequential lifting for $\text{conv}(P_0)$; furthermore, two family of lifted inequalities are shown to be facet-defining for $\text{conv}(P_0)$. In Section 4.4, we discuss multidimensional sequential lifting for $\text{conv}(P)$ when $\mathcal{X}$ involves disjoint cardinality constraints. One family of lifted inequalities is shown to be facet-defining; another family of strong valid inequalities is proposed through a subadditive approximation of the associated lifting function. Our computational study is presented in Section 4.5. Finally, we summarize our results and conclude the chapter in Section 4.6.

Additional notation. Given a positive integer k, let $[k] := \{1, 2, \ldots, k\}$. Denote by $\mathbf{e}_k$ a vector with 1 in its kth component, and 0 in the others. Given a vector $a \in \mathbb{R}^n$ and a set $S \subseteq N$, we define $a(S) := \sum_{j \in S} a_j$; denote by x_S the corresponding incidence vector of S such that $x_j = 1$ for $j \in S$, and $x_j = 0$, otherwise. We use $|S|$ to denote the cardinality of set S. For any set $S \subseteq N$, we let $\bar{S} := N \setminus S$. Finally, to simplify the notation we use $S \cup j$ and $S \setminus j$ to denote the union and subtraction of set S and a singleton set $\{j\}$, respectively.

4.2 Technical Preliminaries

In Section 4.2.1, we consider some basic properties and facet-defining inequalities of $\text{conv}(P)$. We then briefly review the lifting process, including both sequential lifting and SI lifting in Section 4.2.2. More importantly, a new class of piecewise concave subadditive functions is developed in Section 4.2.3, which is then exploited to derive our subsequent results in Section 4.3.

4.2.1 Basic Properties

Given a partition of N, $\{N_i\}_{i=1}^r$ (i.e., $\cup_{i=1}^r N_i = N$ and $N_i \cap N_j = \emptyset$ for any $i \neq j$), and a set of positive integers $d_1, \ldots, d_r$, the mixed-integer set of the submodular maximization problem with multiple disjoint cardinality constraints is formally given by:

$$P_{MC} = \{(w, x) \in \mathbb{R} \times \{0, 1\}^n : w \leq g(a^T x), \sum_{j \in N_i} x_j \leq d_i \; \forall i \in [r]\}.$$

Note that P_0 is a special case of P_{MC} when $r = n$ and $N_j = \{j\}, d_j = 1$ for all $j \in N$. Below we summarize several basic properties for $\text{conv}(P_{MC})$ following [2, 86].

Proposition 4.1. *The following statements hold for conv*(P_{MC})*:*

(*i*) *conv*(P_{MC}) *is full-dimensional.*

(*ii*) *Inequality* $x_j \geq 0$ *defines a facet of conv*(P_{MC}) *for any* $j \in N$.

(*iii*) *Inequality* $x_j \leq 1$ *defines a facet of conv*(P_{MC}) *for any* $j \in N$ *if* $d_i \geq 2$ *for* $i \in [r]$ *such that* $j \in N_i$.

(*iv*) *Inequality* $w \leq f(\emptyset) + \sum_{j \in N} \rho_j(\emptyset) x_j$ *defines a facet of conv*(P_{MC}).

(*v*) *Inequality* $w \leq f(N) - \sum_{j \in N} \rho_j(N \setminus j)(1 - x_j)$ *defines a facet of conv*(P_{MC}) *if* $|N_i| \leq d_i$ *for all* $i \in [r]$.

Denote by relax(P) the continuous relaxation of P, in which the 0-1 binary restriction on x_j is simply replaced by $0 \leq x_j \leq 1$ for all $j \in N$. Without applying the KKT conditions (recall our earlier discussion in Section 4.1), we next extend Corollary 1 in [2] derived for relax(P_0) to the case of a general concave function g for relax(P_{MC}).

Proposition 4.2. *If* (w', x') *is an extreme point of relax*(P_{MC})*, then* $w' = g(a^T x')$ *and the following conditions are satisfied:*

(*i*) *if* $0 < \sum_{j \in N_i} x'_j < d_i$ *for some* $i \in [r]$*, then there is at most one fractional component* x'_j *for* $j \in N_i$*;*

(*ii*) *there is at most one* $i \in [r]$ *such that* $0 < \sum_{j \in N_i} x'_j < d_i$ *and fractional components of* x' *exist in* N_i*;*

(*iii*) *if* $\sum_{j \in N_i} x'_j = d_i$ *for some* $i \in [r]$*, then there are at most two fractional components* x'_j *for* $j \in N_i$.

Proof. It is clear that $w' = g(a^T x')$. We first prove condition (i). Suppose there exist $1, 2 \in N_i$ such that x'_1, x'_2 are fractional. If either $a_1 = 0$ or $a_2 = 0$, then we can easily verify that (w', x') is not an extreme point with the assumption. If $a_1, a_2 > 0$, then let

$$\delta = \min\{x'_1, 1 - x'_1, \frac{d_i - \sum_{j \in N_i} x'_j}{|a_1/a_2 - 1|}, \frac{a_2}{a_1} x'_2, \frac{a_2}{a_1}(1 - x'_2)\} > 0.$$

We consider (w^1, x^1) and (w^2, x^2) defined as $w^1 = w^2 = w'$ and

$$x^1_j = \begin{cases} x'_1 + \delta & j = 1, \\ x'_2 - \frac{a_1}{a_2}\delta & j = 2, \\ x'_j & \text{otherwise}, \end{cases} \qquad x^2_j = \begin{cases} x'_1 - \delta & j = 1, \\ x'_2 + \frac{a_1}{a_2}\delta & j = 2, \\ x'_j & \text{otherwise}. \end{cases} \tag{4.5}$$

It is easy to verify that (w^1, x^1) and (w^2, x^2) are feasible solutions of relax(P_{MC}). Furthermore, $(w', x') = \frac{1}{2}(w^1, x^1) + \frac{1}{2}(w^2, x^2)$, which leads to a contradiction with the assumption that (w', x') is an extreme point.

To prove condition (ii), suppose there exist $i_1, i_2 \in [r]$ such that $0 < \sum_{j \in N_{i_1}} x'_j < d_{i_1}$, $0 < \sum_{j \in N_{i_2}} x'_j < d_{i_2}$, and $1 \in N_{i_1}$, $2 \in N_{i_2}$, and x'_1 and x'_2 are fractional. Let

$$\delta = \min\{x'_1, 1 - x'_1, d_{i_1} - \sum_{j \in N_{i_1}} x'_j, \frac{a_2}{a_1} x'_2, \frac{a_2}{a_1}(1 - x'_2), \frac{a_2}{a_1}(d_{i_2} - \sum_{j \in N_{i_2}} x'_j)\} > 0.$$

The contradiction is obtained by constructing (w^1, x^1) and (w^2, x^2) as in (4.5).

To prove condition (iii), suppose there exist $1, 2, 3 \in N_i$ such that x'_1, x'_2, x'_3 are fractional and $a_1 \geq a_2 \geq a_3$. Assume $a_2 = \lambda a_1 + (1 - \lambda) a_3$ for $\lambda \in [0, 1]$. Let

$$\delta = \min\{\frac{x'_1}{\lambda}, \frac{1 - x'_1}{\lambda}, x'_2, 1 - x'_2, \frac{x'_3}{1 - \lambda}, \frac{1 - x'_3}{1 - \lambda}\} > 0.$$

We consider (w^1, x^1) and (w^2, x^2) defined as $w^1 = w^2 = w'$ and

$$x^1_j = \begin{cases} x'_1 + \lambda\delta & j = 1, \\ x'_2 - \delta & j = 2, \\ x'_3 + (1 - \lambda)\delta & j = 3, \\ x'_j & \text{otherwise}, \end{cases} \qquad x^2_j = \begin{cases} x'_1 - \lambda\delta & j = 1, \\ x'_2 + \delta & j = 2, \\ x'_3 - (1 - \lambda)\delta & j = 3, \\ x'_j & \text{otherwise}. \end{cases}$$

We can verify that (w^1, x^1) and (w^2, x^2) are feasible for relax(P_{MC}). Furthermore, $(w', x') = \frac{1}{2}(w^1, x^1) + \frac{1}{2}(w^2, x^2)$, which leads to a contradiction with the assumption that (w', x') is an extreme point. ■

Corollary 4.1. *Any extreme point of relax(P_0) has at most one fractional component x_j, $j \in N$.*

Proof. Recall that P_0 is reducible to P_{MC} by setting $r = n, N_j = \{j\}$, and $d_j = 1$ for all $j \in N$. The result directly follows from Proposition 4.2(ii). ■

4.2.2 Sequential and Sequence Independent Lifting

Consider a general set, $\mathcal{X}$, in the form $\mathcal{X} = \{x \in \{0,1\}^n : Bx \leq d\}$ and two disjoint subsets $S_0, S_1 \subseteq N$. If we fix the variables with the indices in S_0 and S_1 to 0 and 1, respectively, then P becomes a low-dimensional set, which we denote as:

$$P(S_0, S_1) = \{x \in P : x_j = 0\ \forall j \in S_0, x_j = 1\ \forall j \in S_1\}.$$

Suppose an inequality

$$w \leq \alpha_0 + \sum_{j \in N \setminus (S_0 \cup S_1)} \alpha_j x_j \tag{4.6}$$

is valid for $\text{conv}(P(S_0, S_1))$. The goal of lifting [86] is to determine coefficients $\alpha_j \in \mathbb{R}$ for $j \in S_0 \cup S_1$ such that

$$w \leq \alpha_0 + \sum_{j \in N \setminus (S_0 \cup S_1)} \alpha_j x_j + \sum_{j \in S_0} \alpha_j x_j + \sum_{j \in S_1} \alpha_j (1 - x_j)$$

is a valid inequality for $\text{conv}(P)$. We refer to (4.6) as the *seed inequality*.

For convenience, we discuss next the case of $S_0 = \bar{S}$ and $S_1 = \emptyset$ for some $S \subseteq N$; recall our notation that $\bar{S} := N \setminus S$. Suppose $\bar{S} = \{1, \ldots, \bar{s}\}$, and without loss of generality, assume that the lifting sequence is $1, 2, \ldots, \bar{s}$. Then for each $\ell \in \bar{S}$, the lifting coefficient α_ℓ for $\ell = 1, \ldots, \bar{s}$ can be derived from the function:

$$\begin{aligned}
\zeta_\ell \binom{z}{\mathbf{u}} = \max_{w,x}\ & w - \sum_{j \in S} \alpha_j x_j - \sum_{j=1}^{\ell-1} \alpha_j x_j - \alpha_0 \\
\text{s.t. } & w \leq g(\sum_{j \in S} a_j x_j + \sum_{j=1}^{\ell-1} a_j x_j + z), \\
& \sum_{j \in S} B_j x_j + \sum_{j=1}^{\ell-1} B_j x_j \leq d - \mathbf{u}, \\
& x_j \in \{0,1\} \quad \forall j \in S \cup [\ell - 1],
\end{aligned}$$

where $z \in \mathbb{R}_+$, B_j is the jth column vector of B and $\mathbf{u}$ has the same dimension as d. We set $\zeta_\ell = -\infty$ if the above problem is infeasible. Observe that function ζ_ℓ is nondecreasing in ℓ, that is $\zeta_\ell \binom{z}{\mathbf{u}} \leq \zeta_k \binom{z}{\mathbf{u}}$ for any $k > \ell$.

Denote by $\zeta \binom{z}{\mathbf{u}} = \zeta_1 \binom{z}{\mathbf{u}}$ the *lifting function* of the seed inequality (4.6) for P. Note that if $\mathcal{X} = \{0,1\}^n$ (i.e., $B = 0, d = 0$), then the lifting function ζ is one-dimensional.

Proposition 4.3. *The inequality*

$$w \le \alpha_0 + \sum_{j\in S}\alpha_j x_j + \sum_{\ell\in\bar{S}}\alpha_\ell x_\ell \tag{4.7}$$

is valid for conv(P) *if* $\alpha_\ell \ge \zeta_\ell\binom{a_\ell}{B_\ell}$ *for all* $\ell \in \bar{S}$. *If conv*$(P(\bar{S},\emptyset))$ *and conv*(P) *are full-dimensional, and inequality* (4.6) *is facet-defining for conv*$(P(\bar{S},\emptyset))$*, then inequality* (4.7) *is facet-defining for conv*(P) *if* $\alpha_\ell = \zeta_\ell\binom{a_\ell}{B_\ell}$.

Proof. Although our model is slightly different than those used in the related literature (see Chapter 7 in [28], Section 2 in [45] and Chapter II.2 in [86]), the proof is essentially the same. ■

Note that the lifting coefficients are typically dependent on the lifting sequence in $\bar{S}$. If the lifting coefficients are unique regardless of the lifting order, i.e.,

$$\zeta\begin{pmatrix} z \\ \mathbf{u}\end{pmatrix} = \zeta_\ell\begin{pmatrix} z \\ \mathbf{u}\end{pmatrix} \quad \forall \ell \in \bar{S}, \tag{4.8}$$

then the lifting is said to be *sequence independent (SI)*.

Definition 4.1. *A function* $\phi : \mathbb{R}^n \to \mathbb{R}$ *is subadditive on* $Z \subseteq \mathbb{R}^n$ *if* $\phi(z_1)+\phi(z_2) \ge \phi(z_1+z_2)$ *whenever* $z_1, z_2 \in Z$ *and* $z_1 + z_2 \in Z$. *A function* ϕ *is called supperadditive on* Z *if* $-\phi$ *is subadditive on* Z.

Denote by set Z_j all possible values of $\binom{a_j}{B_j}$, $j \in \bar{S}$. To show that the lifting is SI, the general idea is that we first choose a convex set Z such that $Z_j \subseteq Z$ for all $j \in \bar{S}$, and then the following sufficient condition can be applied.

Theorem 4.1. *If* ζ *is subadditive on* Z*, then the lifting is SI.*

Proof. The proof is similar to Theorem 2 in [45]. ■

However, the subadditivity requirement may be too strong and difficult to verify in a high-dimensional space. In this chapter we also exploit another necessary and sufficient condition, which is similar in spirit to the one used in [114, 129].

Theorem 4.2. *The lifting is SI for all* $\binom{a_j}{B_j} \in Z_j, j \in \bar{S}$*, if and only if for any* $\Gamma \subseteq \bar{S}$ *function* ζ *satisfies:*

$$\sum_{j\in\Gamma} \zeta\begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix} \geq \zeta(\sum_{j\in\Gamma} \begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix}) \quad \forall \begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix} \in Z_j,\ j \in \Gamma. \tag{4.9}$$

Proof. Given a lifting order $1, 2, \ldots, \bar{s}$, we note that

$$\zeta_\ell \begin{pmatrix} z_\ell \\ \mathbf{u}_\ell \end{pmatrix} = \max_{\Gamma\subseteq[\ell-1]} \left\{ \zeta(\sum_{j\in\Gamma} \begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix} + \begin{pmatrix} z_\ell \\ \mathbf{u}_\ell \end{pmatrix}) - \sum_{j\in\Gamma} \zeta_j \begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix} \right\}. \tag{4.10}$$

We first show the "if" part. Since $\zeta_\ell\binom{z_\ell}{\mathbf{u}_\ell} \geq \zeta\binom{z_\ell}{\mathbf{u}_\ell}$, it suffices to prove that $\zeta_\ell\binom{z_\ell}{\mathbf{u}_\ell} \leq \zeta\binom{z_\ell}{\mathbf{u}_\ell}$. If the condition (4.9) holds, then

$$\sum_{j\in\Gamma} \zeta\begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix} + \zeta\begin{pmatrix} z_\ell \\ \mathbf{u}_\ell \end{pmatrix} \geq \zeta(\sum_{j\in\Gamma} \begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix} + \begin{pmatrix} z_\ell \\ \mathbf{u}_\ell \end{pmatrix}),$$

which yields the required result by applying the induction and equation (4.10).

To show the "only if" part, suppose there exists $\Gamma \subseteq \bar{S}$ such that $\sum_{j\in\Gamma} \zeta\binom{z_j}{\mathbf{u}_j} < \zeta(\sum_{j\in\Gamma}\binom{z_j}{\mathbf{u}_j})$ and $\sum_{j\in\Gamma'} \zeta\binom{z_j}{\mathbf{u}_j} \geq \zeta(\sum_{j\in\Gamma'}\binom{z_j}{\mathbf{u}_j})$ for all $\Gamma' \subsetneqq \Gamma$. Assume $\Gamma = \{1, \ldots, \ell\}$, and the lifting sequence is $1, \ldots, \ell$. Thus, based on the assumption and equation (4.10), we have $\zeta_j\binom{z_j}{\mathbf{u}_j} = \zeta\binom{z_j}{\mathbf{u}_j}$ for $j = 1, \ldots, \ell - 1$, and

$$\zeta_\ell \begin{pmatrix} z_\ell \\ \mathbf{u}_\ell \end{pmatrix} \geq \zeta(\sum_{j\in\Gamma} \begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix}) - \sum_{j=1}^{\ell-1} \zeta_j \begin{pmatrix} z_j \\ \mathbf{u}_j \end{pmatrix} > \zeta\begin{pmatrix} z_\ell \\ \mathbf{u}_\ell \end{pmatrix},$$

where the last inequality follows from $\sum_{j\in\Gamma} \zeta\binom{z_j}{\mathbf{u}_j} < \zeta(\sum_{j\in\Gamma}\binom{z_j}{\mathbf{u}_j})$. Therefore, we contradict with (4.8) and the result follows. ■

4.2.3 A Class of Subadditive Function

In this section, we introduce a new class of subadditive functions that contains as a special case the lifting function for the single-node flow set derived in [45].

Theorem 4.3. *Given a sequence of values $a_1, a_2, \ldots$ such that $a_k \geq a_{k+1} \geq 0$, let $A_0 = 0$ and $A_k = \sum_{j=1}^{k} a_j$ for $k = 1, 2, \ldots$. Define a piecewise concave function $\phi : \mathbb{R}_+ \to \mathbb{R}$ as follows:*

$$\phi(z) = \begin{cases} 0 & \text{if } z = 0, \\ g(z - A_k + v_k) + \phi(A_k) - g(v_k) & \text{if } A_k \leq z \leq A_{k+1}, k = 0, 1, \ldots, \end{cases} \tag{4.11}$$

where g is a concave function and $\{v_k\}_{k=0}^{\infty}$ is a sequence of values such that $v_{k-1} + a_k \leq v_k + a_{k+1}$. Then ϕ is a subadditive function on $\mathbb{R}_+$.

The proof the above result is detailed further in this section. Based on the definition of $\phi(z)$ in (4.11), we note that if $v_k = v_{k-1} + a_k$, for all $k = 1, 2, \ldots$, then $\phi(z) = g(z + v_0) - g(v_0)$ for $z \in \mathbb{R}_+$ is simply a concave function. Following this observation, we derive another variant of (4.11).

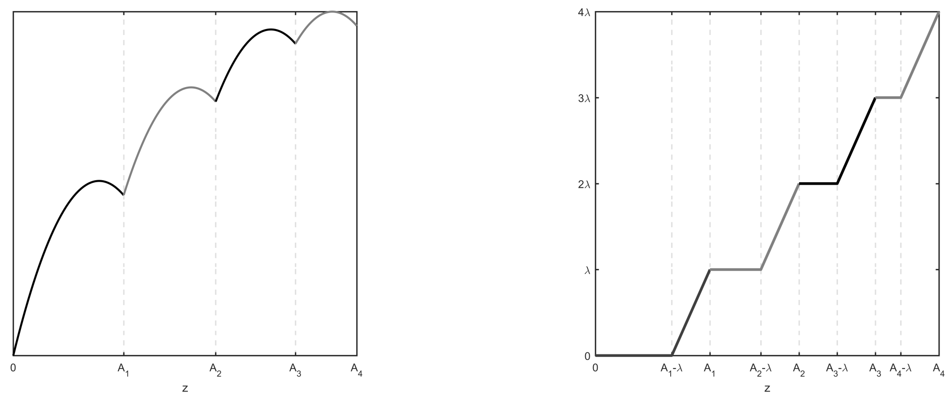
Proposition 4.4. *Given a sequence of values $a_1, a_2, \ldots, a_n$ such that $a_k \geq a_{k+1} \geq 0$, let $A_0 = 0$ and $A_k = \sum_{j=1}^{k} a_j$ for $k \in [n]$. A piecewise concave function $\widehat{\phi} : \mathbb{R}_+ \to \mathbb{R}$ is defined as follows: $\widehat{\phi}(0) = 0$ and*

$$\widehat{\phi}(z) = \begin{cases} g(z - A_k + v_k) + \widehat{\phi}(A_k) - g(v_k) & \text{if } A_k \leq z \leq A_{k+1}, \\ & \quad k = 0, 1, \ldots, n-2, \\ g(z - A_{n-1} + v_{n-1}) + \widehat{\phi}(A_{n-1}) - g(v_{n-1}) & \text{if } z \geq A_{n-1}, \end{cases} \tag{4.12}$$

where g is a concave function and $\{v_k\}_{k=0}^{n-1}$ is a sequence of values such that $v_{k-1} + a_k \leq v_k + a_{k+1}$. Then $\widehat{\phi}$ is a subadditive function on $\mathbb{R}_+$.

