On Bilevel Optimization with Inexact Follower

M. Hosein Zare, Oleg A. Prokopyev

Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15260, {moz3,droleg}@pitt.edu

Denis Sauré

Department of Industrial Engineering, University of Chile, dsaure@dii.uchile.cl

Traditionally, in the bilevel optimization framework a leader chooses her actions by solving an upper-level problem, assuming that a follower chooses an *optimal* reaction by solving a lower-level problem. However, in many settings the lower-level problems might be non-trivial, thus requiring the use of tailored algorithms for their solution. More importantly, in practice such problems might be solved inexactly by heuristics and approximation algorithms. Motivated by this consideration, we study a broad class of bilevel optimization problems where the follower might not react optimally to the leader's actions. In particular, we present a modeling framework in which the leader considers that the follower might use one of a number of known algorithms to solve the lower-level problem either approximately or heuristically. Thus, the leader is able to hedge against the follower's use of suboptimal solutions. We provide algorithmic implementations of the framework for a class of nonlinear bilevel knapsack problem (BKP), and we illustrate the potential impact of incorporating this realistic feature through numerical experiments in the context of defender-attacker problems.

Key words: Bilevel Optimization, Hierarchical Optimization, Robust Optimization, Heuristics, Defender-attacker problem

1. Introduction

Motivation. In bilevel optimization, a *leader* solves an *upper-level* formulation whose objective function depends on a set of *lower-level* decisions, which in turn are made by a *follower* in reaction to the leader's (upper-level) decisions (Bard 1998, Colson et al. 2007).[1] In this framework, the upper-level decisions might affect the lower-level objective function and/or feasible region, and are considered as an input by the follower, who is traditionally assumed to react optimally to them. The outlined setting suggests that, while the leader is able to act strategically by anticipating the follower's (optimal) reaction, the follower's role is rather reactionary. For this reason, bilevel optimization is typically used to, for example, model Stackelberg games (Fudenberg and Tirole 1991), and has been also applied in many areas to model the interaction between rational agents that make decisions sequentially. These areas include transportation (Gzara 2013), energy (Bard et al. 2000), interdiction (Wood 1993, Shen et al. 2012, Zhang et al. 2018), revenue management (Côté and Savard 2003), cybersecurity (Arroyo and Galiana 2005), computational

[1] In the remainder of the paper we use "her" and "his" whenever we refer to the leader and the follower, respectively.

biology (Burgard et al. 2003, Ren et al. 2013) and defense (Brown et al. 2006). For more details on bilevel optimization we refer the reader to Colson et al. (2007) and the references therein.

A traditional assumption in bilevel optimization is that both the leader and follower have the computational means to solve the upper- and lower-level formulations optimally, respectively. However, in many important application areas, one has that either: (*i*) there is no known efficient method to solve the lower-level formulation to optimality (for a given set of upper-level decisions); or (*ii*) the follower either is not sufficiently sophisticated or does not have the computational resources necessary to find an optimal solution to the lower-level problem in a timely manner. In both cases, the follower typically resorts to using computationally-tractable heuristic/approximate algorithms.

In particular, for a given leader's decision the lower-level optimization problem may correspond to a medium- or large-sized instance of an NP-hard problem: while exact algorithms may be available, they often take prohibitively large computing time to produce optimal solutions. A concrete and, perhaps, the simplest example of such a lower-level formulation is the linear *0–1 knapsack problem*, which is NP-hard (Garey and Johnson 1979). In the bilevel optimization literature, problems that involve knapsack-like constraints at the lower level are known as bilevel knapsack problems (see Beheshti et al. (2015), Brotcorne et al. (2009), Caprara et al. (2013), Özaltın et al. (2010)). Exact algorithms for single- and multi-dimensional 0–1 knapsack problems aim at intelligent enumeration of feasible solutions, e.g., based on branch-and-bound schemes (see Horowitz and Sahni (1974), Martello and Toth (1977)), or dynamic programming (see Horowitz and Sahni (1974), Toth (1980)). While a number of such algorithms are specially tailored to solve large-scale instances (see, e.g., Balas and Zemel (1980), Fayard and Plateau (1982), Kellerer et al. (2004), Martello and Toth (1988)), in the worst case it takes them an exponential running time to find an optimal solution.