Proof. Define $a_k = a_n$ for $k = n+1, n+2, \ldots$, and $v_k = v_{k-1} + a_k$ for $k = n, n+1, \ldots$. Then function $\widehat{\phi}(z)$ can be written in the form of (4.11), and the results follows. ■

Furthermore, the superadditivity of ϕ can be verified when g is a convex function.



(a) $g(z) = -(z-b)^2$.

(b) $g(z) = \max\{0, z - (a_1 - \lambda)\}$.

Figure 6: The subadditive and superadditive examples that illustrate Theorem 4.3 and Corollary 4.2, respectively.

Corollary 4.2. *If function g is convex, then $\phi(z)$ in the form of either (4.11) or (4.12) is a superadditive function on $\mathbb{R}_+$.*

Proof. Let $\tilde{g} = -g$ and denote $\tilde{\phi}(z)$ as the function ϕ that replaces g by $\tilde{g}$. It follows that $\tilde{g}$ is a concave function, and $\tilde{\phi}(z) = -\phi(z)$ for all $z \in \mathbb{R}_+$. Therefore, based on Theorem 4.3 and Proposition 4.4, we conclude that $\tilde{\phi}$ is subadditive and the result follows. ■

Note that in the above results we do not specify any particular functional form of g and require only its concavity. Thus, subadditive function ϕ can be constructed no matter whether g is linear, monotonic or differentiable. Figure 6 illustrates two examples when function g is quadratic and piecewise linear.

We point out that Theorem 4.3 provides a framework to either verify the function's subadditivity or to construct its subadditive approximation for functions, ϕ, with particular structures. As long as we recognize that the function is piecewise concave, then our results might be applicable. We next present one immediate application to the single-node flow set F, which matches the corresponding result in [45].

Example 4.1 (Adapted from Section 5 in [45])**.** *Consider the single-node flow set F, with $N^- = \emptyset$ and $\mathcal{X} = \{0,1\}^n$. Given a flow cover S such that $\lambda = \sum_{j\in S} a_j - b > 0$, the flow cover inequality*

$$\sum_{j\in S} x_j + \sum_{j\in S} \max\{0, a_j - \lambda\}(1 - y_j) \le b$$

defines a facet of conv$(F(\bar{S}, \emptyset))$. Suppose $S = \{1, \ldots, s\}$ and $a_1 \ge \cdots \ge a_\ell > \lambda \ge a_{\ell+1} \ge \cdots \ge a_s$, let $A_0 = 0$ and $A_k = \sum_{j=1}^{k} a_k$ for $k \in [\ell]$. Then the lifting function of the flow cover inequality (depicted in Figure 6(b)) is $\phi(0) = 0$ and

$$\phi(z) = \begin{cases} g(z - A_k + a_1 - a_{k+1}) + \phi(A_k) & \text{if } A_k \le z \le A_{k+1}, k = 0, 1, \ldots, \ell - 2, \\ g(z - A_\ell + a_1) + \phi(A_{\ell-1}) & \text{if } z \ge A_{\ell-1}, \end{cases}$$

where $\phi(A_k) = k\lambda$ for $k = 0, 1, \ldots, \ell - 1$, and $g(z) = \max\{0, z - (a_1 - \lambda)\}$ is a convex function. Let $v_k = a_1 - a_{k+1}$ for $k = 0, 1 \ldots, \ell - 1$, then $g(v_k) = 0$ and the above lifting function can be written in the form of (4.12). Also, observe that $v_{k-1} + a_k = a_1$ for all $k \in [\ell]$. Therefore, based on Corollary 4.2, the lifting function of the flow cover inequality is superadditive on $\mathbb{R}_+$.

We next provide the detailed proof of Theorem 4.3, with some preliminary results presented first in the following two lemmata.

Lemma 4.1. *Let $z \in [A_k, A_{k+1}]$ for some integer $k \ge 1$. Then for any $\Delta \ge 0$ and $z + \Delta \le A_{k+1}$, we have*

$$\phi(z + \Delta) - \phi(z) \le \phi(z - \sum_{j=\ell}^{k+1} a_j + \Delta) - \phi(z - \sum_{j=\ell}^{k+1} a_j) \quad \forall \ell = 2, \ldots, k + 1.$$

Proof. Note that for any $\ell \ge 2$, $z - \sum_{j=\ell}^{k+1} a_j \ge A_k - \sum_{j=\ell}^{k+1} a_j \ge a_1 - a_{k+1} \ge 0$ due to $z \in [A_k, A_{k+1}]$ and $a_1 \ge a_k \ge a_{k+1}$. We next show the statement for each $\ell = 2, \ldots, k+1$ by induction.

a) If $\ell = k+1$, then $z - a_{k+1} \in [A_{k-1}, A_k]$ and $z - a_{k+1} + \Delta \leq A_k$. Let $\Omega = z - A_{k-1}$, we have that

$$\begin{aligned}
\phi(z+\Delta) - \phi(z) &= g(z - A_k + v_k + \Delta) - g(z - A_k + v_k) \\
&= g(\Omega + v_k - a_k + \Delta) - g(\Omega + v_k - a_k) \\
&\leq g(\Omega + v_{k-1} - a_{k+1} + \Delta) - g(\Omega + v_{k-1} - a_{k+1}) \\
&= \phi(z - a_{k+1} + \Delta) - \phi(z - a_{k+1}),
\end{aligned}$$

where the inequality follows from $v_{k-1} + a_k \leq v_k + a_{k+1}$ (recall the assumptions in Theorem 4.3) and the concavity of g; that is, $g(z_0 + \Delta) - g(z_0) \geq g(z_1 + \Delta) - g(z_1)$ if $z_0 \leq z_1$ and $\Delta \geq 0$.

b) If the statement holds for $\ell = \ell_0 + 1$, $2 \leq \ell_0 \leq k$, then we show that the statement also holds for $\ell = \ell_0$. Based on the induction hypothesis, we have

$$\phi(z+\Delta) - \phi(z) \leq \phi(z - \sum_{j=\ell_0+1}^{k+1} a_j + \Delta) - \phi(z - \sum_{j=\ell_0+1}^{k+1} a_j) = \phi(z' + \Delta) - \phi(z'),$$

where we let $z' = z - \sum_{j=\ell_0+1}^{k+1} a_j$. Also, observe that $z' \in [A_{\ell_0 - 1}, A_{\ell_0}]$ and $z' + \Delta = z + \Delta - \sum_{j=\ell_0+1}^{k+1} a_j \leq A_{k+1} - \sum_{j=\ell_0+1}^{k+1} a_j = A_{\ell_0}$. Thus, based on a), we have

$$\phi(z'+\Delta) - \phi(z') \leq \phi(z' - a_{\ell_0} + \Delta) - \phi(z' - a_{\ell_0}) = \phi(z - \sum_{j=\ell_0}^{k+1} a_j + \Delta) - \phi(z - \sum_{j=\ell_0}^{k+1} a_j),$$

which yields the result.

■

Lemma 4.2. *Let $\Delta \in [0, a_{k+1}]$ for some integer $k \geq 0$. Then for any $z \geq A_k$, we have*

$$\phi(A_k + \Delta) - \phi(A_k) \geq \phi(z + \Delta) - \phi(z).$$

Proof. Suppose $z \in [A_{k_1}, A_{k_1+1}]$ and $z + \Delta \in [A_{k_2}, A_{k_2+1}]$, where $k \leq k_1 \leq k_2$. We establish the result for each $k_2 - k_1$ by induction.

a) If $k_2 = k_1$, then based on Lemma 4.1, we have

$$\begin{aligned}\phi(z+\Delta)-\phi(z) &\le \phi(z-\sum_{j=k+2}^{k_1+1} a_j+\Delta)-\phi(z-\sum_{j=k+2}^{k_1+1} a_j)\\ &= g(\Omega+v_k+\Delta)-g(\Omega+v_k)\\ &\le g(v_k+\Delta)-g(v_k)=\phi(A_k+\Delta)-\phi(A_k),\end{aligned}$$

where $\Omega = z - A_k - \sum_{j=k+2}^{k_1+1} a_j$, and the second inequality follows from the fact that $a_{k+1} \ge a_{k_1+1}$ and $\Omega \ge z - A_k - \sum_{j=k+1}^{k_1} a_j = z - A_{k_1} \ge 0$.

b) If the statement holds for $k_2 - k_1 = m \ge 0$, then we show that the statement also holds for $k_2 - k_1 = m + 1 \ge 1$. Let $z' = A_{k_1+1}$, and $\Delta' = A_{k_1+1} - z$, then $\Delta \ge \Delta'$ due to $z + \Delta \ge A_{k_2} \ge A_{k_1+1}$. We have that

$$\begin{aligned}\phi(A_{k_1+1})-\phi(z) &= \phi(z')-\phi(z'-\Delta')\\ &\le \phi(z'-\sum_{j=k+2}^{k_1+1} a_j)-\phi(z'-\Delta'-\sum_{j=k+2}^{k_1+1} a_j)\\ &= \phi(A_{k+1})-\phi(A_{k+1}-\Delta')\\ &\le \phi(A_k+\Delta)-\phi(A_k+\Delta-\Delta'),\end{aligned}$$

where the first inequality follows from Lemma 4.1 with the conditions that $\Delta' \le a_{k_1+1}$ and $z' - \Delta' \in [A_{k_1}, A_{k_1+1}]$; and the second inequality follows from the fact that ϕ is concave on $[A_k, A_{k+1}]$ and $A_{k+1} - \Delta' \in [A_k, A_{k+1}], A_k + \Delta \in [A_k, A_{k+1}]$.

Also, note that $z' \in [A_{k_1+1}, A_{k_1+2}]$ and $z + \Delta = z' + \Delta - \Delta' \in [A_{k_2}, A_{k_2+1}]$. Since $k_2 - (k_1 + 1) = m$, by the induction hypothesis, we have

$$\phi(z+\Delta)-\phi(A_{k_1+1})=\phi(z'+\Delta-\Delta')-\phi(z')\le\phi(A_k+\Delta-\Delta')-\phi(A_k).$$

Summing the above two inequalities, we obtain the desired inequality.

■

Theorem 4.3. To show that ϕ is subadditive on $[0, +\infty)$, it is sufficient to prove that the following inequality holds:

$$\phi(z) - \phi(0) \geq \phi(z + \Delta) - \phi(\Delta) \quad \forall z \geq 0, \Delta \geq 0.$$

First, let $\Delta' = a_{j+1}$ for some $j \geq 0$ and $z' = A_j + \Delta$ for any $\Delta \geq 0$. Observe that $z' \geq A_j$ and $\Delta' \in [0, a_{j+1}]$. Therefore, by Lemma 2, we have

$$\phi(A_j + \Delta') - \phi(A_j) \geq \phi(z' + \Delta') - \phi(z') \quad \forall j = 0, 1, \ldots,$$

which can be rewritten as

$$\phi(A_{j+1}) - \phi(A_j) \geq \phi(A_{j+1} + \Delta) - \phi(A_j + \Delta) \quad \forall j = 0, 1, \ldots. \tag{4.13}$$

Suppose $z \in [A_k, A_{k+1}]$ for some $k \geq 0$. Let $z'' = A_k + \Delta$ and $\Delta'' = z - A_k$, then observe that $z'' \geq A_k$ and $\Delta'' \in [0, a_{k+1}]$. By Lemma 4.2 we have

$$\phi(A_k + \Delta'') - \phi(A_k) \geq \phi(z'' + \Delta'') - \phi(z''). \tag{4.14}$$

Note that $A_k + \Delta'' = z$ and $z'' + \Delta'' = z + \Delta$. Finally, summing inequalities (4.13) over $j = 0, 1, \ldots, k-1$ and (4.14), yields the desired result.

■

4.3 Lifting for conv(P_0)

In this section, we study the lifting procedure for the convex hull of P when $\mathcal{X} = \{0, 1\}^n$. Recall that

$$P_0 = \{(w, x) \in \mathbb{R} \times \{0, 1\}^n : \ w \leq g(a^T x)\}.$$

We first derive a set of valid inequalities by performing exact lifting on variables fixed at zeros in Section 4.3.1. As it turns out, the lifting is SI and the resulting inequalities are facet-defining for conv(P_0). Similarly, in Section 4.3.2, another family of facet-defining inequalities are derived by lifting variables fixed at ones.

4.3.1 Lifted Inequalities From $P_0(\bar{S}, \emptyset)$

Given a set $S \subseteq N$, consider set $P_0(\bar{S}, \emptyset)$ by fixing $x_j = 0$ for $j \in \bar{S}$ in P_0, i.e.,

$$P_0(\bar{S}, \emptyset) = \{(w, x) \in \mathbb{R} \times \{0,1\}^n : \ w \leq g(a^T x), \ x_j = 0 \ \forall j \in \bar{S}\}.$$

Following Proposition 4.1, the inequality

$$w \leq f(S) - \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j) \tag{4.15}$$

is facet-defining for the convex hull of $P_0(\bar{S}, \emptyset)$. To lift seed inequality (4.15) for $\text{conv}(P_0)$, the corresponding lifting function is then given by:

$$\begin{aligned}
\gamma_0(z) = \max_{w,x} \ & w + \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j) - f(S) \\
\text{s.t. } & w \leq g(\sum_{j \in S} a_j x_j + z), \\
& x_j \in \{0,1\} \qquad \forall j \in S,
\end{aligned}$$

where $z \in \mathbb{R}_+$. For some $z \in \mathbb{R}_+$ and any subset $\Lambda \subseteq S$, denote the objective function value for x_Λ, and $w = g(a^T x_\Lambda + z)$ of the above lifting problem $\gamma_0(z)$ as: $h(z, \Lambda) = g(a(\Lambda) + z) + \sum_{j \in S \setminus \Lambda} \rho_j(S \setminus j) - f(S)$.

Two basic properties of h are discussed next. We note that h is not directly defined in [2, 122], but its properties are used implicitly there. In particular, Lemma 4.4 below is also provided in [122] (see Lemma 1), admittedly in a somewhat different form. Nevertheless, for the sake of completeness, we provide self-contained proofs in our discussion below.

Lemma 4.3. *Let $\Lambda \subseteq S$ and $j \in S \setminus \Lambda$, then*

(i) *if $a(\Lambda) + a_j + z \leq a(S)$, then $h(z, \Lambda \cup j) \geq h(z, \Lambda)$;*

(ii) *if $a(\Lambda) + a_j + z \geq a(S)$, then $h(z, \Lambda \cup j) \leq h(z, \Lambda)$.*

Proof. We prove only statement (i) below, and (ii) can be proved in a similar manner. Let $\Omega = a(\Lambda) + z$, then $\Omega + a_j \leq a(S)$, and

$$\begin{aligned} h(z, \Lambda \cup j) - h(z, \Lambda) &= g(a(\Lambda) + a_j + z) - g(a(\Lambda) + z) - \rho_j(S \setminus j) \\ &= [g(\Omega + a_j) - g(\Omega)] - [g(a(S)) - g(a(S) - a_j)] \geq 0, \end{aligned}$$

where the inequality follows from the concavity of g. ■

Lemma 4.4. *Let $\Lambda \subseteq S$ and $j \in S \setminus \Lambda$. If there exists $i \in \Lambda$ such that $a_i \leq a_j$, then*

(i) *if $a(\Lambda) + a_j + z \leq a(S)$, then $h(z, \Lambda \cup j \setminus i) \geq h(z, \Lambda)$;*

(ii) *if $a(\Lambda) + a_j + z \geq a(S)$, then $h(z, \Lambda \cup j \setminus i) \leq h(z, \Lambda)$.*

Proof. We prove only statement (i) below, and (ii) can be proved in a similar manner. Let $\Delta = a_j - a_i$ and $\Omega = a(\Lambda) + z$, then $\Delta \geq 0$ and $\Omega \leq a(s) - a_j$, and

$$\begin{aligned} h(z, \Lambda \cup j \setminus i) - h(z, \Lambda) &= g(a(\Lambda) + a_j - a_i + z) + \rho_i(S \setminus i) - g(a(\Lambda) + z) - \rho_j(S \setminus j) \\ &= [g(\Omega + \Delta) - g(\Omega)] - [g(a(S) - a_i) - g(a(S) - a_j)] \\ &= [g(\Omega + \Delta) - g(\Omega)] - [g(a(S) - a_j + \Delta) - g(a(S) - a_j)] \\ &\geq 0, \end{aligned}$$

where the inequality follows from the concavity of g. ■

With a similar strategy as in Proposition 5 of [2], we can derive exactly the same formula for $\gamma_0(z)$ and for any general concave function g.

Proposition 4.5. *Suppose $S = \{1, \ldots, s\}$ and $a_1 \geq \cdots \geq a_s$, let $A_0 = 0$ and $A_k = \sum_{j=1}^{k} a_j$ for $k \in S$. The lifting function $\gamma_0(z)$ is computed as follows:*

$$\gamma_0(z) = \begin{cases} g(z - A_{k+1} + a(S)) + \gamma_0(A_k) - g(a(S) - a_{k+1}) & \text{if } A_k \leq z \leq A_{k+1}, \\ & \quad k = 0, \ldots, s-2, \\ g(z) + \gamma_0(A_{s-1}) - g(a(S) - a_s) & \text{if } z \geq A_{s-1}, \end{cases}$$

where $\gamma_0(A_k) = \sum_{j=1}^{k} \rho_j(S \setminus j)$ for any $k \in S$.

Proof. If $A_k \le z \le A_{k+1}$ for some $k = 0, \ldots, s-2$, then it suffices to show that $\{k+2, \ldots, s\}$ is an optimal solution set to compute $\gamma_0(z)$ in the lifting problem. Suppose there exists a different optimal solution set $\Lambda^* \subseteq S$. Based on Lemma 4.3, we can find a Λ^* such that $a(\Lambda^*) + z \le a(S)$ and $a(\Lambda^* \cup j) + z > a(S)$ for all $j \in S \setminus \Lambda^*$. Therefore, if $\{k+2, \ldots, s\} \subseteq \Lambda^*$, then $\Lambda^* = \{k+2, \ldots, s\}$.

Suppose $\{k+2, \ldots, s\} \nsubseteq \Lambda^*$. Let k_0 be the largest index in S such that $k_0 \notin \Lambda^*$, then $k_0 \ge k+2$. Next, we construct a new optimal solution set $\tilde{\Lambda}$ such that $\{k_0, \ldots, s\} \subseteq \tilde{\Lambda}$. Since $a(\Lambda^*) + a_{k_0} + z > a(S)$, there must exist some $j_0 \in \Lambda^*$ such that $j_0 < k_0$, otherwise $a(\Lambda^*) + a_{k_0} + z = \sum_{j=k_0}^{s} a_j + z \le \sum_{j=k+2}^{s} a_j + z \le a(S)$. Since $a_{j_0} \ge a_{k_0}$, then based on Lemma 4.4, we have $h(z, \Lambda^* \cup k_0 \setminus j_0) \ge h(z, \Lambda^*)$. Thus, $\tilde{\Lambda} = \Lambda^* \cup k_0 \setminus j_0$ is also an optimal solution. Proceeding in an iterative manner, we have that $\{k+2, \ldots, s\}$ is an optimal solution.

If $z \ge A_{s-1}$, then for any nonempty set $\Lambda \subseteq S$, we have $a(\Lambda) + z \ge a(S)$. Based on Lemma 4.3, it immediately yields that $h(z, \Lambda) \le h(z, \emptyset)$. Therefore, $\emptyset$ is an optimal solution set to compute $\gamma_0(z)$ in the lifting problem and the result follows. ■

Let $v_k = a(S) - a_{k+1}$, then $v_k + a_{k+1} = a(S)$ for $k = 0, 1, \ldots, s-1$. Given that $\gamma_0(A_k) = \sum_{j=1}^{k} \rho_j(S \setminus j)$ and $g(v_k) = g(a(S) - a_{k+1})$, it follows from Proposition 4.4 that the lifting function $\gamma_0(z)$ is subadditive on $\mathbb{R}_+$. Thus, based on Proposition 4.3 and Theorem 4.1, the exact lifting is SI, and the resulting inequality from lifting is facet-defining for $\text{conv}(P_0)$. This result is formally stated as follows.

Theorem 4.4. *For any $S \subseteq N$, the inequality*

$$w \le f(S) - \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j) + \sum_{j \in \bar{S}} \gamma_0(a_j) x_j \tag{4.16}$$

is facet-defining for conv(P_0).

Proof. It directly follows from the above discussion. ■

Note that Ahmed and Atamtürk [2] derive a subadditive approximation to the lifting function $\gamma_0(z)$ by applying the continuous relaxation and then using the KKT conditions to solve the convex program. Nevertheless, as pointed out in Proposition 4.4 and Theorem 4.4 the exact lifting function is naturally subadditive and we can directly obtain facet-defining inequalities without any approximations. Furthermore, similar to the results in [2], we can show that any fractional extreme points of relax(P_0) can be cut off by the lifted facet-defining inequalities (4.16); the proofs in this chapter are omitted for brevity.

Next, we provide an example to illustrate the difference between inequalities obtained by approximate lifting [2] and our exact lifting.

Example 4.2. *Consider* $P_0 = \{(w, x) \in \mathbb{R} \times \{0,1\}^n : \; w \leq -\exp(-a^T x)\}$*, where* $n = 7$*, and* $a = (0.8, 0.7, 0.7, 0.6, \; 0.5, 0.3, 0.2)^T$*. Let* $S = \{3, 4, 5, 6\}$*. The seed inequality* (4.15) *is*

$$w \leq -0.4695 + 0.1241x_3 + 0.1007x_4 + 0.0794x_5 + 0.0428x_6.$$

The approximate lifted inequality in [2] is computed as:

$$\begin{aligned} w \leq & -0.4695 + 0.1484x_1 + 0.1317x_2 + 0.1241x_3 + 0.1007x_4 \\ & + 0.0794x_5 + 0.0428x_6 + 0.0447x_7. \end{aligned}$$

The lifted inequality (4.16)*, as in the following, dominates the above approximation by having smaller coefficients of* x_1 *and* x_2*:*

$$\begin{aligned} w \leq & -0.4695 + \mathbf{0.1454}x_1 + \mathbf{0.1241}x_2 + 0.1241x_3 + 0.1007x_4 \\ & + 0.0794x_5 + 0.0428x_6 + 0.0447x_7. \end{aligned}$$

Figure 7 shows the lifting function $\gamma_0(z)$ *and its approximation* $\hat{\gamma}_0(z)$ *proposed in [2]. In around 80% intervals, the approximations* $\hat{\gamma}_0$ *are strictly larger than the exact values* γ_0*. In particular, the largest difference ratio in this example is around 20%.*

Denote by K_0 and F_0 the mixed 0-1 knapsack set and the single-node flow set when $\mathcal{X} = \{0,1\}^n$, respectively. As mentioned in Section 4.1, the well-known mixed 0-1 knapsack set K_0, is a special case of P_0. Next, we show that the lifted facet-defining inequalities for conv(K_0) can be obtained directly via (4.16). A class of valid inequalities for conv(F_0) can also be easily derived through the reduction to K_0, which is well studied in [83]. Hence, our study on this submodular set unifies those classical results.

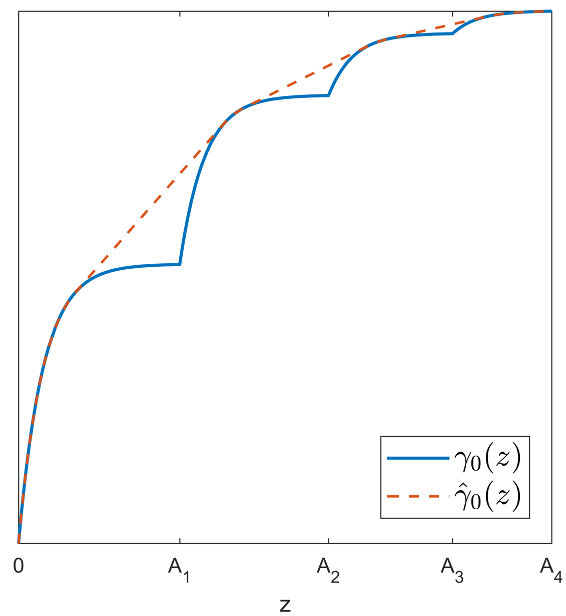


Figure 7: Lifting function $\gamma_0(z)$ and its approximation $\hat{\gamma}_0(z)$ proposed in [2].

Corollary 4.3 ([83]). *Consider any $S \subseteq N$ such that $\lambda = a(S) - b > 0$. Suppose $S = \{1, \ldots, s\}$ is such that $a_1 \geq \cdots \geq a_\ell > \lambda \geq \cdots \geq a_s$. Then the inequality*

$$-\pi \leq -\lambda + \sum_{j=1}^{\ell} \lambda(1 - x_j) + \sum_{j=\ell+1}^{s} a_j(1 - x_j) + \sum_{j \in \bar{S}} \gamma_0(a_j)x_j$$

is facet-defining for $conv(K_0)$, where $\gamma_0(z)$ is computed as

$$\gamma_0(z) = \begin{cases} -k\lambda & \textit{if } A_k \leq z \leq A_{k+1} - \lambda, k = 0, \ldots, \ell - 1, \\ -(k+1)\lambda + A_{k+1} - z & \textit{if } A_{k+1} - \lambda \leq z \leq A_{k+1}, k = 0, \ldots, \ell - 1, \\ -\ell\lambda + A_\ell - z & \textit{if } z \geq A_\ell. \end{cases}$$

Proof. Recall from the discussion of Section 4.1 that if we define $w = -\pi$ and $g(z) = \min\{0, b - z\}$, then K_0 is equivalent to the form of P_0. Therefore, for any $S \subseteq N$ such that $\lambda = a(S) - b > 0$, we have $f(S) = g(a(S)) = -\lambda$ and

$$\rho_j(S \setminus j) = -\lambda - \min\{0, a_j - \lambda\} = \begin{cases} -\lambda & \text{if } j = 1, \ldots, \ell, \\ -a_j & \text{if } j = \ell + 1, \ldots, s. \end{cases}$$

If $A_k \le z \le A_{k+1}$ for some $k = 0, \dots, \ell - 1$, then based on Proposition 4.5, we have $\sum_{j=1}^{k} \rho_j(S \setminus j) = -k\lambda$, $g(a(S) - a_{k+1}) = 0$ and

$$\gamma_0(z) = g(z - A_{k+1} + a(S)) - k\lambda = \min\{0, A_{k+1} - z - \lambda\} - k\lambda.$$

If $A_k \le z \le A_{k+1}$ for some $k = \ell, \dots, s-2$, then $\sum_{j=1}^{k} \rho_j(S \setminus j) = -\ell\lambda - \sum_{j=\ell+1}^{k} a_j$ and $g(a(S) - a_{k+1}) = -\lambda + a_{k+1}$. Based on Proposition 4.5, we have

$$\begin{aligned}\gamma_0(z) &= \min\{0, A_{k+1} - z - \lambda\} - (\ell - 1)\lambda - \sum_{j=\ell+1}^{k+1} a_j \\ &= A_{k+1} - z - \lambda - (\ell - 1)\lambda - \sum_{j=\ell+1}^{k+1} a_j = A_\ell - z - \ell\lambda,\end{aligned}$$

where the second equality follows from $A_{k+1} - z - \lambda \le a_{k+1} - \lambda \le 0$.

If $z \ge A_{s-1}$, then the derivation is omitted for brevity as it is similar to the case where $A_k \le z \le A_{k+1}$ for some $k = \ell, \dots, s-2$.

■

4.3.2 Lifted Inequalities From $P_0(\emptyset, S)$

Given a set $S \subseteq N$, consider set $P_0(\emptyset, S)$ by fixing $x_j = 1$ for $j \in S$ in P_0, i.e.,

$$P_0(\emptyset, S) = \{(w, x) \in \mathbb{R} \times \{0,1\}^n : \ w \le g(a^T x), x_j = 1 \ \forall j \in S\}.$$

It follows from Proposition 4.1 that the inequality

$$w \le f(S) + \sum_{j \in \bar{S}} \rho_j(S) x_j \tag{4.17}$$

defines a facet of the convex hull of $P_0(\emptyset, S)$. To lift seed inequality (4.17) for $\text{conv}(P_0)$, the corresponding lifting function is given as:

$$\begin{aligned}\eta_0(z) = \max_{w,x} \ & w - \sum_{j \in \bar{S}} \rho_j(S) x_j - f(S) \\ \text{s.t. } & w \le g\Big(\sum_{j \in \bar{S}} a_j x_j + a(S) - z\Big), \\ & x_j \in \{0,1\} \qquad \forall j \in \bar{S},\end{aligned} \tag{4.18}$$

where $z \in \mathbb{R}_+$. For some $z \in \mathbb{R}_+$ and any subset $\Lambda \subseteq \bar{S}$, denote the objective function value for x_Λ and $w = g(a^T x_\Lambda + a(S) - z)$ of the above lifting problem as $\chi(z, \Lambda) = g(a(\Lambda) + a(S) - z) - \sum_{j \in \Lambda} \rho_j(S) - f(S)$.

Next, we discuss two basic properties of χ. Similar to the derivations in Section 4.3.1, we note that χ is not directly defined in [2, 122], but its properties are used implicitly there. As in the above, we provide self-contained proofs for the sake of completeness in our discussion below.

Lemma 4.5. *Let $\Lambda \subseteq \bar{S}$ and $j \in \bar{S} \setminus \Lambda$, then*

(i) if $a(\Lambda) - z \leq 0$, then $\chi(z, \Lambda \cup j) \geq \chi(z, \Lambda)$;

(ii) if $a(\Lambda) - z \geq 0$, then $\chi(z, \Lambda \cup j) \leq \chi(z, \Lambda)$.

Proof. We establish only statement (i) below, and (ii) can be proved in a similar manner. Let $\Omega = a(\Lambda) + a(S) - z$, then $\Omega \leq a(S)$ and

$$\begin{aligned} \chi(z, \Lambda \cup j) - \chi(z, \Lambda) &= g(\Omega + a_j) - \rho_j(S) - g(\Omega) \\ &= [g(\Omega + a_j) - g(\Omega)] - [g(a(S) + a_j) - g(a(S))] \geq 0, \end{aligned}$$

where the inequality follows from the fact that g is concave. ■

Lemma 4.6. *Let $\Lambda \subseteq \bar{S}$ and $j \in \bar{S} \setminus \Lambda$. If there exists $i \in \Lambda$ such that $a_i \leq a_j$, then*

(i) if $a(\Lambda) - a_i - z \leq 0$, then $\chi(z, \Lambda \cup j \setminus i) \geq \chi(z, \Lambda)$;

(ii) if $a(\Lambda) - a_i - z \geq 0$, then $\chi(z, \Lambda \cup j \setminus i) \leq \chi(z, \Lambda)$.

Proof. We prove only statement (i) below, and (ii) can be proved in a similar manner. Let $\Delta = a_j - a_i$ and $\Omega = a(\Lambda) + a(S) - z$, then $\Delta \geq 0$, $\Omega \leq a(S) + a_i$ and

$$\begin{aligned} \chi(z, \Lambda \cup j \setminus i) - \chi(z, \Lambda) &= g(\Omega + a_j - a_i) - \rho_j(S) - g(\Omega) + \rho_i(S) \\ &= [g(\Omega + \Delta) - g(\Omega)] - [g(a(S) + a_j) - g(a(S) + a_i)] \\ &= [g(\Omega + \Delta) - g(\Omega)] - [g(a(S) + a_i + \Delta) - g(a(S) + a_i)] \\ &\geq 0, \end{aligned}$$

where the inequality follows from the concavity of g . ■

We can compute $\eta_0(z)$ using the same formula as in [2] (see Proposition 10 there), as given below. Furthermore, we strengthen the result in [2] by showing that the lifting function is, in fact, subadditive.

Proposition 4.6. *Suppose $\bar{S} = \{1, 2, \ldots, \bar{s}\}$ such that $a_1 \geq \cdots \geq a_{\bar{s}}$, let $A_0 = 0$ and $A_k = \sum_{j=1}^{k} a_j$ for $k \in \bar{S}$. The lifting function $\eta_0(z)$ is computed as follows:*

$$\eta_0(z) = \begin{cases} g(A_{k+1} + a(S) - z) + \eta_0(A_k) - g(a(S) + a_{k+1}) & \text{if } A_k \leq z \leq A_{k+1}, \\ & \quad k = 0, \ldots, \bar{s} - 2, \\ g(a(N) - z) + \eta_0(A_{\bar{s}-1}) - g(a(S) + a_{\bar{s}}) & \text{if } z \geq A_{\bar{s}-1}, \end{cases}$$

where $\eta_0(A_k) = -\sum_{j=1}^{k} \rho_j(S)$. Furthermore, function $\eta_0(z)$ is subadditive on $\mathbb{R}_+$.

Proof. If $A_k < z \leq A_{k+1}$ for some $k = 0, 1, \ldots, \bar{s}-1$, then it suffices to show that $\{1, \ldots, k+1\}$ is an optimal solution set to compute $\eta_0(z)$ in the lifting problem. Suppose there exists a different optimal solution set $\Lambda^* \subseteq \bar{S}$. Based on Lemma 4.5, there exists a Λ^* such that $a(\Lambda^*) - z \geq 0$ and $a(\Lambda^* \setminus j) - z \leq 0$ for any $j \in \Lambda^*$. Therefore, if $\{1, \ldots, k+1\} \subseteq \Lambda^*$, then $\Lambda^* = \{1, \ldots, k+1\}$.

Suppose $\{1, \ldots, k+1\} \nsubseteq \Lambda^*$. Let k_0 be the smallest index in $\bar{S}$ such that $k_0 \notin \Lambda^*$, then $k_0 \leq k+1$. We next construct a new optimal solution set $\tilde{\Lambda}$ such that $\{1, \ldots, k_0\} \subseteq \tilde{\Lambda}$. First, there must exist some $j_0 \in \Lambda^*$ such that $j_0 > k+1$, otherwise $a(\Lambda^*) \leq A_{k+1} - a_{k_0} < z$. Let $\tilde{\Lambda} = \Lambda \cup k_0 \setminus j_0$, then based on Lemma 4.6, we have $\chi(z, \tilde{\Lambda}) \geq \chi(z, \Lambda^*)$ as $a_{k_0} \geq a_{j_0}$ and $a(\Lambda^* \setminus j_0) - z \leq 0$. It implies that $\tilde{\Lambda}$ is also an optimal solution. Proceeding in an iterative manner, we can conclude that $\{1, \ldots, k+1\}$ is an optimal solution.

If $z \geq A_{\bar{s}-1}$, then we have $a(\Lambda) - z \leq 0$ for any set $\Lambda \subsetneqq \bar{S}$, which yields that $\chi(z, \Lambda) \leq \chi(z, \bar{S})$ based on Lemma 4.5. Therefore, $\bar{S}$ is an optimal solution set to compute $\eta_0(z)$ in the lifting problem and the result follows.

To show the subadditivity of $\eta_0(z)$, let $\tilde{g}(z) = g(-z)$. Then $\tilde{g}$ is also a concave function. Thus, $\eta_0(z)$ can be rewritten as

$$\eta_0(z) = \begin{cases} \tilde{g}(z - A_{k+1} - a(S)) + \eta_0(A_k) - \tilde{g}(-a(S) - a_{k+1}) & \text{if } A_k \leq z \leq A_{k+1}, \\ & \quad k = 0, \ldots, \bar{s} - 2, \\ \tilde{g}(z - a(N)) + \eta_0(A_{\bar{s}-1}) - \tilde{g}(-a(S) - a_{\bar{s}}) & \text{if } z \geq A_{\bar{s}-1}. \end{cases}$$

Let $v_k = -a(S) - a_{k+1}$, then $v_k + a_{k+1} = -a(S)$ for all $k = 0, 1, \ldots, \bar{s} - 1$. It immediately follows from Proposition 4.4 that η_0 is subadditive on $\mathbb{R}_+$. ■

Consequently, based on Proposition 4.3 and Theorem 4.1, the lifting is SI, and the lifted inequality is facet-defining for conv(P_0). Again, it unifies some classical results for the mixed 0-1 knapsack set K_0 and the single-node flow set F_0.

Theorem 4.5. *For any $S \subseteq N$, the inequality*

$$w \leq f(S) + \sum_{j \in \bar{S}} \rho_j(S) x_j + \sum_{j \in S} \eta_0(a_j)(1 - x_j) \tag{4.19}$$

is facet-defining for conv(P_0).

Proof. It directly follows from Proposition 4.6. ■

Corollary 4.4 ([83])**.** *Consider any $S \subseteq N$ such that $\mu = b - a(S) > 0$. Suppose $\bar{S} = \{1, \ldots, \bar{s}\}$ is such that $a_1 \geq \cdots \geq a_{\bar{\ell}} > \mu \geq \cdots \geq a_{\bar{s}}$. Then the inequality*

$$-\pi \leq \sum_{j=1}^{\bar{\ell}} (\mu - a_j) x_j + \sum_{j \in S} \eta_0(a_j)(1 - x_j)$$

is facet-defining for conv(K_0), where $\eta_0(z)$ is computed as:

$$\eta_0(z) = \begin{cases} -k\mu + A_k & \text{if } A_k - \mu \leq z \leq A_k, k = 1, \ldots, \bar{\ell}, \\ -k\mu + z & \text{if } A_k \leq z \leq A_{k+1} - \mu, k = 0, \ldots, \bar{\ell} - 1, \\ -\bar{\ell}\mu + A_{\bar{\ell}} & \text{if } z \geq A_{\bar{\ell}}. \end{cases}$$

Proof. Recall that we define $w = -\pi$ and $g(z) = \min\{0, b - z\}$, then K_0 is equivalent to the form of P_0. We then focus on computing the lifting function η_0. For any $S \subseteq N$ such that $\mu = b - a(S) > 0$, we have $f(S) = g(a(S)) = 0$ and

$$\rho_j(S) = \min\{0, \mu - a_j\} = \begin{cases} \mu - a_j & \text{if } j = 1, \ldots, \bar{\ell}, \\ 0 & \text{if } j = \bar{\ell} + 1, \ldots, \bar{s}. \end{cases}$$

If $A_k \le z \le A_{k+1}$ for some $k = 0, \dots, \bar{\ell} - 2$, then $\sum_{j=1}^{k} \rho_j(S) = k\mu - A_k$, and $g(a(S) + a_{k+1}) = \mu - a_{k+1}$. Based on Proposition 4.6, we have

$$\begin{aligned}\eta_0(z) &= g(A_{k+1} + a(S) - z) - k\mu + A_k - \mu + a_{k+1} \\ &= \min\{0, \mu + z - A_{k+1}\} + A_{k+1} - (k+1)\mu.\end{aligned}$$

If $A_k \le z \le A_{k+1}$ for some $k = \bar{\ell}, \dots, \bar{s} - 1$, then $\sum_{j=1}^{k} \rho_j(S) = \bar{\ell}\mu - A_{\bar{\ell}}$ and $g(a(S) + a_{k+1}) = 0$. Based on Proposition 4.6, we have

$$\begin{aligned}\eta_0(z) &= g(A_{k+1} + a(S) - z) - \bar{\ell}\mu + A_{\bar{\ell}} \\ &= \min\{0, \mu + z - A_{k+1}\} - \bar{\ell}\mu + A_{\bar{\ell}} = -\bar{\ell}\mu + A_{\bar{\ell}},\end{aligned}$$

where the third equality follows from $\mu + z - A_{k+1} \ge \mu - a_k \ge 0$.

If $z \ge A_{\bar{s}-1}$, then the derivation is omitted for brevity as it is similar to the case where $A_k \le z \le A_{k+1}$ for some $k = \bar{\ell}, \dots, \bar{s} - 1$.

■

4.4 Lifting for conv(P_{MC})

Next, we study the multidimensional lifting for conv(P) when $\mathcal{X}$ contains multiple disjoint cardinality constraints. Recall that

$$P_{MC} = \{(w, x) \in \mathbb{R} \times \{0,1\}^n : \ w \le g(a^T x), \sum_{j \in N_i} x_j \le d_i \ \forall i \in [r]\},$$

where $\{N_i\}_{i=1}^{r}$ form a partition of N. Throughout this section, we define a mapping function $\sigma : N \to [r]$ such that $\sigma(j) = i$ if $j \in N_i$ for some $i \in [r]$. For any set $S \subseteq N$, let $S_i := S \cap N_i$ for any $i \in [r]$.

In this section, we show that the multidimensional lifting on variables fixed at zeros is SI and a family of facet-defining inequalities is then derived for conv(P_{MC}) in Section 4.4.1. These results extend previous studies on the mixed 0-1 knapsack set and the single-node flow set. In Section 4.4.2, we investigate the lifting procedure on variables fixed at ones. Unfortunately, the multidimensional lifting is not SI in general, but we develop a family of strong valid inequalities for conv(P_{MC}) through subadditive approximation.