In general, when faced with such "hard" optimization models, a common approach in many practical settings is to resort to either heuristic or approximation algorithms that find fairly good solutions in reasonable time. For example, many such algorithms have been proposed to find approximate solutions for the linear 0–1 knapsack problem, including a rather simple greedy-based 1/2-approximation algorithm (see, e.g., Kellerer et al. (2004)) and more complex fully polynomial-time approximation schemes (see Ibarra and Kim (1975)). We refer the reader to Martello and Toth (1990) and Kellerer et al. (2004) for detailed overviews of exact and approximation algorithms for different types of knapsack problems.

Objectives and assumptions. In this paper, we depart from the assumption of a resourceful follower and study bilevel optimization in settings where the follower might use one of many algorithms to react to the upper-level decisions. In particular, we assume that the leader does not know upfront the algorithm to be used by the follower, but knows that it belongs to a known finite set of algorithms, which is denoted by $\mathcal{H}$. We refer to the uncertainty about the follower's choice of an algorithm from $\mathcal{H}$ as the *lower-level algorithmic uncertainty* or simply, the *lower-level uncertainty.*

The related case of uncertainty in the parameters of bilevel problems can be viewed as that of asymmetric information between two parties. Using robust optimization techniques, Nikoofal and Zhuang (2012) study defender-attacker problems with a strategic attacker whose valuation of targets is unknown but with a known range. In the same context, Nikoofal and Zhuang (2015) explore the impact of information asymmetry and shows that the attacker's private information may significantly affect defender's first-mover advantage. Unlike extant literature, where stochastic and robust optimization methods are used to handle uncertainty issues related to the problem parameters and input data, our framework deals with uncertainty on the solution method used by the lower-level decision-maker, i.e., the follower.

In the first model, we assume that the leader takes a conservative approach towards lower-level uncertainty. That is, the leader assumes that, given an upper-level decision, the follower uses the algorithm from $\mathcal{H}$ that produces the most damaging (to the leader) lower-level decision.

In the second approach we follow a less conservative method and assume that the follower instead selects the algorithm that produces a lower-level decision that is the Γ-th least damaging to the leader, where $\Gamma \in \{1, \ldots, |\mathcal{H}|\}$ is a parameter pre-defined by the decision-maker, i.e., the leader. Clearly, if $\Gamma = |\mathcal{H}|$, then both models coincide. By changing the value of Γ, the leader can control her level of conservatism, which is somewhat similar to the classical robust optimization approach of Bertsimas and Sim (2004, 2003) for dealing with data uncertainty in mathematical programming.

In our third approach, we assume that the leader has prior information about the likelihood that the follower would use one of the algorithms, which is represented by a probability distribution over the set of possible algorithms. Using this information, the leader minimizes the expected value of her objective function, where the expectation is taken with respect to the follower's choice of an algorithm.

In developing our results we make several assumptions that are rather common in the bilevel optimization literature. In particular, we assume that the upper-level decisions are irrevocable and fully observed by the follower before selecting the lower-level decisions, and that the lower-level problem is well defined (i.e., it has a non-empty and bounded feasible region) for any possible set of upper-level decisions. Furthermore, we assume that the leader knows the set of algorithms, $\mathcal{H}$, that might be used by the follower. Recall that in the standard bilevel optimization framework, the leader has full information about the follower's optimization problem. Thus, it is reasonable to assume that the leader should be able to construct a set of algorithms $\mathcal{H}$ that contains one of the solution methods used by the follower. In this regard, traditional bilevel optimization can be seen through the lens of our framework when the follower always uses an exact algorithm. That is, the set

$\mathcal{H}$ consists of this algorithm and thus, $|\mathcal{H}| = 1$. Moreover, the optimistic and pessimistic models as well as the strong-weak approach (Cao and Leung (2002), Zare et al. (2018)) of the standard bilevel optimization can also be viewed as special cases of our framework, see the discussion in Section 2.2.