4.4.1 Lifted Inequalities From $P_{MC}(\bar{S}, \emptyset)$

Given a set $S \subseteq N$ such that $|S_i| \leq d_i$ for all $i \in [r]$, we start with the lifting for $\text{conv}(P_{MC})$ by restricting $x_j = 0$ for $j \in \bar{S}$. Let $P_{MC}(\bar{S}, \emptyset) = \{(w, x) \in P_{MC} : \; x_j = 0 \; \forall j \in \bar{S}\}$. By Proposition 4.1, the inequality $w \leq f(S) - \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j)$ defines a facet of $\text{conv}(P_{MC}(\bar{S}, \emptyset))$. To lift this seed inequality for $\text{conv}(P_{MC})$, the corresponding lifting function is given by:

$$
\begin{aligned}
\gamma\binom{z}{\mathbf{u}} = \max_{w,x} \; & w + \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j) - f(S) \\
\text{s.t. } & w \leq g(\sum_{j \in S} a_j x_j + z), \\
& \sum_{j \in S_i} x_j + \mathbf{u}_i \leq d_i \quad \forall i \in [r], \\
& x_j \in \{0, 1\} \qquad \forall j \in S,
\end{aligned}
$$

where $z \in \mathbb{R}_+$ and $\mathbf{u} \in \mathbb{Z}_+^r$. Note that if $r = n$ and $d_i = 1$ for all $i \in [r]$, then $\gamma\binom{z}{\mathbf{u}} = \gamma_0(z)$ as discussed in Section 4.3.1.

Although the value of $\mathbf{u}$ is multidimensional, the coefficients column of x_j is $\mathbf{e}_{\sigma(j)}$ for $j \in \bar{S}$; recall that the mapping function $\sigma(j) = i$ if $j \in N_i$ for some $i \in [r]$. It implies that all possible values of $\binom{z}{\mathbf{u}}$ for lifting variable x_j are of a special structure, i.e., they are in $Z_j := \mathbb{R}_+ \times \{\mathbf{e}_{\sigma(j)}\}$ for $j \in \bar{S}$. Because of this observation, a "weaker subadditivity" condition following Theorem 4.2 is sufficient to ensure the sequence independent lifting. We formalize this result in the next lemma with the proof omitted, given that it is a direct consequence of Theorem 4.2.

Lemma 4.7. *The multidimensional lifting of $w \leq f(S) - \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j)$ is SI for all possible values of $\binom{z}{\mathbf{u}} \in \mathbb{R}_+ \times \{\mathbf{e}_1, \ldots, \mathbf{e}_r\}$ if and only if for any $\Gamma \subseteq \bar{S}$ such that $|\Gamma_i| \leq d_i$, we have*

$$
\sum_{j \in \Gamma} \gamma\binom{z_j}{\mathbf{e}_{\sigma(j)}} \geq \gamma(\sum_{j \in \Gamma} \binom{z_j}{\mathbf{e}_{\sigma(j)}}) \quad \forall z_j \geq 0, j \in \Gamma. \tag{4.20}
$$

By showing that lifting function $\gamma\binom{z}{\mathbf{u}}$ satisfies (4.20) in Lemma 4.7, we have the following result as the main one of this subsection.

Theorem 4.6. *If $|S_i| \le d_i$ for some $S \subseteq N$ and all $i \in [r]$, then the inequality*

$$w \le f(S) - \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j) + \sum_{j \in \bar{S}} \gamma \binom{a_j}{\mathbf{e}_{\sigma(j)}} x_j \tag{4.21}$$

is facet-defining for $conv(P_{MC})$.

Proof. See Appendix . ■

Next, we discuss how to compute $\gamma\binom{z}{\mathbf{u}}$, and we introduce a simpler problem that replaces the cardinality constraints by fixing a subset of variables. Given $T \subseteq S$, define

$$\gamma_T(z) = \max \left\{ \begin{array}{r} w + \sum\limits_{j \in S} \rho_j(S \setminus j)(1 - x_j) - f(S) : \; x_j = 0 \; \forall j \in T, \\ w \le g(\sum\limits_{j \in S} a_j x_j + z), x_j \in \{0,1\} \; \forall j \in S \end{array} \right\}. \tag{4.22}$$

Note that if $T = \emptyset$, then $\gamma_\emptyset(z) = \gamma_0(z)$. If $|T_i| \ge |S_i| + \mathbf{u}_i - d_i$ for all $i \in [r]$, then problem (4.22) can also be interpreted as the optimization problem for the lifting function $\gamma\binom{z}{\mathbf{u}}$ when we fix variables in T to 0. Consequently, $\gamma\binom{z}{\mathbf{u}}$ is reducible to solving a set of subproblems given by $\gamma_T(z)$. Specifically, for any $\Gamma \subseteq \bar{S}$, we have that

$$\gamma \binom{z}{\sum_{j \in \Gamma} \mathbf{e}_{\sigma(j)}} = \max_{T \subseteq S} \Big\{ \gamma_T(z) : |T_i| = \max\{0, |S_i| + |\Gamma_i| - d_i\} \; \forall i \in [r] \Big\}, \tag{4.23}$$

and our question of interest now is how to compute $\gamma_T(z)$. Next, we assume that $S = \{1, \dots, s\}$ such that $a_1 \ge \dots \ge a_s$. Let $A_0 = 0$ and $A_k = \sum_{j=1}^k a_j$ for $k \in S$.

Lemma 4.8. *Suppose $T = \{\ell_1, \dots, \ell_{|T|}\} \subseteq S$ such that $\ell_1 < \dots < \ell_{|T|}$. Let $A_t^T = a(T) - \sum_{j=1}^t a_{\ell_j}$ for $t = 1, \dots, |T|$, and $A_0^T = a(T)$, $\ell_0 = 0$. Then $\gamma_T(z)$ is computed as follows:*

Case 1 If $0 \le z \le a(T)$, then

$$\gamma_T(z) = g(a(S) - a(T) + z) + \sum_{j \in T} \rho_j(S \setminus j) - f(S).$$

Case 2 If $A_k + A_t^T \le z \le A_{k+1} + A_t^T$ for $k = \ell_t, \dots, \ell_{t+1} - 2$ and $t = 0, 1, \dots, |T| - 1$, then

$$\gamma_T(z) = g(a(S) - A_{k+1} - A_t^T + z) + \sum_{j \in [k+1] \cup T} \rho_j(S \setminus j) - f(S).$$

Case 3 If $A_k \leq z \leq A_{k+1}$ for $k = \ell_{|T|}, \ldots, s-2$, then

$$\gamma_T(z) = g(a(S) - A_{k+1} + z) + \sum_{j \in [k+1]} \rho_j(S \setminus j) - f(S).$$

Case 4 If $z \geq A_{s-1}$, then

$$\gamma_T(z) = g(z) + \sum_{j \in S} \rho_j(S \setminus j) - f(S).$$

Proof. Recall from Section 4.3.1 that for any $\Lambda \subseteq S$, we denote the objective function value for z and x_Λ of the lifting problem as $h(z, \Lambda) = g(a(\Lambda) + z) + \sum_{j \in S \setminus \Lambda} \rho_j(S \setminus j) - f(S)$.

Case 1 If $0 \leq z \leq a(T)$, then for any $\Lambda \subsetneq S \setminus T$, we have $a(\Lambda) + z \leq a(S)$, which implies that $h(z, \Lambda) \leq h(z, S \setminus T)$ based on Lemma 4.3. Therefore, $S \setminus T$ is an optimal solution set to compute $\gamma_T(z)$ and the result follows.

Case 2 If $A_k + A_t^T < z \leq A_{k+1} + A_t^T$ for $k = \ell_t, \ldots, \ell_{t+1} - 2$ and $t = 0, 1, \ldots, |T| - 1$, then it suffices to show that $\{k+2, \ldots, s\} \setminus T$ is an optimal solution set to compute $\gamma_T(z)$. Suppose $\Lambda^* \subseteq S \setminus T$ is an optimal solution set for $\gamma_T(z)$. Note that based on Lemma 4.3, there exists Λ^* such that $a(\Lambda^*) + z \leq a(S)$ and $a(\Lambda^*) + a_j + z > a(S)$ for all $j \in S \setminus (T \cup \Lambda^*)$. Therefore, if $\{k+2, \ldots, s\} \setminus T \subseteq \Lambda^*$, then $\Lambda^* = \{k+2, \ldots, s\} \setminus T$.

Suppose $\{k+2, \ldots, s\} \setminus T \nsubseteq \Lambda^*$. Let k_0 be the largest index such that $k_0 \notin \Lambda^* \cup T$, then $k_0 \geq k+2$. We next construct a new optimal solution set $\tilde{\Lambda}$ such that $\{k_0, \ldots, s\} \setminus T \subseteq \tilde{\Lambda}$. Since $a(\Lambda^*) + a_{k_0} + z > a(S)$, there must exist some $j_0 \in \Lambda^*$ such that $j_0 < k_0$. It follows from Lemma 4.4 that $h(z, \Lambda^* \cup k_0 \setminus j_0) \geq h(z, \Lambda^*)$ because of $a_{j_0} \geq a_{k_0}$. Thus, $\tilde{\Lambda} = \Lambda^* \cup k_0 \setminus j_0$ is also an optimal solution set. Proceeding in a recursive manner, we can conclude that $\{k+2, \ldots, s\} \setminus T$ is an optimal solution.

Case 3 If $A_k \leq z \leq A_{k+1}$ for some $k = \ell_{|T|}, \ldots, s-2$, then similar to *Case 2*, we can show that $\{k+2, \ldots, s\}$ is an optimal solution set to compute $\gamma_T(z)$.

Case 4 If $z \geq A_{s-1}$, then for any nonempty set $\Lambda \subseteq S \setminus T$, $a(\Lambda) + z \geq a_s + z \geq a(S)$. It immediately follows that $h(z, \Lambda) \leq h(z, \emptyset)$ based on Lemma 4.3. Therefore, $\emptyset$ is an optimal solution set to compute $\gamma_T(z)$ and the result follows.

■

Observe that $\gamma_T(z) = \gamma_0(z)$ when $z \geq A_{\ell_{|T|}}$ for any $T \subseteq S$. Also, function $\gamma_T(z)$ is nonincreasing on T, that is $\gamma_T(z) \leq \gamma_{T'}(z) \leq \gamma_0(z)$ if $T' \subseteq T$.

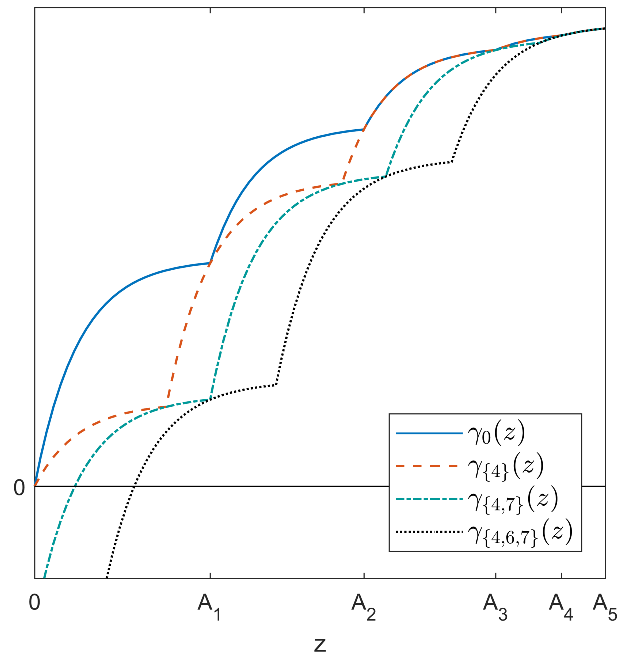


Figure 8: The functions $\gamma_T(z)$ for Example 4.3.

Example 4.3. *Consider* $P_{MC} = \{(w, x) \in \mathbb{R} \times \{0, 1\}^7 : \ w \leq -\exp(-a^T x), x_3 + x_5 + x_6 \leq 2, x_1 + x_4 + x_7 \leq 2\}$, *where* $n = 7$ *and* $a = (1, 0.8, 0.7, 0.6, 0.5, 0.3, 0.2)^T$. *Let* $S = \{2, 3, 4, 6, 7\}$. *The functions* $\gamma_T(z)$ *are depicted in Figure 8. In Section 4.3.1, we show that* $\gamma_0(z)$ *is a subadditive function on* $\mathbb{R}_+$. *However, we can see from Figure 8 that function* $\gamma_T(z)$ *is not subadditive on* $z \in \mathbb{R}_+$ *in general. One numerical example is also provided as follows: for* $\gamma_{\{4\}}(z)$, *we have* $\gamma_{\{4\}}(0.4) + \gamma_{\{4\}}(0.5) = 0.0446 + 0.0533 < 0.1039 = \gamma_{\{4\}}(0.9)$.

Next, we further simplify computing $\gamma\binom{z}{\mathbf{e}_i}$ using the following observation.

Proposition 4.7. *For any* $i \in [r]$, *assume* $S_i = \{i_1, \ldots, i_\ell\}$ *such that* $a_{i_1} \geq \cdots \geq a_{i_\ell}$. *If* $|S_i| < d_i$, *then* $\gamma\binom{z}{\mathbf{e}_i} = \gamma_0(z)$. *If* $|S_i| = d_i$, *then*

$$\gamma\begin{pmatrix} z \\ \mathbf{e}_i \end{pmatrix} = \max\{\gamma_{\{i_1\}}(z), \gamma_{\{i_\ell\}}(z)\}.$$

Proof. First, by (4.23) we have

$$\gamma\binom{z}{\mathbf{e}_i} = \max_{T \subseteq S_i}\{\gamma_T(z) : |T| = \max\{0, |S_i| + 1 - d_i\}\}.$$

If $|S_i| < d_i$, then it is clear that the result follows. If $|S_i| = d_i$, then $\gamma\binom{z}{\mathbf{e}_i} = \max_{i_j \in S_i}\{\gamma_{\{i_j\}}(z)\}$. It follows that $\max\{\gamma_{\{i_1\}}(z), \gamma_{i_\ell}(z))\} \leq \gamma\binom{z}{\mathbf{e}_i}$. Assume $\Lambda^* \subseteq S$ such that x_{Λ^*} is an optimal solution to compute $\gamma\binom{z}{\mathbf{e}_i}$. Note that $\Lambda^* \neq S_i$ due to the cardinality constraints $\sum_{j \in S_i} x_j \leq d_i - 1$ and $|S_i| = d_i$.

Suppose $i_1, i_\ell \in \Lambda^*$ and $i_j \in S_i \setminus \Lambda^*$, then we can construct a new optimal solution set $\tilde{\Lambda}$ such that either $i_1 \notin \tilde{\Lambda}$ or $i_\ell \notin \tilde{\Lambda}$. If $a(\Lambda^*) + a_{i_j} + z \leq a(S)$, then based on Lemma 4.4, we have $h(z, \Lambda^* \cup i_j \setminus i_\ell) \geq h(z, \Lambda^*)$ as $a_{i_j} \geq a_{i_\ell}$. It follows that $\tilde{\Lambda} = \Lambda^* \cup i_j \setminus i_\ell$ is also an optimal set and $i_\ell \notin \tilde{\Lambda}$. Otherwise, if $a(\Lambda^*) + a_{i_j} + z \geq a(S)$, then based on Lemma 4.4, we have $h(z, \Lambda^* \cup i_j \setminus i_1) \geq h(z, \Lambda^*)$ as $a_{i_1} \geq a_{i_j}$. It follows that $\tilde{\Lambda} = \Lambda^* \cup i_j \setminus i_1$ is also an optimal set and $i_1 \notin \tilde{\Lambda}$.

Hence, there exists an optimal set Λ^* such that either $i_1 \notin \Lambda^*$ or $i_\ell \notin \Lambda^*$. It yields that $\gamma\binom{z}{\mathbf{e}_i} = h(z, \Lambda^*) \leq \max\{\gamma_{\{i_1\}}(z), \gamma_{\{i_\ell\}}(z)\}$. ■

Example 3 (Cont.) Let $S = \{2, 3, 4, 6, 7\}$. To lift variable x_1, based on Proposition 4.7 the corresponding lifting function (depicted in Figure 9(a)) is given by

$$\gamma\binom{z}{\mathbf{e}_2} = \max\{\gamma_{\{4\}}(z), \gamma_{\{7\}}(z)\}.$$

If the cardinality constraints are ignored, then the inequality (4.16) for $\text{conv}(P_0)$ is:

$$\begin{aligned} w \leq & -0.3441 + 0.1181x_1 + 0.091x_2 + 0.0753x_3 + 0.0611x_4 \\ & + 0.065x_5 + 0.026x_6 + 0.0164x_7. \end{aligned}$$

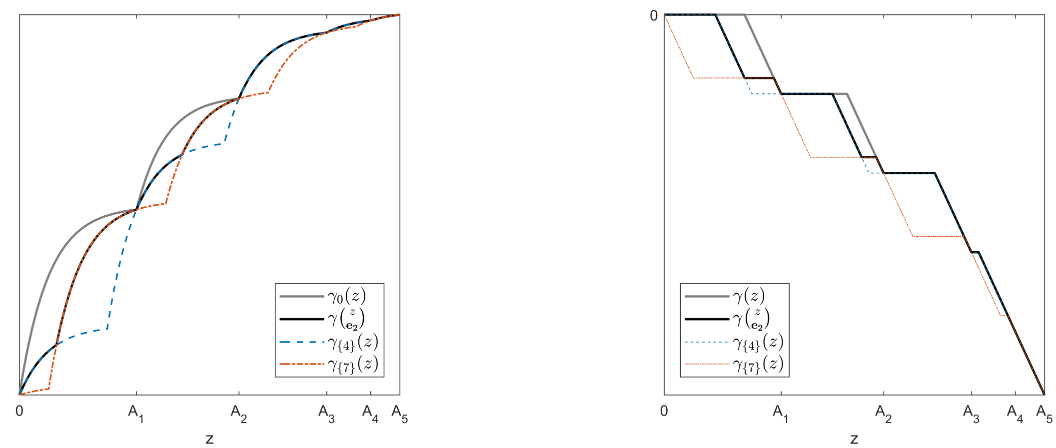
In contrast, the lifted inequality (4.21) for $\text{conv}(P_{MC})$ is given by:

$$\begin{aligned} w \leq & -0.3441 + \mathbf{0.1156}x_1 + 0.091x_2 + 0.0753x_3 + 0.0611x_4 \\ & + \mathbf{0.0589}x_5 + 0.026x_6 + 0.0164x_7. \end{aligned}$$

Remark 4.1. *Below numerical example shows that lifting function* $\gamma\binom{z}{\mathbf{u}}$ *is not subadditive on* $\binom{z}{\mathbf{u}} \in \mathbb{R}_+ \times \mathbb{Z}_+^r$ *in general:*

$$\gamma\begin{pmatrix} 2 \\ \mathbf{e}_1 \end{pmatrix} + \gamma\begin{pmatrix} 0.2 \\ \mathbf{e}_1 + \mathbf{e}_2 \end{pmatrix} = 0.2196 + 0.0164 < 0.2369 = \gamma\begin{pmatrix} 2.2 \\ 2\mathbf{e}_1 + \mathbf{e}_2 \end{pmatrix}.$$

Therefore, the subadditivity as a sufficient condition to ensure SI lifting, is not applicable to prove Theorem 4.6.



(a) $g(z) = -\exp(-z)$.

(b) $g(z) = \min\{0, b - z\}$.

Figure 9: The lifting functions $\gamma\binom{z}{\mathbf{u}}$ for Examples 4.3 and 4.4.

We next discuss several immediate implications of Theorem 4.6 and Proposition 4.7 that extend and strengthen some previous works. If $N_1 = N$ and $d_1 = d$, then let $P_C = \{(w, x) \in \{0,1\}^n : w \leq g(a^T x), \sum_{j \in N} x_j \leq d\}$. In [122], Yu and Ahmed provide a subadditive approximation by solving the continuous relaxation of the lifting problem for P_C. Here, as the consequence of Theorem 4.6, we show that the lifting procedure is already SI and the approximations are not required.

Corollary 4.5. *If* $|S| \leq d$ *for any* $S \subseteq N$*, then the inequality* $w \leq f(S) - \sum_{j \in S} \rho_j(S \setminus j)(1 - x_j) + \sum_{j \in \bar{S}} \gamma_0(a_j) x_j$ *defines a facet of conv*(P_C).

Proof. Based on Theorem 4.6, we have that for any $S \subseteq N$ such that $|S| \leq d$, $w \leq f(S) - \sum_{j \in S} \rho_j(S \setminus j)(1-x_j) + \sum_{j \in \bar{S}} \gamma\binom{a_j}{1} x_j$ defines a facet of conv(P_C). Thus, it remains to show that $\gamma\binom{z}{1} = \gamma_0(z)$. Suppose $S = \{1, \ldots, s\}$ such that $a_1 \geq \cdots \geq a_s$. Then based on Proposition 4.7, we have

$$\gamma\begin{pmatrix} z \\ 1 \end{pmatrix} = \max\{\gamma_{\{1\}}(z), \gamma_{\{s\}}(z)\}.$$

By Lemma 4.8, we have $\gamma_{\{1\}}(z) = \gamma_0(z)$. Since $\gamma_T(z) \leq \gamma_0(z)$ for any $T \subseteq S$, it yields the result.

■

Denote by K_{MC} and F_{MC} the mixed 0-1 knapsack set K and single-node flow set F when $\mathcal{X}$ contains multiple disjoint cardinality constraints, respectively. Then our results can be extended to obtain a family of facet-defining inequalities for conv(K_{MC}). Note that the polyhedral study of K_{MC} has not been provided in the related literature (to the best of our knowledge). Meantime, the valid inequalities for conv(F_{MC}) can be derived through the reduction from those for conv(K_{MC}).

Corollary 4.6. *Consider a set $S \subseteq N$ and $\lambda = a(S) - b > 0$. Suppose $S = \{1, \ldots, s\}$ such that $a_1 \geq \cdots \geq a_\ell > \lambda \geq \cdots \geq a_s$. If $|S_i| \leq d_i$ for all $i \in [r]$, then the inequality*

$$-\pi \leq -\lambda + \sum_{j=1}^{\ell} \lambda(1-x_j) + \sum_{j=\ell+1}^{s} a_j(1-x_j) + \sum_{j \in \bar{S}} \gamma\begin{pmatrix} a_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix} x_j$$

is facet-defining for conv(K_{MC}), where $S_i = \{i_1, \ldots, i_\ell\}$ such that $i_1 < \cdots < i_\ell$, and

$$\gamma\begin{pmatrix} z \\ \mathbf{e}_i \end{pmatrix} = \begin{cases} \gamma_0(z) & \text{if } |S_i| < d_i, \\ \max\{\gamma_{\{i_1\}}(z), \gamma_{i_\ell}(z)\} & \text{if } |S_i| = d_i, \end{cases}$$

where $\gamma_0(z)$ is given in Corollary 4.3. For $j_0 \in \{i_1, i_\ell\}$ we have that if $j_0 \leq \ell$, then

$$\gamma_{\{j_0\}}(z) = \begin{cases} \min\{0, a_{j_0} - \lambda - z\} & \text{if } 0 \leq z \leq a_{j_0}, \\ -k\lambda + \min\{0, A_{k+1} + a_{j_0} - \lambda - z\} & \text{if } A_k + a_{j_0} \leq z \leq A_{k+1} + a_{j_0}, \\ & \quad k = 0, \ldots, j_0 - 2, \\ \gamma_0(z) & \text{if } z \geq A_{j_0}, \end{cases}$$

and if $j_0 > \ell$, then

$$\gamma_{\{j_0\}}(z) = \begin{cases} -z & \text{if } 0 \le z \le a_{j_0}, \\ -(k-1)\lambda - a_{j_0} & \text{if } A_k + a_{j_0} \le z \le A_{k+1} + a_{j_0} - \lambda, \\ & \quad k = 0, \dots, \ell - 1, \\ -k\lambda + A_{k+1} - z & \text{if } A_{k+1} + a_{j_0} - \lambda \le z \le A_{k+1} + a_{j_0}, \\ & \quad k = 0, \dots, \ell - 1, \\ -\ell\lambda + A_\ell - z & \text{if } z \ge A_\ell. \end{cases}$$

Proof. Similar to the proof in Corollary 4.3, it directly follows from Theorem 4.6 and Proposition 4.7. ■

Example 4.4. *Consider $K_{MC} = \{(\pi, x) \in \mathbb{R}_+ \times \{0,1\}^n : a^T x \le b + \pi, x_3 + x_5 + x_6 \le 2, x_1 + x_4 + x_7 \le 2\}$, where $n = 7$ and $a = (1, 0.8, 0.7, 0.6, 0.5, 0.3, 0.2)^T$. Let $S = \{2, 3, 4, 6, 7\}$, then the lifting function $\gamma\binom{z}{\mathbf{e}_2}$ is depicted in Figure 9(b).*

4.4.2 Lifted Inequalities From $P_{MC}(\emptyset, S)$

In this section, we describe valid inequalities for $\text{conv}(P_{MC})$ through lifting procedure from $P_{MC}(\emptyset, S)$. Given a set $S \subseteq N$ such that $|S_i| \le d_i$ for all $i \in [r]$, the lifting function of seed inequality $w \le f(S) + \sum_{j \in \bar{S}} \rho_j(S) x_j$ is given by:

$$\begin{aligned} \eta\binom{z}{\mathbf{u}} = \max_{w,x} \; & w - \sum_{j \in \bar{S}} \rho_j(S) x_j - f(S) \\ \text{s.t. } & w \le g\Big(\sum_{j \in \bar{S}} a_j x_j + a(S) - z\Big), \\ & \sum_{j \in \bar{S}_i} x_j - \mathbf{u}_i \le d_i - |S_i| \quad \forall i \in [r], \\ & x_j \in \{0,1\} \qquad \forall j \in \bar{S}, \end{aligned}$$

where $z \in \mathbb{R}_+$ and $\mathbf{u} \in \mathbb{Z}_+^r$. Note that if $\mathbf{u}_i \ge |N_i| - d_i$ for all $i \in [r]$, then the cardinality constraints in the above problem can be removed, and $\eta\binom{z}{\mathbf{u}} = \eta_0(z)$. Similar to the discussion on $\gamma\binom{z}{\mathbf{u}}$, we only need to consider the value of $\mathbf{u} = \mathbf{e}_i$ for $i \in [r]$. The sufficient and necessary conditions for SI lifting is derived below based on Theorem 4.2; the proof is omitted as it follows directly.

Lemma 4.9. *The lifting of $w \le f(S) + \sum_{j\in\bar{S}} \rho_j(S)x_j$ is SI for all possible values of $\binom{z}{\mathbf{u}} \in \mathbb{R}_+ \times \{\mathbf{e}_1, \ldots, \mathbf{e}_r\}$ if and only if for any $\Gamma \subseteq S$*

$$\sum_{j\in\Gamma} \eta\begin{pmatrix} z_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix} \ge \eta(\sum_{j\in\Gamma} \begin{pmatrix} z_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix}) \quad \forall z_j \ge 0, j \in \Gamma. \tag{4.24}$$

To verify condition (4.24), our first question of interest is to compute $\eta\binom{z}{\mathbf{u}}$. We introduce a simpler optimization problem as given below:

$$\eta_T(z) = \max \left\{ \begin{aligned} & w - \sum_{j\in\bar{S}} \rho_j(S)x_j - f(S) : \ x_j = 0 \ \forall j \in T, \\ & \qquad\qquad w \le g(\sum_{j\in\bar{S}} a_j x_j + a(S) - z), x_j \in \{0,1\} \ \forall j \in \bar{S} \end{aligned} \right\}, \tag{4.25}$$

where T is a subset of $\bar{S}$. Note that if $T = \emptyset$, then $\eta_\emptyset(z) = \eta_0(z)$. In this section, we assume that $\bar{S} = \{1, 2, \ldots, \bar{s}\}$ such that $a_1 \ge \cdots \ge a_{\bar{s}}$. Let $A_0 = 0$ and $A_k = \sum_{j=1}^{k} a_j$ for $k \in \bar{S}$.

Lemma 4.10. *Suppose $T = \{\ell_1, \ldots, \ell_{|T|}\} \subseteq \bar{S}$ such that $\ell_1 < \cdots < \ell_{|T|}$. Let $A_t^T = \sum_{j=1}^{t} a_{\ell_j}$ for $t = 1, \ldots, |T|$, and $A_0^T = 0, \ell_0 = 0$. Then $\eta_T(z)$ is computed as follows:*

Case 1. *If $A_k - A_t^T \le z \le A_{k+1} - A_t^T$ for $k = \ell_t, \ldots, \ell_{t+1} - 2$, and $t = 0, 1, \ldots, |T| - 1$, then*

$$\eta_T(z) = g(a(S) + A_{k+1} - A_t^T - z) - \sum_{j\in[k+1]\setminus T} \rho_j(S) - f(S).$$

Case 2. *If $A_k - a(T) \le z \le A_{k+1} - a(T)$ for $k = \ell_{|T|}, \ldots, \bar{s} - 2$, then*

$$\eta_T(z) = g(a(S) + A_{k+1} - a(T) - z) - \sum_{j\in[k+1]\setminus T} \rho_j(S) - f(S).$$

Case 3. *If $z \ge A_{\bar{s}-1} - a(T)$, then*

$$\eta_T(z) = g(a(N \setminus T) - z) - \sum_{j\in\bar{S}\setminus T} \rho_j(S) - f(S).$$

Furthermore, $\eta_T(z)$ is subadditive on $z \in \mathbb{R}_+$.

Proof. Let $N' = N \setminus T, S' = S$, then $\bar{S}' = \bar{S} \setminus T$. Consider

$$P_0' = \{(w, x) \in \mathbb{R} \times \{0,1\}^{|N'|} : w \le g(a^T x)\}.$$

Then we note that $\eta_T(z)$ is the lifting function of seed inequality $w \le f(S') + \sum_{j \in \bar{S}'} \rho_j(S')x_j$ for $\operatorname{conv}(P_0')$. Therefore, we can compute $\eta_T(z)$ following from Proposition 4.6 and conclude that $\eta_T(z)$ is subadditive on $\mathbb{R}_+$.

■

Proposition 4.8. *Suppose $\bar{S}_i = \{i_1, \ldots, i_{\bar{s}_i}\}$ such that $i_1 < \cdots < i_{\bar{s}_i}$ for all $i \in [r]$. For any $\Gamma \subseteq S$, let $d_i' = \min\{|\bar{S}_i|, d_i - |S_i| + |\Gamma_i|\}$ and*

$$T = \cup_{i \in [r]} \{i_{d_i'+1}, \ldots, i_{\bar{s}_i}\}.$$

Then $\eta\binom{z}{\sum_{j\in\Gamma} \mathbf{e}_{\sigma(j)}} = \eta_T(z)$.

Proof. It is sufficient to show that there exists an optimal solution set $\Lambda^* \subseteq \bar{S} \setminus T$ to compute $\eta\binom{z}{\sum_{j\in\Gamma} \mathbf{e}_{\sigma(j)}}$. Suppose not, let Λ_0 be an optimal solution set for $\eta\binom{z}{\sum_{j\in\Gamma} \mathbf{e}_{\sigma(j)}}$. Assume $j_0 \in \Lambda_0 \cap T$, and $j_0 \in N_{i_0}$ for some $i_0 \in [r]$. By Lemma 4.5, we can find Λ_0 such that $a(\Lambda_0) - a_{j_0} \le z$

Note that there exists at least one $j_1 \in N_{i_0}$ and $a_{j_1} \ge a_{j_0}$ such that $j_1 \notin \Lambda_0$ due to the cardinality constraint $\sum_{j \in \bar{S}_{i_0}} x_j \le d_{i_0} - |S_{i_0}| + |\Gamma_{i_0}|$. Consider $\tilde{\Lambda} = \Lambda_0 \cup j_1 \setminus j_0$. It follows from Lemma 4.6 that $\chi(z, \tilde{\Lambda}) \ge \chi(z, \Lambda_0)$ as $a_{j_1} \ge a_{j_0}$. Therefore, $\tilde{\Lambda}$ is also an optimal solution to compute $\eta\binom{z}{\sum_{j\in\Gamma} \mathbf{e}_{\sigma(j)}}$. Then the result follows after the iterative construction. ■

Unfortunately, the following example indicates that the lifting procedure of seed inequality $w \le f(S) + \sum_{j \in \bar{S}} \rho_j(S)x_j$ for $\operatorname{conv}(P_{MC})$ might be sequence dependent in some cases.

Example 4.5. *Consider $P_{MC} = \{(w, x) \in \mathbb{R} \times \{0,1\}^7 : \; w \le -\exp(-a^T x), \sum_{j=1}^7 x_j \le 4\}$, where $n = 7$ and $a = (0.8, 0.7, 0.6, 0.4, 2.2, 0.1)^T$. Let $S = \{5, 6\}$, then we have*

$$\eta\binom{2.2}{\mathbf{e}_1} + \eta\binom{0.1}{\mathbf{e}_1} = -0.1615 - 0.0047 < \eta\binom{2.3}{2\mathbf{e}_1} = -0.1658.$$

In view of the above example, we focus next on approximating η by subadditive functions with some quality guarantees, which are consequently exploited to derive valid inequalities for P_{MC}.

Denote by $Z = \mathbb{R}_+ \times \{0, 1, \ldots, |S_1|\} \times \cdots \times \{0, 1, \ldots, |S_r|\}$ the domain of $\eta\binom{z}{\mathbf{u}}$.

Definition 4.2. *We say function $\psi : Z \to \mathbb{R}$ is a subadditive valid approximation of $\eta\binom{z}{\mathbf{u}}$ if $\psi\binom{z}{\mathbf{u}} \geq \eta\binom{z}{\mathbf{u}}$ for all $\binom{z}{\mathbf{u}} \in Z$ and*

$$\sum_{j\in\Gamma} \psi\begin{pmatrix} z_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix} \geq \psi(\sum_{j\in\Gamma} \begin{pmatrix} z_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix}) \quad \forall z_j \geq 0, \Gamma \subseteq S.$$

To compare the strengths of subadditive valid approximate functions, Gu et al. [45] develop two criteria, namely, non-dominance and maximality. Zeng and Richard [129] further generalize these concepts for multidimensional lifting procedure. We follow these two criteria to describe a "good" quality subadditive valid approximation.

Let $\underline{Z} = \mathbb{R}_+ \times \{\mathbf{e}_1, \ldots, \mathbf{e}_r\}$ and $E = \{\binom{z}{\mathbf{u}} \in \underline{Z} : \eta\binom{z}{\mathbf{u}} = \eta_j\binom{z}{\mathbf{u}}$ for all $a_j \geq 0,$ and all $j \in S,$ and all lifting sequences$\}$. Then we define:

Definition 4.3. *A subadditive valid approximate function ψ is non-dominated if there does not exist any valid subadditive approximation ψ' such that $\psi\binom{z}{\mathbf{u}} \geq \psi'\binom{z}{\mathbf{u}}$ for all $\binom{z}{\mathbf{u}} \in \underline{Z}$, and $\psi\binom{z_0}{\mathbf{u}_0} > \psi'\binom{z_0}{\mathbf{u}_0}$ for some $\binom{z_0}{\mathbf{u}_0} \in \underline{Z}$.*

Definition 4.4. *A subadditive valid approximate function ψ is maximal if $\psi\binom{z}{\mathbf{u}} = \eta\binom{z}{\mathbf{u}}$ for all $\binom{z}{\mathbf{u}} \in E$.*

Proposition 4.9. *The subadditive valid approximate function $\psi\binom{z}{\mathbf{u}} = \eta\binom{z}{\mathbf{u}_0}$ for all $\binom{z}{\mathbf{u}} \in Z$, where $\mathbf{u}_0 = \sum_{i\in[r]} \lambda_i \mathbf{e}_i$ and $\lambda_i = \min\{|S_i|, |N_i| - d_i\}$ is non-dominated and maximal.*

Proof. It is clear that $\psi\binom{z}{\mathbf{u}} \geq \eta\binom{z}{\mathbf{u}}$ for all $\binom{z}{\mathbf{u}} \in Z$. By Lemma 4.10, we have that $\eta\binom{z}{\mathbf{u}_0}$ is subadditive on $\mathbb{R}_+$. Thus, for any $\Gamma \subseteq S$ and $z_j \geq 0$, we have

$$\sum_{j\in\Gamma} \psi\begin{pmatrix} z_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix} = \sum_{j\in\Gamma} \eta\begin{pmatrix} z_j \\ \mathbf{u}_0 \end{pmatrix} \geq \eta\begin{pmatrix} z(\Gamma) \\ \mathbf{u}_0 \end{pmatrix} = \psi\begin{pmatrix} z(\Gamma) \\ \sum_{j\in\Gamma} \mathbf{e}_{\sigma(j)} \end{pmatrix}.$$

Therefore, $\psi\binom{z}{\mathbf{u}}$ is a subadditive valid approximation for η.

We next prove that $\psi\binom{z}{\mathbf{u}}$ is non-dominated. Suppose $\psi\binom{z}{\mathbf{u}}$ is dominated by another subadditive valid approximation $\psi'\binom{z}{\mathbf{u}}$. Firstly, note that $\psi\binom{0}{\mathbf{u}} = \eta\binom{0}{\mathbf{u}_0} = 0$ and $0 = \eta\binom{0}{\mathbf{u}} \leq$

$\psi'\binom{0}{\mathbf{u}} \leq \psi\binom{0}{\mathbf{u}}$, which yields that $\psi'\binom{0}{\mathbf{u}} = 0$ for all $\mathbf{u}$. Suppose there exists $\binom{z_0}{\mathbf{e}_{i_0}} \in \underline{Z}$ such that $\psi\binom{z_0}{\mathbf{e}_{i_0}} > \psi'\binom{z_0}{\mathbf{e}_{i_0}}$.

If $\lambda_{i_0} = 0$, then $d_i = |N_i|$ and $\eta\binom{z}{\mathbf{e}_{i_0}} = \eta_0(z)$. Therefore, $\psi'\binom{z_0}{\mathbf{e}_{i_0}} \geq \eta\binom{z}{\mathbf{e}_{i_0}} = \eta_0(z) \geq \eta\binom{z_0}{\mathbf{u}_0}$, where the last inequality follows from $\eta\binom{z}{\mathbf{u}} \leq \eta_0(z)$ for all $\binom{z}{\mathbf{u}} \in Z$. On the other hand, if $\lambda_{i_0} > 0$, then based on Definition 4.2 of ψ', we have $\sum_{i\in[r]\setminus i_0} \lambda_i \psi'\binom{0}{\mathbf{e}_i} + (\lambda_{i_0} - 1)\psi'\binom{0}{\mathbf{e}_{i_0}} + \psi'\binom{z_0}{\mathbf{e}_{i_0}} \geq \psi'\binom{z_0}{\mathbf{u}_0} \geq \eta\binom{z_0}{\mathbf{u}_0}$. It immediately follows that $\psi'\binom{z_0}{\mathbf{e}_{i_0}} \geq \eta\binom{z_0}{\mathbf{u}_0}$.

Since $\psi\binom{z_0}{\mathbf{e}_{i_0}} = \eta\binom{z_0}{\mathbf{u}_0}$ based on the construction of ψ, we have

$$\eta\begin{pmatrix} z_0 \\ \mathbf{u}_0 \end{pmatrix} = \psi\begin{pmatrix} z_0 \\ \mathbf{e}_{i_0} \end{pmatrix} > \psi'\begin{pmatrix} z_0 \\ \mathbf{e}_{i_0} \end{pmatrix} \geq \eta\begin{pmatrix} z_0 \\ \mathbf{u}_0 \end{pmatrix},$$

which is a contradiction. It follows that ψ is a non-dominated approximation.

Finally, we need to show that $\psi\binom{z}{\mathbf{u}}$ is maximal. Suppose $\psi\binom{z_0}{\mathbf{e}_{i_0}} > \eta\binom{z_0}{\mathbf{e}_{i_0}}$ for some $\binom{z_0}{\mathbf{e}_{i_0}} \in E$, which implies that $\eta\binom{z_0}{\mathbf{u}_0} > \eta\binom{z_0}{\mathbf{e}_{i_0}}$. For any $j_0 \in S_{i_0}$, let $a_{j_0} = z_0$, and $a_j = 0$ for other $j \in S$. Consider the lifting process that lifts j_0 in the last step and thus, based on equation (4.10), we have

$$\begin{aligned} \eta_{|S|}\begin{pmatrix} z_0 \\ \mathbf{e}_{i_0} \end{pmatrix} &\geq \max\left\{\eta\begin{pmatrix} z_0 \\ \mathbf{e}_{i_0} \end{pmatrix}, \eta\begin{pmatrix} z_0 \\ \sum_{j\in S} \mathbf{e}_{\sigma(j)} \end{pmatrix} - \sum_{j\in S} \eta\begin{pmatrix} 0 \\ \mathbf{e}_{\sigma(j)} \end{pmatrix}\right\} \\ &\geq \eta\begin{pmatrix} z_0 \\ \sum_{j\in S} \mathbf{e}_{\sigma(j)} \end{pmatrix} = \eta\begin{pmatrix} z_0 \\ \sum_{i\in[r]} |S_i|\mathbf{e}_i \end{pmatrix} \\ &\geq \eta\begin{pmatrix} z_0 \\ \mathbf{u}_0 \end{pmatrix} > \eta\begin{pmatrix} z_0 \\ \mathbf{e}_{i_0} \end{pmatrix}, \end{aligned}$$

where the second inequality follows from $\eta\binom{0}{\mathbf{u}} = 0$ and the third inequality follows from the fact that $\eta\binom{z}{\mathbf{u}}$ is monotone increasing on $\mathbf{u}$ for a fixed z. Therefore, we obtain a contradiction with our initial assumption that $\binom{z_0}{\mathbf{e}_{i_0}} \in E$. ■

Theorem 4.7. *Consider* $\psi\binom{z}{\mathbf{u}} = \eta\binom{z}{\mathbf{u}_0}$ *for all* $\binom{z}{\mathbf{u}} \in Z$, *where* $\mathbf{u}_0 = \sum_{i\in[r]} \lambda_i \mathbf{e}_i$ *and* $\lambda_i = \min\{|S_i|, |N_i| - d_i\}$. *Then the inequality*

$$w \leq f(S) + \sum_{j\in\bar{S}} \rho_j(S)x_j + \sum_{j\in S} \psi\begin{pmatrix} a_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix}(1 - x_j) \tag{4.26}$$

is valid for conv(P_{MC}).

Proof. It directly follows from $\psi\binom{z}{\mathbf{u}} \geq \eta\binom{z}{\mathbf{u}}$ for all $\binom{z}{\mathbf{u}} \in Z$ and Proposition 4.9. ■

4.5 Computational Experiments

In our computational study we evaluate the lifted inequalities for conv(P_0) and conv(P_{MC}). Our experiments are performed using Gurobi 8.1.1 [47] on a Windows 10 PC with a 3.2 GHz CPU and 8 GB of RAM. The time limit is set to 1800 seconds and the relative MIP optimality gap is set to 0.01%.

For each fixed instance size (i.e., each row in any of the tables in this section), we randomly generate 10 instances and report the solver's average performance in all tables. In particular, for those instances solved to optimality within the time limit, we report their average solution time in seconds (under column "Time"), the number of added cuts (under column "Cuts"), and the number of explored branch-and-cut nodes (under column "Nodes"). For instances that cannot be solved to optimality within the time limit, we show their average relative gaps between the best known upper bound and the lower bound at termination ("Egap"); also, the number of unsolved instances (out of 10) is presented as the superscript over the gap value.

Next, in Section 4.5.1 we discuss a heuristic separation algorithm to generate lifted inequalities. The computational results are then discussed in Sections 4.5.2 and 4.5.3 for the monotonic and non-monotonic submodular functions, respectively.

4.5.1 Separation Problem

To derive the violated inequalities of the form (4.16) and (4.19) that separate solution $(\bar{w}, \bar{x}) \in \mathbb{R} \times \mathbb{R}^n_+$, the key step is to identify initial set S and the seed inequality. If $\bar{x}$ is binary, then the separation is trivial by setting $S := \{j \in N : \bar{x}_j = 1\}$. Otherwise, i.e., when some of $\bar{x}$'s components are not binary, we are interested in finding S such that the corresponding inequality of the form (4.16) is most violated. That is, we consider the following optimization problem:

$$\min_{S \in \mathcal{I}} \{f(S) - \sum_{j \in S} \rho_j(S)(1 - \bar{x}_j) + \sum_{j \in \bar{S}} \gamma_0^S(a_j)\bar{x}_j\}, \tag{4.27}$$

where $\mathcal{I}$ is a collection of feasible sets, and γ_0^S is the lifting function of the seed inequality $w \le f(S) - \sum_{j \in \bar{S}} \rho_j(S)x_j$ for conv(P_0).

Algorithm 7 Separation Algorithm for constructing S

$S_0 = \{j \in N : \bar{x}_j \geq 1 - \epsilon_0\}$; $S = S_0$;
$R = \{j \in N : \epsilon_1 < \bar{x}_j < 1 - \epsilon_0\}$; assume $R = \{1, 2, \ldots, |R|\}$;
Sort $(\rho_j(S) - \gamma_0^S(a_j))\bar{x}_j$ for $j \in R$ in monotonic increasing order;
for $j = 1, 2, \ldots |R|$ **do**
    if $S \cup j_0 \in \mathcal{I}$ **then**
        $S = S \cup j_0$;
    end if
end for
Return S;

We first discuss the property of the objective function in (4.27). Given $S_0 \subseteq N$ and $j_0 \in N \backslash S_0$, denote by z_0 and z_1 the objective function values of (4.27) for S_0 and $S_1 := S_0 \cup j_0$, respectively. We have

$$\begin{aligned} z_1 - z_0 &= f(S_1) - f(S_0) + \Delta_1 + \Delta_2 - \rho_{j_0}(S_0)(1 - \bar{x}_{j_0}) - \gamma_0^{S_0}(a_{j_0})\bar{x}_{j_0} \\ &= [\rho_{j_0}(S_0) - \gamma_0^{S_0}(a_{j_0})]\bar{x}_{j_0} + \Delta_1 + \Delta_2, \end{aligned}$$

where $\Delta_1 = \sum_{j \in \bar{S}_1} [\rho_j(S_0) - \rho_j(S_1)](1 - \bar{x}_j)$ and $\Delta_2 = \sum_{j \in S_0} [\gamma_0^{S_1}(a_j) - \gamma_0^{S_0}(a_j)]\bar{x}_j$.

Based on the submodularity definition, we have that $\rho_j(S_0) \geq \rho_j(S_1)$ and $\Delta_1 \geq 0$. We can also verify that $\gamma_0^{S_1}(z) \leq \gamma_0^{S_0}(z)$ for any $z \geq 0$, which implies that $\Delta_2 \leq 0$. Thus, the separation problem becomes rather complex if we take Δ_1 and Δ_2 into account, in particular, with respect to the required computational efforts.

Instead, we drop the terms Δ_1 and Δ_2 and consider the term $[\rho_{j_0}(S_0) - \gamma_0^{S_0}(a_{j_0})]\bar{x}_{j_0}$ only, within a greedy algorithm to construct S. The pseudo-code is outlined in Algorithm 7. We set $\epsilon_0 = 0.5$ and $\epsilon_1 = 10^{-4}$ in the experiments.

Remark 4.2. *If we ignore $\gamma_0^S(a_j)\bar{x}_j$ in (4.27), then the difference between z_1 and z_0 becomes*

$$z_1 - z_0 = \rho_{j_0}(S_0)\bar{x}_{j_0} + \Delta_1.$$

If f is an increasing function, then $\rho_{j_0}(S_0) \geq 0$ and $\Delta_1 \geq 0$. It yields that $z_1 - z_0 \geq 0$ for any j_0, and $\emptyset$ is optimal for the problem (4.27). Hence, it is necessary to consider the term $\gamma_0^S(a_j)\bar{x}_j$ in (4.27) to derive a seed inequality.

4.5.2 Monotonic Submodular Function f

We evaluate the lifted inequalities using the expected utility maximization problem from [2, 122]. Given a set of investment options N, let $v_i \in \mathbb{R}^n_+$ be the value of investments in the future at scenario $i \in [m]$ with probability $\pi_i, i = 1, \ldots, m$. We use the exponential function $1 - \exp(z/\lambda)$ as the utility function with risk tolerance λ. Then the problem is formulated as:

$$\max \left\{ \sum_{i=1}^{m} \pi_i w_i : \; w_i \leq -\exp(-\frac{v_i^T x}{\lambda}) \;\; \forall i \in [m], x \in \mathcal{X} \right\}.$$

We generate the values of v_i using exactly the same settings as in [2]. Specifically, the probability of each scenario is equal to $\pi_i = \frac{1}{m}$ for all $i \in [m]$. The value of investment j at scenario i is

$$v_{ij} = a_j \cdot \exp(\alpha_j + \beta_j \ln f_i + \epsilon_{ij}), \tag{4.28}$$

where a_j is uniformly generated from $[0, 0.2]$. For each $j \in N$, we draw the active return α_j from the uniform distribution $[0.05, 0.1]$. The passive return is represented by $\beta_j \log f_i$, where β_j is uniformly generated from $[0, 1]$ and $\log f_i$ is from Normal$(0.05, 0.0025)$. We generate the residual error ϵ_{ij} from Normal$(0, 0.0025)$.

In our first set of experiments, we consider the cardinality constrained problem with $\mathcal{X} = \{x \in \{0,1\}^n : \sum_{j=1}^{n} x_j \leq d\}$. We set $d = 15$ for all test instances. Note that the lifted inequalities developed in [122] coincide with those in [2] under the cardinality constraint $\mathcal{X}$.

The computational results that compare cutting planes using the approximate lifted inequalities from [2, 122] and our exact lifted inequalities (4.16) and (4.19) are reported in Table 15. Based on those results, we observe a few significant advantages of our exact lifted inequalities (4.16) and (4.19) over the approximated ones. First, cutting planes using our inequalities (4.16) and (4.19) demonstrate a superior computational capacity, given that all instances can be solved exactly with negligible computational times. The approximate lifted inequalities from [2] often need hundred times more computational efforts, and fail to compute a large number of instances within the time limit. Second, utilizing (4.16) and (4.19) makes the branch-and-cut algorithm very scalable. Unlike the case with the lifted inequalities from [2], which is sensitive to the size of instances and parameter λ, the branch-and-cut algorithm with (4.16) and (4.19) has a rather stable performance across all instances, regardless of sizes or parameter values. Finally, the exact lifted inequalities (4.16) and (4.19) provide a very strong polyhedral description to the convex hull of P_0. Although theoretically they are not sufficient to describe the whole convex hull, it is interesting to note that all instances are solved to optimality at the root node without branching. This observation suggests that (4.16) and (4.19) represent a large portion of facet-defining inequalities of $\text{conv}(P_0)$.

Next, in our second set of experiments, we evaluate the lifted inequalities for $\text{conv}(P_{MC})$. We set $r = 5$, and divide N equally likely into 5 subsets. For each $i \in [r]$, we then generate integer d_i uniformly from $[1, 5]$. Table 16 reports the average performance of the approximate lifted inequalities from [2], the lifted inequalities (4.16) and (4.19) for $\text{conv}(P_0)$, and the lifted inequalities (4.21) and (4.26) for $\text{conv}(P_{MC})$. As expected, the overall performance by applying the approximate lifted inequalities from [2] is worst.

By comparing the lifted inequalities (4.16) and (4.19) with respect to (4.21) and (4.26), we observe from Table 16 that the number of added cuts and branch-and-cut nodes are reasonably close for each instance size (i.e., each row in Table 16). It is not surprising due to the fact that the latter group is a direct extension of the former one. Nevertheless, for $m = 100$, we can see a considerable reduction in the number of cuts, and similar improvements in the number of nodes and in the computational time can also be observed.

Another interesting observation from Table 16 is that one instance with $n = 100, m = 100$ and $\lambda = 0.4$ cannot be solved to optimality within the time limit by applying the lifted inequalities (4.16) and (4.19), while all instances in this class can be solved within 21 seconds on average by applying the lifted inequalities (4.21) and (4.26). Hence, we believe that for more difficult instances considering the underlying structure of $\text{conv}(P_{MC})$, rather than the simple set P_0, and then applying the corresponding exact lifted inequalities have a good potential to be more effective.

4.5.3 Non-Monotonic Submodular Function f

As pointed out earlier, unlike [2, 122], our results do not depend on the monotonic property of submodular function f. Hence, it is of interest to carry out a computational study with instances that involve non-monotonic f, and then to evaluate and analyze the performance of our lifted inequalities. Specifically, we consider the power function $-(\frac{z-b}{\lambda})^{\frac{12}{5}}$; see an illustrative example in Figure 10(a). To be consistent and comparable to our previous results, a cardinality constraint is also considered. Hence, we have:

$$\max\left\{\sum_{i=1}^{m}\pi_i w_i:\ w_i \le -(\frac{v_i^T x - b_i}{\lambda})^{\frac{12}{5}}\ \ \forall i\in[m],\ \sum_{j=1}^{n} x_j \le d\right\}.$$

In our experiments, for each $i \in [m]$, we set $\pi_i = \frac{1}{m}$ and $b_i = 2.5$, and then generate v_i according to (4.28). The cardinality budget d is generated from the discrete uniform distribution $[10, 20]$. Since the approximate lifted inequalities from [2, 121] are not applicable, we simply compare our exact lifted inequalities (4.16) and (4.19) with submodular inequalities proposed by Nemhauser and Wolsey [86]. The coefficients of the lifting variables computed by $\gamma_0(z)$ and $\tilde{\gamma}_0(z)$ proposed in [86] are depicted in Figure 10(b). Our numerical results are summarized in Table 17.

Comparing to the results in Table 15, the non-monotonic instances are more challenging than their monotonic counterparts. For the non-monotonic instances, almost half of them cannot be solved within the time limit using the standard submodular formulation based on the submodular inequalities from [86]. For some unsolved instances with small λ, the average gaps at termination can be very large. For example, the average optimality gap at termination over four unsolved instances with $n = 50, m = 50$ and $\lambda = 0.8$ is 55.32%. It is worth pointing out that, a very large number of submodular inequalities are often generated by the branch-and-cut algorithm, despite the fact that our test instances are rather small. A similar observation is also made by Ahmed and Atamtürk [2] on monotonic instances. Hence, it is reasonable to believe that the standard submodular formulation from [86] is often not a practical strategy to solve general submodular maximization problems.

In contrast, after applying our exact lifted inequalities, all instances can be solved to optimality within less than half an hour. Compared to the standard submodular formulation, the numbers of added cuts and explored branch-and-cut nodes are significantly smaller, especially for large values of λ. Hence, these lifted inequalities can be computationally effective, and, in general, strengthening the submodular inequalities is critical for solving the submodular maximization problem.

By further comparing the results for inequalities (4.16) and (4.19) in Tables 15 and 17, we observe that the numbers of cuts and branch-and-cut nodes for the non-monotonic submodular function are much larger than those for the monotonic one. Clearly, it demonstrates that the set of non-monotonic submodular function is much harder to solve, while its continuous relaxation provides, perhaps, a weak approximation. Indeed, we note that it always requires a number of branching operations to find a feasible solution. It indicates that our lifted inequalities probably represent only a small set of facet-defining inequalities. Therefore, developing different types of valid inequalities and more advanced branch-and-cut strategies for the non-monotonic submodular function maximization problems would be two interesting questions to address in future research.

4.6 Concluding Remarks

In this chapter, we study the mixed-integer set of the submodular maximization problem through sequence independent lifting. We strengthen and generalize the previous results from [2] in the following three aspects: (i) our results can be applied to any general concave function g that allows us to build the connections with the mixed 0-1 knapsack set and the single-node flow set; (ii) for P when $\mathcal{X} = \{0,1\}^n$, we prove that the lifting functions on two classes of seed inequalities are naturally subadditive, which immediately leads to two family of facet-defining inequalities; (iii) we further investigate the convex hull of P when $\mathcal{X}$ involves disjoint cardinality constraints and a family of facets is developed by exploiting multidimensional lifting; another family of strong valid inequalities is also developed with some quality guarantees. The computational experiments with monotonic and non-monotonic submodular functions illustrate the effectiveness of the proposed lifted inequalities.

Table 15: The performance of lifted inequalities for P when f is monotonic and $\mathcal{X}$ is a cardinality constraint, i.e., $\mathcal{X} = \{x \in \{0,1\}^n : \sum_{j=1}^{n} x_j \le d\}$. For each class of instances, the number of unsolved instances (out of 10) is presented as the superscript over the "Egap" value.

			Lifted ineqs. from [2]				Ineqs. (4.16) & (4.19)		
n	m	λ	Cuts	Nodes	Time (s)	Egap (%)	Cuts	Nodes	Time (s)
100	100	0.4	1000	0	4		1000	0	2
100	100	0.6	6540	73	42		800	0	2
100	100	0.8	3349	30	21		600	0	1
100	100	1	3030	23	18		600	0	1
100	150	0.4	1500	0	6		1500	0	4
100	150	0.6	8985	67	58		1200	0	3
100	150	0.8	3209	12	19		900	0	2
100	150	1	2910	10	18		900	0	2
200	100	0.4	1000	0	8	5.3^{4}	1000	0	4
200	100	0.6	21342	423	376	1.43^{3}	800	0	3
200	100	0.8	16120	248	221	0.4^{1}	600	0	2
200	100	1	11690	217	208		600	0	2
200	150	0.4	1500	0	11	6.92^{4}	1500	0	6
200	150	0.6	30422	595	852	2.03^{2}	1200	0	5
200	150	0.8	19617	159	251	0.5^{1}	900	0	4
200	150	1	15990	118	194		900	0	4
300	100	0.4	1019	0	12	12.37^{8}	1000	0	6
300	100	0.6	22800	299	439	1.87^{9}	800	0	5
300	100	0.8	24235	795	1053	0.63^{2}	600	0	4
300	100	1	19830	344	483		600	0	4
300	150	0.4	1575	0	19	13.61^{6}	1500	0	10
300	150	0.6	22650	665	1414	2.44^{8}	1200	0	7
300	150	0.8	25250	286	637	1.22^{4}	900	0	5
300	150	1	20960	168	393	0.4^{3}	900	0	5

Table 16: The performance of lifted inequalities for P when f is monotonic and $\mathcal{X}$ involves multiple disjoint cardinality constraints. For each class of instances, the number of unsolved instances (out of 10) is presented as the superscript over the “Egap” value.

			Lifted ineqs. from [2]				Ineqs. (4.16) & (4.19)				Ineqs. (4.21) & (4.26)		
n	m	λ	Cuts	Nodes	Time (s)	Egap (%)	Cuts	Nodes	Time (s)	Egap (%)	Cuts	Nodes	Time (s)
100	50	0.4	12695	495	348		730	2	6		730	2	6
100	50	0.6	3014	62	77		999	8	11		1119	9	13
100	50	0.8	2290	36	56		790	6	9		865	8	10
100	50	1	1760	22	43		690	6	8		750	6	9
100	100	0.4	2225	10	33	8.86^{2}	1491	4	12	2.22^{1}	2299	12	21
100	100	0.6	6078	54	117	3.07^{1}	4080	41	37		2269	11	22
100	100	0.8	7428	82	177		3347	25	35		1997	10	22
100	100	1	4358	36	95		2830	21	28		1639	8	18
200	50	0.4	6450	246	362	7.41^{5}	820	5	13		770	4	13
200	50	0.6	10524	301	563	2.42^{2}	887	7	18		829	6	18
200	50	0.8	6289	155	291		719	5	15		810	6	18
200	50	1	3596	67	160		535	3	11		545	4	11
200	100	0.4	4836	70	221	9.63^{3}	1600	4	23		1226	2	18
200	100	0.6	17500	232	764	4.19^{5}	2630	16	44		2158	12	38
200	100	0.8	13004	151	530	1.15^{2}	2761	17	50		2699	16	51
200	100	1	9656	110	418	0.86^{1}	1730	8	35		1679	8	33
300	50	0.4	583	0	23	11.82^{7}	836	6	19		1285	13	34
300	50	0.6	19750	640	1539	4.47^{9}	776	6	18		820	7	21
300	50	0.8	8836	241	527	1.88^{6}	655	5	17		588	4	16
300	50	1	11507	380	869	1.24^{2}	810	7	21		900	8	25
300	100	0.4	1200	0	47	13.72^{5}	1080	0	20		1240	1	27
300	100	0.6	22500	280	1536	4.33^{9}	1180	2	24		880	0	18
300	100	0.8	11924	144	696	1.98^{7}	1358	5	35		1158	3	30
300	100	1	12329	143	770	1.08^{3}	1463	6	39		923	2	24

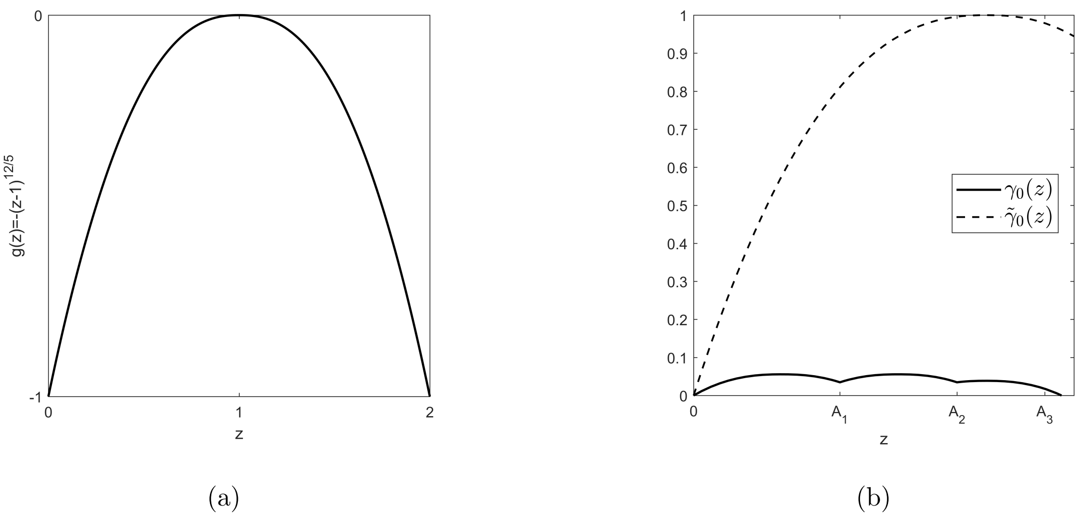


Figure 10: (a) The plot of concave function $-(z-1)^{12/5}$; (b) The plot of coefficients for the lifting variables computed by our proposed $\gamma_0(z)$ and $\tilde{\gamma}_0(z)$, which is derived from submodular inequalities [86].

Table 17: The performance of lifted inequalities for P when f is non-monotonic and $\mathcal{X}$ is a cardinality constraint, i.e., $\mathcal{X} = \{x \in \{0,1\}^n : \sum_{j=1}^{n} x_j \leq d\}$. For each class of instances, the number of unsolved instances (out of 10) is presented as the superscript over the "Egap" value.

			Submodular ineqs. from [86]				Ineqs. (4.16) & (4.19)		
n	m	λ	Cuts	Nodes	Time (s)	Egap (%)	Cuts	Nodes	Time (s)
50	25	0.8	21698	714	147	33.09^{6}	2867	132	22
50	25	1	17444	549	112	17.53^{6}	2935	126	22
50	25	2	12531	373	77	2.99^{6}	2302	89	16
50	25	4	17617	508	114	0.56^{6}	1808	62	12
50	50	0.8	25698	417	158	55.32^{4}	9633	214	69
50	50	1	24831	376	145	25.71^{4}	10046	227	74
50	50	2	29921	454	182	3.68^{4}	7911	173	59
50	50	4	23627	365	143	0.71^{4}	3645	70	25
50	100	0.8	56324	430	338	33.54^{4}	15793	176	118
50	100	1	75853	556	464	15.31^{4}	16002	174	118
50	100	2	65417	479	405	2.39^{4}	14760	160	113
50	100	4	58850	413	349	0.46^{4}	7480	74	53
100	25	0.8	26230	907	325	8.7^{3}	7799	340	99
100	25	1	23687	826	285	4.82^{3}	8018	334	101
100	25	2	33483	1057	438	1.02^{3}	7763	321	103
100	25	4	9502	371	114	0.25^{2}	3079	129	44
100	50	0.8	56978	982	733	7.83^{6}	19255	446	263
100	50	1	65942	1010	878	3.56^{6}	15629	361	216
100	50	2	72781	1129	958	0.87^{6}	15706	353	223
100	50	4	46631	662	583	0.14^{6}	12448	261	185
100	100	0.8	84090	815	1012	15.82^{3}	46021	494	637
100	100	1	96181	862	1162	6.93^{4}	43179	459	594
100	100	2	92225	775	1147	1.48^{3}	41406	428	587
100	100	4	59216	487	724	0.28^{3}	23576	231	313

5.0 Conclusion

In this dissertation, we study two broad classes of mixed-integer optimization problems that is, bilevel optimization and submodular maximization problems. In particular, for bilevel problems our novel results are based on exploiting the concept of the local optimality at the lower level, where we assume that the follower's variables are all binary. Furthermore, we provide several mathematical models with strong valid inequalities for solving these two classes of problems. Some interesting connections between the bilevel optimization and submodular optimization are also explored.

Chapter 2 studies a class of bilevel matroid problem, i.e., bilevel spanning tree problems (BST), with two types of objective functions. The polynomial-time algorithms are developed for all considered BST variants, except the sum-sum BST problem. To solve the sum-sum BST problem we derive an equivalent single-level linear mixed-integer programming formulation and explore its scalability with computational experiments. However, in our computational experiments the standard solver can handle only moderately sized network instances. Thus, for future work it could be of interest to develop more advanced solution approaches, e.g., those based on branch-and-cut ideas that are commonly used for challenging network design problems; see examples in [63].

Next, in Chapter 3 we are motivated by the practical situation that in the bilevel optimization settings, the follower might be not fully rational due to computational limitations. A generalized bilevel framework is then proposed, which naturally provides a hierarchy of upper and lower bounds for the standard bilevel optimization problem. The computational experiments suggest that these bounds are substantially better than those based on the widely used single-level relaxation method. Disjunctive approach and submodularity are exploited to develop two efficient single-level MILP formulations. Therefore, embedding our framework into general purpose branch-and-cut solvers for mixed integer bilevel optimization problems provides a promising direction for future research.

As for the submodular maximization problem, we consider a class of submodular functions that are represented by a concave function composed with a linear function. In Chapter

4, we present the results of our polyhedral study for the submodular maximization problem. Two families of facet-defining inequalities are derived via lifting techniques. The developed polyhedral results complement nicely some classical well-known results on the mixed 0-1 knapsack and single-node flow sets. The computational experiments also show that these lifted facet-defining inequalities are very effective for solving the considered submodular maximization problems to optimality for medium-sized instances within the branch-and-cut framework. Furthermore, for many test instances, we observe that the optimality gaps can be decreased to reasonable small values (e.g., 1%) very quickly using a rather small number of cutting planes and branch nodes. However, closing the gap to terminated optimality gap (e.g., 0.01%) is often more difficult. Also, our proposed linear valid inequalities can be viewed as the approximation to the nonlinear constraint involving a submodular function. It provides an interesting avenue for the future research to explore the approximation approaches that linearize the nonlinear constraints for large-scale submodular maximization instances.

Appendix Proof of Theorem 4.6

Let us first consider the function $\gamma_T(z)$ (defined in equation (4.22)). As shown in Example 4.3, $\gamma_T(z)$ is not subadditive on $z \in \mathbb{R}_+$ in general for $T \neq \emptyset$. However, we can still find the subadditive structure inside $\gamma_T(z)$ in the following discussion. Consider a new function $\tilde{\gamma}_T(z)$ as follows:

$$\tilde{\gamma}_T(z) = \gamma_T(z + a(T)) - \sum_{j \in T} \rho_j(S \setminus j) \quad \forall z \geq 0.$$

Recall that we assume $S = \{1, \ldots, s\}$ such that $a_1 \geq \cdots \geq a_s$. Let $\tilde{A}_k = \sum_{j \in [k] \setminus T} a_j$ for $k \in S$ and $\tilde{A}_0 = 0$. Note that if $k \in T$, then $\tilde{A}_k = \tilde{A}_{k-1}$. Then following Lemma 4.8, we have $\tilde{\gamma}_T(\tilde{A}_k) = \sum_{j \in [k] \setminus T} \rho_j(S \setminus j)$ and

$$\tilde{\gamma}_T(z) = \begin{cases} g(a(S) - \tilde{A}_{k+1} + z) + \tilde{\gamma}_T(\tilde{A}_{k+1}) - f(S) & \text{if } \tilde{A}_k \leq z \leq \tilde{A}_{k+1}, \\ & \quad k = 0, \ldots, s-2, \\ g(z + a(T)) + \tilde{\gamma}_T(\tilde{A}_s) - f(S) & \text{if } z \geq \tilde{A}_{s-1}. \end{cases} \tag{.1}$$

Consequently, based on Proposition 4.4, we can verify that function $\tilde{\gamma}_T(z)$ is subadditive on $z \in \mathbb{R}_+$ for a given $T \subseteq S$.

To establish the proof of Theorem 4.6, our basic idea is to exploit inequalities (4.20) in Lemma 4.7. To show (4.20), we first establish that inequalities similar in spirit to (4.20) hold for $\gamma_T(z)$ in Lemmata A.3 and A.4, which, in turn, rely on the technical results in Lemmata A.1 and A.2. Then we use the fact that $\gamma\binom{z}{\mathbf{u}}$ can be computed through equation (4.23) and $\gamma_T(z)$ to complete the proof of Theorem 4.6.

Lemma A.1. *Function $\tilde{\gamma}_T(z)$ is nonincreasing on T, that is $\tilde{\gamma}_T(z) \leq \tilde{\gamma}_{T'}(z) \leq \gamma_0(z)$ for any $T' \subseteq T \subseteq S$.*

Proof. Note that if $T = \emptyset$, then $\tilde{\gamma}_\emptyset(z) = \gamma_\emptyset(z) = \gamma_0(z)$. Thus, it suffices to show that $\tilde{\gamma}_T(z) \leq \tilde{\gamma}_{T \setminus j_0}(z)$ for any $j_0 \in T$. Let $T' = T \setminus j_0$ and $\tilde{A}'_k = \sum_{j \in [k] \setminus T'} a_j$ for $k \in S$.

If $z \leq \tilde{A}_{j_0 - 1}$, then $\tilde{A}'_k = \tilde{A}_k$ for $k = 0, \ldots, j_0 - 1$. By (.1), it is easy to verify that $\tilde{\gamma}_T(z) = \tilde{\gamma}_{T'}(z)$.

If $z > \tilde{A}_{j_0-1}$, then $\tilde{A}'_k = \tilde{A}_k + a_{j_0}$ for $k = j_0, \ldots, s$. By (.1), it is easy to verify that $\tilde{\gamma}_{T'}(z + a_{j_0}) - \tilde{\gamma}_{T'}(\tilde{A}'_{j_0}) = \tilde{\gamma}_T(z) - \tilde{\gamma}_T(\tilde{A}_{j_0-1})$. Meantime, we can verify that $\tilde{\gamma}_{T'}(z)$ has the form of (4.12) in Section 4.2.3. Then by Lemma 4.2 we have

$$\begin{aligned}\tilde{\gamma}_{T'}(z) - \tilde{\gamma}_{T'}(\tilde{A}'_{j_0-1}) &\geq \tilde{\gamma}_{T'}(z + a_{j_0}) - \tilde{\gamma}_{T'}(\tilde{A}'_{j_0}) \\ &= \tilde{\gamma}_T(z) - \tilde{\gamma}_T(\tilde{A}_{j_0-1}),\end{aligned}$$

which implies that $\tilde{\gamma}_{T'}(z) \geq \tilde{\gamma}_T(z)$ as $\tilde{\gamma}_{T'}(\tilde{A}'_{j_0-1}) = \tilde{\gamma}_T(\tilde{A}_{j_0-1})$.

■

Lemma A.2. *Let $\Delta \in [0, a_j]$ for some $j \in T$. If $0 \leq z \leq \tilde{A}_{j-1}$, then*

$$\gamma_T(a(T)) - \gamma_T(a(T) - \Delta) \leq \gamma_T(z + a(T)) - \gamma_T(z + a(T) - \Delta).$$

Proof. Firstly, by Lemma 4.8, we have

$$\gamma_T(a(T)) - \gamma_T(a(T) - \Delta) = g(a(S)) - g(a(S) - \Delta).$$

We assume that $z \in [\tilde{A}_{k-1}, \tilde{A}_k]$ for some $k \in S \setminus T$, then $k \leq j$ and $a_k \geq a_j$. If $z - \Delta \leq 0$, then $k = 1$. Let $\Omega = a(S) + z \geq a(S)$. By Lemma 4.8, we have

$$\begin{aligned}&[\gamma_T(z + a(T)) - \gamma_T(z + a(T) - \Delta)] - [\gamma_T(a(T)) - \gamma_T(a(T) - \Delta)] \\ &= [g(\Omega - a_1) - g(\Omega - \Delta)] - [g(a(S) - a_1) - g(a(S) - \Delta)] \\ &\geq 0,\end{aligned}$$

where the inequality follows from $a_1 \geq a_j \geq \Delta$ and the concavity of g.

If $z - \Delta \geq 0$, then $\gamma_T(z + a(T)) - \gamma_T(z + a(T) - \Delta) = \tilde{\gamma}_T(z) - \tilde{\gamma}_T(z - \Delta)$ based on the definition of $\tilde{\gamma}_T(z)$. Assume $z - \Delta \in [\tilde{A}_{\ell-1}, \tilde{A}_\ell]$ for some $\ell \leq k$. Since $\Delta \leq a_j \leq a_k$, then either $\ell = k$ or $\ell = k - 1$. If $\ell = k$, then

$$\begin{aligned}\tilde{\gamma}_T(z) - \tilde{\gamma}_T(z - \Delta) &= g(a(S) - \tilde{A}_k + z) - g(a(S) - \tilde{A}_k + z - \Delta) \\ &\geq g(a(S)) - g(a(S) - \Delta),\end{aligned}$$

where the inequality follows from $z \leq \tilde{A}_k$ and the concavity of g. If $\ell = k - 1$, let $\Omega = a(S) - \tilde{A}_{k-1} + z \geq a(S)$. By equation (.1), we have

$$
\begin{aligned}
&[\tilde{\gamma}_T(z) - \tilde{\gamma}_T(z - \Delta)] - [\gamma_T(a(T)) - \gamma_T(a(T) - \Delta)] \\
&= [g(\Omega - a_k) - g(\Omega - \Delta)] - [g(a(S) - a_k) - g(a(S) - \Delta)] \\
&\geq 0,
\end{aligned}
$$

where the inequality follows from $\Delta \leq a_k$ and the concavity of g. ■

Lemma A.3. *For any $T \subseteq S$, we have*

$$\sum_{j \in T} \gamma_{\{j\}}(z_j) \geq \gamma_T(z(T)) \quad \forall z_j \geq 0,$$

where $z(T) = \sum_{j \in T} z_j$.

Proof. We prove the result by induction. If $|T| = 1$, then the statement is trivial.

If the statement holds for $|T| = k - 1$, then we establish that the statement still holds for $|T| = k$. Observe that it is sufficient to show that there exists some $\ell \in T$ such that

$$\gamma_{\{\ell\}}(z_\ell) + \gamma_{T \setminus \ell}(z(T')) \geq \gamma_T(z(T)),$$

where $T' = T \setminus \ell$. There are four possible cases to consider:

Case 1 $\exists\, \ell \in T$ such that $z_\ell \geq a_\ell, z(T') \geq a(T')$:

Based on the assumption, we have $z(T) \geq a(T)$, it follows that

$$
\begin{aligned}
\gamma_{\{\ell\}}(z_\ell) + \gamma_{T'}(z(T')) &= \tilde{\gamma}_{\{\ell\}}(z_\ell - a_\ell) + \tilde{\gamma}_{T'}(z(T') - a(T')) + \sum_{j \in T} \rho_j(S \setminus j) \\
&\geq \tilde{\gamma}_T(z_\ell - a_\ell) + \tilde{\gamma}_T(z(T') - a(T')) + \sum_{j \in T} \rho_j(S \setminus j) \\
&\geq \tilde{\gamma}_T(z_\ell + z(T') - a(T)) + \sum_{j \in T} \rho_j(S \setminus j) \\
&= \gamma_T(z(T)),
\end{aligned}
$$

where the first inequality is based on Lemma A.1, and the second inequality follows from the fact that $\tilde{\gamma}_T(z)$ is subadditive on $z \in \mathbb{R}_+$ for a given T.

Case 2 $\exists\, \ell \in T$ such that $z_\ell \leq a_\ell, z(T') \leq a(T')$:

Based on the assumption, we have $z(T) \leq a(T)$. Let $\Omega = a(S) + z(T') - a(T')$, then $\Omega \leq a(S)$. By Lemma 4.8, it follows that

$$\begin{aligned}
&\gamma_{\{\ell\}}(z_\ell) + \gamma_{T'}(z(T')) - \gamma_T(z(T)) \\
&= g(a(S) - a_\ell + z_\ell) + g(\Omega) - g(a(S)) - g(\Omega - a_\ell + z_\ell) \\
&= [g(a(S) - a_\ell + z_\ell) - g(a(S))] - [g(\Omega - a_\ell + z_\ell) - g(\Omega)] \\
&\geq 0,
\end{aligned}$$

where the inequality follows from the concavity of g and $z_\ell - a_\ell \leq 0$.

Case 3 $\exists\, \ell \in T$ such that $z_\ell \leq a_\ell, z(T') \geq a(T')$:

Based on Lemma 4.8, we have $\gamma_{\{\ell\}}(z_\ell) = g(a(S) - a_\ell + z_\ell) - g(a(S) - a_\ell)$. Without loss of generality, we assume ℓ is the smallest index in T. Then there have two cases to consider:

- if $z(T') \geq a(T') + \tilde{A}_{\ell-1}$, we have

$$\begin{aligned}
&\gamma_{T'}(z_\ell + z(T')) - \gamma_{T'}(z(T')) \\
&= \tilde{\gamma}_{T'}(z(T') - a(T') + z_\ell) - \tilde{\gamma}_{T'}(z(T') - a(T')) \\
&\leq \tilde{\gamma}_{T'}(A_{\ell-1} + z_\ell) - \tilde{\gamma}_{T'}(A_{\ell-1}) = \gamma_{\{\ell\}}(z_\ell),
\end{aligned}$$

  where the inequality follows from Lemma 4.2 that $\tilde{\gamma}$ has the form of (4.11). By Lemma A.1, we have that

$$\gamma_{\{\ell\}}(z_\ell) + \gamma_{T'}(z(T')) \geq \gamma_{T'}(z_\ell + z(T')) \geq \gamma_T(z(T)).$$

- if $z(T') \leq a(T') + \tilde{A}_{\ell-1}$, then by Lemma 4.8 it can be verified that

$$\gamma_T(z(T') + a_\ell) = \gamma_{T'}(z(T')) + \rho_\ell(S \setminus \ell).$$

  Let $\Delta = a_\ell - z_\ell \geq 0$. Note that $\Delta \leq a_\ell$ and $z(T') - a(T') \leq \tilde{A}_{\ell-1}$, it follows from Lemma A.2 that

$$\begin{aligned}
\gamma_T(z(T') + a_\ell) - \gamma_T(z(T') + z_\ell) &\geq \gamma_T(a(T)) - \gamma_T(a(T) - \Delta) \\
&= g(a(S)) - g(a(S) - \Delta)
\end{aligned}$$

$$= \rho_\ell(S \setminus \ell) - \gamma_{\{\ell\}}(z_\ell).$$

Replacing $\gamma_T(z(T') + a_\ell)$ with $\gamma_{T'}(z(T')) + \rho_\ell(S \setminus \ell)$ in the above inequality, it yields the desired results.

Case 4 $\exists\, \ell \in T$ such that $z_\ell \geq a_\ell, z(T') \leq a(T')$:

Since $z(T') \leq a(T')$, then there exists $\ell' \in T$ such that $z_{\ell'} \leq a_{\ell'}$. We can reduce this case to either Case 2 or Case 3.

In summary, the claim holds for $|T| = k$ and we complete the proof. ∎

Lemma A.4. *For any $j \in S$ and $z_0, z_1 \geq 0$, we have*

$$\gamma_0(z_0) + \gamma_{\{j\}}(z_1) \geq \gamma_{\{j\}}(z_0 + z_1).$$

Proof. If $z_1 \geq A_{j-1}$, then $\gamma_{\{j\}}(z) = \gamma_0(z)$ and $\gamma_{\{j\}}(z_0 + z_1) = \gamma_0(z_0 + z_1)$ based on Lemma 4.8. It yields that the claim holds due to the subadditivity of γ_0 in this case.

If $a_j \leq z_1 \leq A_{j-1}$, then $\gamma_{\{j\}}(z_1) = \tilde{\gamma}(z_1 - a_j) + \rho_j(S \setminus j)$. Note that $\gamma_0(z) = \tilde{\gamma}_{\{j\}}(z)$ when $z \leq A_{j-1}$ through (.1), thus

$$\begin{aligned}
\gamma_0(z_0) + \gamma_{\{j\}}(z_1) &= \gamma_0(z_0) + \gamma_0(z_1 - a_j) + \rho_j(S \setminus j) \\
&\geq \gamma_0(z_0 + z_1 - a_j) + \rho_j(S \setminus j) \\
&\geq \tilde{\gamma}_{\{j\}}(z_0 + z_1 - a_j) + \rho_j(S \setminus j) = \gamma_{\{j\}}(z_0 + z_1),
\end{aligned}$$

where the second inequality follows from Lemma A.1 that $\tilde{\gamma}_{\{j\}}(z) \leq \gamma_0(z)$.

If $z_1 \leq a_j$ and $z_0 + z_1 \leq a_j$, then by Lemma 4.8, we have

$$\begin{aligned}
\gamma_{\{j\}}(z_0 + z_1) - \gamma_{\{j\}}(z_1) &= g(a(S) - a_j + z_0 + z_1) - g(a(S) - a_j + z_1) \\
&\leq g(a(S) - a_1 + z_0 + z_1) - g(a(S) - a_1 + z_1) \\
&= \gamma_0(z_0 + z_1) - \gamma_0(z_1) \leq \gamma_0(z_0),
\end{aligned}$$

where the first inequality follows from $a_j \leq a_1$, and the second inequality follows from the subadditivity of γ_0.

If $z_1 \leq a_j$, $z_0 + z_1 \geq a_j$ and $z_0 \leq A_j$, then let $\Delta = a_j - z_1 \leq z_0$. By Lemmata 4.1 and 4.8, we have that

$$\gamma_0(z_0) - \gamma_0(z_0 - \Delta) \geq \gamma_0(A_j) - \gamma_0(A_j - \Delta) = \gamma_{\{j\}}(a_j) - \gamma_{\{j\}}(z_1).$$

Let $z_1' = a_j$ and $z_0' = z_0 - \Delta$, then $z_0' + z_1' = z_0 + z_1$. Since $a_j \leq z_1' \leq A_j$, based on the aforementioned case, we have

$$\gamma_0(z_0 - \Delta) + \gamma_{\{j\}}(z_1') \geq \gamma_{\{j\}}(z_0 + z_1).$$

Summing the above two inequalities, we get the desired result.

If $z_1 \leq a_j$ and $z_0 \geq A_{j-1}$, then by Lemma 4.8, we have $\gamma_{\{j\}}(z_1) = \gamma_0(A_{j-1} + z_1) - \gamma_0(A_{j-1})$ and $\gamma_{\{j\}}(z_0 + z_1) = \gamma_0(z_0 + z_1)$. Then the result directly follows from Lemma 4.2.

■

We now come to prove Theorem 4.6.

Theorem 4.6. It suffices to show that the lifting function $\gamma\binom{z}{\mathbf{u}}$ satisfies the SI condition (4.20). Let $\Gamma \subseteq \bar{S}$ and $z_j \geq 0$ for $j \in \Gamma$. Then for any $T \subseteq S$ such that $|T_i| = \max\{0, |S_i| + |\Gamma_i| - d_i\}$ for all $i \in [r]$, we construct $\{T^j\}_{j \in \Gamma}$ as follows: for each $i \in [r]$, suppose $\Gamma_i = \{1, \ldots, |\Gamma_i|\}$ and $T_i = \{i_1, \ldots, i_{|T_i|}\}$,

- if $|S_i| < d_i$, then $|T_i| < |\Gamma_i|$. Let $T^j = \{i_j\}$ for $j = 1, \ldots, |T_i|$, and $T^j = \emptyset$ for $j = |T_i| + 1, \ldots, |\Gamma_i|$.
- if $|S_i| = d_i$, then $|T_i| = |\Gamma_i|$. Let $T^j = \{i_j\}$ for $j \in \Gamma_i$.

Therefore, by Lemmata A.3 and A.4, we have

$$\sum_{j \in \Gamma} \gamma_{T^j}(z_j) \geq \gamma_T(z(\Gamma)).$$

Recall equation (4.23), we have

$$\begin{aligned}
\gamma\begin{pmatrix} z(\Gamma) \\ \sum_{j \in \Gamma} \mathbf{e}_{\sigma(j)} \end{pmatrix} &= \max_{T \subseteq S} \{\gamma_T(z(\Gamma)) : |T_i| = \max\{0, |S_i| + |\Gamma_i| - d_i\} \ \forall i \in [r]\} \\
&\leq \max_{\{T^j\}_{j \in \Gamma}} \{\sum_{j \in \Gamma} \gamma_{T^j}(z_j) : |T^j| = \max\{0, |S_i| + 1 - d_i\}, T^j \subseteq S_i, i = \sigma(j)\}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j \in \Gamma} \max_{T^j} \{\gamma_{T^j}(z_j) : |T^j| = \max\{0, |S_i| + 1 - d_i\}, T^j \subseteq S_i, i = \sigma(j)\} \\
&= \sum_{j \in \Gamma} \gamma \begin{pmatrix} z_j \\ \mathbf{e}_{\sigma(j)} \end{pmatrix}.
\end{aligned}
$$

■

Bibliography

[1] Emile Aarts, Emile HL Aarts, and Jan Karel Lenstra. *Local Search in Combinatorial Optimization*. Princeton University Press, Princeton, NJ, USA, 2003.

[2] Shabbir Ahmed and Alper Atamtürk. Maximizing a class of submodular utility functions. *Mathematical Programming*, 128(1-2):149–169, 2011.

[3] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1993.

[4] Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin, and M.R. Reddy. Applications of network optimization. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Handbooks in Operations Research and Management Science, Vol. 7*, volume 7, pages 1–83. Elsevier, Amsterdam, 1995.

[5] Alper Atamtürk and Andrés Gómez. Maximizing a class of utility functions over the vertices of a polytope. *Operations Research*, 65(2):433–445, 2017.

[6] Alper Atamtürk, Simge Küçükyavuz, and Birce Tezel. Path cover and path pack inequalities for the capacitated fixed-charge network flow problem. *SIAM Journal on Optimization*, 27(3):1943–1976, 2017.

[7] Alper Atamtürk and Vishnu Narayanan. Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618–622, 2008.

[8] Alper Atamtürk and Vishnu Narayanan. The submodular knapsack polytope. *Discrete Optimization*, 6(4):333–344, 2009.

[9] Alper Atamtürk, George L. Nemhauser, and Martin W.P. Savelsbergh. The mixed vertex packing problem. *Mathematical Programming*, 89(1):35–53, 2000.

[10] Charles Audet, Pierre Hansen, Brigitte Jaumard, and Gilles Savard. Links between linear bilevel and mixed 0–1 programming problems. *Journal of Optimization Theory and Applications*, 93(2):273–300, 1997.

[11] Cristina Bazgan, Sonia Toubaline, and Zsolt Tuza. The most vital nodes with respect to independent set and vertex cover. *Discrete Applied Mathematics*, 159(17):1933–1946, 2011.

[12] Behdad Beheshti, Osman Y. Özaltın, M. Hosein Zare, and Oleg A. Prokopyev. Exact solution approach for a class of nonlinear bilevel knapsack problems. *Journal of Global Optimization*, 61(2):291–310, 2015.

[13] Behdad Beheshti, Oleg A. Prokopyev, and Eduardo L. Pasiliao. Exact solution approaches for bilevel assignment problems. *Computational Optimization and Applications*, 64(1):215–242, 2016.

[14] Omar Ben-Ayed and Charles E. Blair. Computational difficulties of bilevel linear programming. *Operations Research*, 38(3):556–560, 1990.

[15] Francis T. Boesch, Appajosyula Satyanarayana, and Charles L. Suffel. A survey of some network reliability analysis and synthesis results. *Networks*, 54(2):99–107, 2009.

[16] Juan S. Borrero, Oleg A. Prokopyev, and Denis Sauré. Sequential interdiction with incomplete information and learning. *Operations Research*, 67(1):72–89, 2019.

[17] Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005.

[18] Luce Brotcorne, Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model for toll optimization on a multicommodity transportation network. *Transportation Science*, 35(4):345–358, 2001.

[19] Niv Buchbinder and Moran Feldman. Submodular functions maximization problems. In *Handbook of Approximation Algorithms and Metaheuristics, Second Edition*, pages 771–806. CRC Press, 2018.

[20] Christoph Buchheim, Dorothee Henke, and Felix Hommelsheim. On the complexity of the bilevel minimum spanning tree problem. *arXiv preprint arXiv:2012.12770*, 2020.

[21] Alberto Caprara, Margarida Carvalho, Andrea Lodi, and Gerhard J. Woeginger. A complexity and approximability study of the bilevel knapsack problem. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 98–109. Springer, 2013.

[22] Alberto Caprara, Margarida Carvalho, Andrea Lodi, and Gerhard J. Woeginger. A study on the computational complexity of the bilevel knapsack problem. *SIAM Journal on Optimization*, 24(2):823–838, 2014.

[23] Alberto Caprara, Margarida Carvalho, Andrea Lodi, and Gerhard J. Woeginger. Bilevel knapsack with interdiction constraints. *INFORMS Journal on Computing*, 28(2):319–333, 2016.

[24] Jean Cardinal, Erik D. Demaine, Samuel Fiorini, Gwenaël Joret, Stefan Langerman, Ilan Newman, and Oren Weimann. The Stackelberg minimum spanning tree game. *Algorithmica*, 59(2):129–144, 2011.

[25] Barun Chandra, Howard Karloff, and Craig Tovey. New results on the old k-opt algorithm for the traveling salesman problem. *SIAM Journal on Computing*, 28(6):1998–2029, 1999.

[26] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[27] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, 2007.

[28] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*. Springer, Berlin, 2014.

[29] Federico Della Croce and Rosario Scatamacchia. An exact approach for the bilevel knapsack problem with interdiction constraints and extensions. *Mathematical Programming*, 2020. To appear.

[30] Stephan Dempe. *Foundations of Bilevel Programming*. Kluwer Academic Publishers, Dordrecht, 2002.

[31] Stephan Dempe, Boris S. Mordukhovich, and Alain Bertrand Zemkoho. Sensitivity analysis for two-level value functions with applications to bilevel programming. *SIAM Journal on Optimization*, 22(4):1309–1343, 2012.

[32] Scott DeNegre. *Interdiction and discrete bilevel linear programming*. PhD thesis, Lehigh University, 2011.

[33] Brian W Dolhansky and Jeff A Bilmes. Deep submodular functions: Definitions and learning. In *Advances in Neural Information Processing Systems*, pages 3404–3412, 2016.

[34] Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298. ACM, 2009.

[35] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.

[36] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, 172(1-2):77–103, 2018.

[37] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. Interdiction games and monotonicity, with application to knapsack problems. *INFORMS Journal on Computing*, 31(2):390–410, 2019.

[38] Greg N. Frederickson and Roberto Solis-Oba. Increasing the weight of minimum spanning trees. *Journal of Algorithms*, 33(2):244–266, 1999.

[39] Satoru Fujishige. *Submodular functions and optimization*. Elsevier, Netherlands, 2005.

[40] Ziyou Gao, Jianjun Wu, and Huijun Sun. Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, 39(6):479–495, 2005.

[41] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 2002.

[42] Elisabeth Gassner. The computational complexity of continuous-discrete bilevel network problems. Technical report, 2009.

[43] Elisabeth Gassner and Bettina Klinz. The computational complexity of bilevel assignment problems. *4OR*, 7(4):379–394, 2009.

[44] Bryan Gilbert and Wendy Myrvold. Maximizing spanning trees in almost complete graphs. *Networks*, 30(2):97–104, 1997.

[45] Zonghao Gu, George L Nemhauser, and Martin WP Savelsbergh. Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4(1):109–129, 2000.

[46] Oktay Günlük and Yves Pochet. Mixing mixed-integer inequalities. *Mathematical Programming*, 90(3):429–457, 2001.

[47] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2019.

[48] Gregory Gutin and Abraham P Punnen. *The Traveling Salesman Problem and Its Variations*, volume 12. Kluwer Academic Publishers, New York, 2004.

[49] Richard W Hamming. *Coding and information theory*. Prentice-Hall, Inc., 1986.

[50] Reiner Horst, Panos M. Pardalos, and Nguyen Van Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht, 2000.

[51] IBM. ILOG CPLEX Optimization Studio (12.8), 2017. Accessed April 9, 2020.

[52] Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

[53] Toshinari Itoko and Satoru Iwata. Computational geometric approach to submodular function minimization for multiclass queueing systems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 267–279. Springer, 2007.

[54] Robert G. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32(2):146–164, 1985.

[55] A. N. C. Kang, R. C. T. Lee, Chin-Liang Chang, and Shi-Kuo Chang. Storage reduction through minimal spanning trees and spanning forests. *IEEE Transactions on Computers*, C-26(5):425–434, 1977.

[56] Kengo Katayama, Akihiro Hamamoto, and Hiroyuki Narihisa. An effective local search for the maximum clique problem. *Information Processing Letters*, 95(5):503–511, 2005.

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

[58] Hervé Kerivin and A. Ridha Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

[59] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2):291–307, 1970.

[60] Fatma Kılınç-Karzan, Simge Küçükyavuz, and Dabeen Lee. Joint chance-constrained programs and the intersection of mixing sets through a submodularity lens. *arXiv preprint arXiv:1910.01353*, 2019. Accessed 1 April 2020.

[61] Tamás Kis and András Kovács. On bilevel machine scheduling problems. *OR Spectrum*, 34(1):43–68, 2012.

[62] TD Klastorin. On a discrete nonlinear and nonseparable knapsack problem. *Operations Research Letters*, 9(4):233–237, 1990.

[63] Thorsten Koch and Alexander Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32(3):207–232, 1998.

[64] Thorsten Koch, Alexander Martin, and Stefan Voß. Steinlib: An updated library on steiner tree problems in graphs. In Xiu Zhen Cheng and Ding-Zhu Du, editors, *Steiner Trees in Industry*, pages 285–325. Springer, Boston, MA, 2001.

[65] Andreas Krause and Daniel Golovin. Submodular function maximization. In Lucas Bordeaux, Youssef Hamadi, and PushmeetEditors Kohli, editors, *Tractability: Practical Approaches to Hard Problems*, page 71–104. Cambridge University Press, Cambridge, 2014.

[66] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654, 2007.

[67] Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec):2761–2801, 2008.

[68] Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, 44(12-part-1):1608–1622, 1998.

[69] Eugene L Lawler. *Combinatorial Optimization: Networks and Matroids*. Courier Corporation, 2001.

[70] Jon Lee and Jennifer Ryan. Matroid applications and algorithms. *ORSA Journal on Computing*, 4(1):70–98, 1992.

[71] Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

[72] Jian Li and Amol Deshpande. Maximizing expected utility for stochastic combinatorial optimization problems. *Mathematics of Operations Research*, 44(1):354–375, 2019.

[73] Ning Li, Jennifer C. Hou, and Lui Sha. Design and analysis of an MST-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, 2005.

[74] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

[75] Andrea Lodi, Ted K. Ralphs, and Gerhard J. Woeginger. Bilevel programming and the separation problem. *Mathematical Programming*, 146(1-2):437–458, 2014.

[76] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[77] Leonardo Lozano and J Cole Smith. A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research*, 65(3):768–786, 2017.

[78] James Luedtke, Shabbir Ahmed, and George L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, 2010.

[79] Thomas L. Magnanti and Laurence A. Wolsey. Optimal trees. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Handbooks in Operations Research and Management Science, Vol. 7*, volume 7, pages 503–615. Elsevier, Amsterdam, 1995.

[80] Thomas L. Magnanti and Laurence A. Wolsey. Optimal trees. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Models*, volume 7

of *Handbooks in Operations Research and Management Science*, pages 503 – 615. Elsevier, 1995.

[81] Thomas L. Magnanti and Richard T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[82] Rajiv T Maheswaran, Jonathan P Pearce, and Milind Tambe. Distributed algorithms for dcop: A graphical-game-based approach. In *ISCA PDCS*, pages 432–439, 2004.

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

[84] Silvano Martello, David Pisinger, and Paolo Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.

[85] James T. Moore and Jonathan F. Bard. The mixed integer linear bilevel programming problem. *Operations Research*, 38(5):911–921, 1990.

[86] George L Nemhauser and Laurence A Wolsey. Integer programming and combinatorial optimization. *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, 20:8–12, 1988.

[87] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[88] Mark Newman. *Networks*. Oxford University Press, New York, 2018.

[89] James B Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

[90] James Oxley. What is a matroid. *Cubo*, 5:179–218, 2003.

[91] Harilaos N Psaraftis. k-interchange procedures for local search in a precedence-constrained routing problem. *European Journal of Operational Research*, 13(4):391–402, 1983.

[92] Paat Rusmevichientong, Zuo-Jun Max Shen, and David B Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680, 2010.

[93] Shinsaku Sakaue and Masakazu Ishihata. Accelerated best-first search with upper-bound computation for submodular function maximization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[94] Martin WP Savelsbergh. Local search in routing problems with time windows. *Annals of Operations research*, 4(1):285–305, 1985.

[95] Alejandro A Schäffer. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.

[96] Paul JH Schoemaker. The expected utility model: Its variants, purposes, evidence and limitations. *Journal of Economic Literature*, pages 529–563, 1982.

[97] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[98] Xueyu Shi, Oleg Prokopyev, and Bo Zeng. Sequence independent lifting for the set of submodular maximization problem. In *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 2020.

[99] Xueyu Shi, Bo Zeng, and Oleg A. Prokopyev. On bilevel minimum and bottleneck spanning tree problems. *Networks*, 74(3):251–273, 2019.

[100] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

[101] J. Cole Smith, Churlzu Lim, and Fransisca Sudargho. Survivable network design under optimal and heuristic interdiction scenarios. *Journal of Global Optimization*, 38(2):181–199, 2007.

[102] Peter Stobbe and Andreas Krause. Efficient minimization of decomposable submodular functions. In *Advances in Neural Information Processing Systems*, pages 2208–2216, 2010.

[103] Sahar Tahernejad and Ted K. Ralphs. Valid inequalities for mixed integer bilevel optimization problems. *COR@L Laboratory Report 20T-013*, 2020. Accessed June 7, 2021.

[104] Sahar Tahernejad, Ted K. Ralphs, and Scott T. DeNegre. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Mathematical Programming Computation*, 2019. To appear.

[105] Yen Tang, Jean-Philippe P. Richard, and J. Cole Smith. A class of algorithms for mixed-integer bilevel min–max optimization. *Journal of Global Optimization*, 66(2):225–262, 2016.

[106] Sebastian Tschiatschek, Rishabh K Iyer, Haochen Wei, and Jeff A Bilmes. Learning mixtures of submodular functions for image collection summarization. In *Advances in neural information processing systems*, pages 1413–1421, 2014.

[107] Oleksii Ursulenko. *Exact methods in fractional combinatorial optimization*. PhD thesis, Texas A & M University, 2011.

[108] Luis Vicente, Gilles Savard, and J. Judice. Discrete linear bilevel programming problem. *Journal of Optimization Theory and Applications*, 89(3):597–614, 1996.

[109] Lizhi Wang and Pan Xu. The watermelon algorithm for the bilevel integer linear programming problem. *SIAM Journal on Optimization*, 27(3):1403–1430, 2017.

[110] Xiaochun Wang, Xiali Wang, and D. Mitchell Wilkes. A divide-and-conquer approach for minimum spanning tree-based clustering. *IEEE Transactions on Knowledge and Data Engineering*, 21(7):945–958, 2009.

[111] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963, 2015.

[112] Ningji Wei, Jose L. Walteros, and Foad Mahdavi Pajouh. Integer programming formulations for minimum spanning tree interdiction. *Optimization online*, 2019. Acessed April 9, 2020.

[113] Neil White and Neil M White. *Matroid Applications*. Cambridge University Press, 1992.

[114] Laurence A Wolsey. Valid inequalities and superadditivity for 0–1 integer programs. *Mathematics of Operations Research*, 2(1):66–77, 1977.

[115] Laurence A Wolsey. Submodularity and valid inequalities in capacitated fixed charge networks. *Operations Research Letters*, 8(3):119–124, 1989.

[116] Bang Ye Wu and Kun-Mao Chao. *Spanning trees and optimization problems*. CRC Press, Boca Raton, 2004.

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

[118] Hao-Hsiang Wu and Simge Küçükyavuz. An exact method for constrained maximization of the conditional value-at-risk of a class of stochastic submodular functions. *arXiv preprint arXiv:1903.08318*, 2019. Accessed 1 April 2020.

[119] Huile Xu, Yi Zhang, Christos G. Cassandras, Li Li, and Shuo Feng. A bi-level cooperative driving strategy allowing lane changes. *arXiv preprint arXiv:1912.11495*, 2019. Accessed April 9, 2020.

[120] Ying Xu, Victor Olman, and Dong Xu. Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning trees. *Bioinformatics*, 18(4):536–545, 2002.

[121] Jiajin Yu and Shabbir Ahmed. Maximizing expected utility over a knapsack constraint. *Operations Research Letters*, 44(2):180–185, 2016.

[122] Jiajin Yu and Shabbir Ahmed. Maximizing a class of submodular utility functions with constraints. *Mathematical Programming*, 162(1-2):145–164, 2017.

[123] Jiajin Yu and Shabbir Ahmed. Polyhedral results for a class of cardinality constrained submodular minimization problems. *Discrete Optimization*, 24:87–102, 2017.

[124] Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, pages 2483–2491, 2011.

[125] Charles T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 100(1):68–86, 1971.

[126] M. Hosein Zare, Juan S. Borrero, Bo Zeng, and Oleg A. Prokopyev. A note on linearized reformulations for a class of bilevel linear integer problems. *Annals of Operations Research*, 272(1-2):99–117, 2019.

[127] M. Hosein Zare, Oleg A. Prokopyev, and Denis Sauré. On bilevel optimization with inexact follower. *Decision Analysis*, 17(1):74–95, 2020.

[128] Bo Zeng and Yu An. Solving bilevel mixed integer program by reformulations and decomposition. *Optimization online*, 2014. Accessed April 9, 2020.

[129] Bo Zeng and Jean-Philippe P Richard. A polyhedral study on 0–1 knapsack problems with disjoint cardinality constraints: strong valid inequalities by sequence-independent lifting. *Discrete Optimization*, 8(2):259–276, 2011.

[130] Ming Zhao, Kai Huang, and Bo Zeng. A polyhedral study on chance constrained program with random right-hand side. *Mathematical Programming*, 166(1-2):19–64, 2017.