Main contributions. Our contribution can be summarized as follows. First, we propose a framework to quantifying the impact of relaxing the assumption that the follower reacts optimally to the leader's decisions. This ought to fit settings where the follower either does not have the computational resources or is not sufficiently sophisticated to implement an exact approach to solve the lower-level problem. This is arguably the case in many application areas, thus our work ought to contribute to closing the gap between the theory and practice in the bilevel optimization area.

In this regard, we propose an approach to deal with lower-level uncertainty. We propose three different models to handle lower-level uncertainty that differ in their degree of conservatism and use of prior information on the likelihood of the use of any given algorithm by the follower. Furthermore, we propose different metrics to evaluate the (leader's) loss in the upper-level objective function due to the lower-level uncertainty, that can be used to compare different approaches. In particular, we present a series of results that interconnect the different approaches towards uncertainty and the different metrics alluded above. These results allow quantifying and/or bounding upfront the leader's loss due to the lower-level uncertainty, and thus might be used in practice for selecting an appropriate approach to handling said uncertainty.

The bilevel optimization literature considering "*inexact*" followers is scant. Smith et al. (2007) study a network interdiction setting where the follower uses one specific heuristic method, i.e., in the context of our work, $|\mathcal{H}| = 1$. Furthermore, the modeling approach by Shan and Zhuang (2013), while designed specifically for the case of defender-attacker settings, is similar to ours in the sense that the one of the players (a centralized government) does not know the other player (terrorist) type, i.e., they may be either strategic or non-strategic, and choose an unanticipated solution. In particular, the model presented in Shan and Zhuang (2013) can be seen as related to a special case of the third modeling approach proposed in this paper, where $|\mathcal{H}| = 2$. Note that Shan and Zhuang (2013) assume that the probability of the other attacker being strategic or non-strategic is available to the defender. In this regard, the first and second modeling approaches in this paper still can be applied when such information is missing.

For our second contribution, we provide a prescriptive approach to the lower-level uncertainty for a broad class of bilevel knapsack problems. In particular, we formulate the leader's (upper-level) problem when it is known that the follower selects its algorithm from a family of greedy approaches or implements an exact solution approach (see Section 4.2 for more details). We show that, in general, the upper-level problem remains NP-hard even when the follower is known to use a greedy method for solving the lower-level problem. Also, we provide a single-level mixed integer programming (MIP) formulation to the leader's decision problem. Single-level formulations of bilevel programs are common in settings where the lower-level problem admits a linear programming (LP) formulation; see, e.g., Audet et al. (1997), Zare et al. (2019). Remarkably, we obtain such a representation when the follower applies a greedy solution approach.

Finally, we illustrate the potential impact of considering inexact followers through a series of numerical experiments on a specific class of bilevel knapsack problems. In particular, we consider a class of non-linear defender-attacker problems, which can be casted through the bilevel modeling framework (see Brown et al. (2006) and references therein), and study settings where the attacker might have limited computational resources and uses either an exact approach or one of two greedy-like approaches. Our results aim in general at illustrating the potential impact of relaxing full-rationality assumptions, and in particular to illustrate the algorithmic use of the proposed framework to help the leader in hedging against the lower-level uncertainty in the context of defender-attacker problems. While the settings we consider are not intended to represent any real-life instances of defender-attacker interactions, we exploit them to provide insights on the use of the framework, to support our discussion, and to validate our structural results.

Organization of the paper. The next section provides background material on bilevel optimization and presents the proposed modeling framework to address the lower-level uncertainty, while Section 3 analyzes the leader's loss in performance. Section 4 presents our prescriptive approach to a broad class of bilevel knapsack problems. Section 5 presents a numerical study for the case of the defender-attacker model when the defender (leader) does not know the solution approach taken by the attacker (follower) but knows that it belongs to a family of greedy approaches. Section 6 presents our conclusions and final remarks. Finally, the proofs of our theoretical results are relegated to appendix.

2. Modeling Framework for Inexact Follower

In this paper we consider a general class of bilevel mixed integer programs of the form

$$[\mathbf{BMIP}] \ \min_{\mathbf{y}\in\mathcal{Y}} \quad f(\mathbf{y},\mathbf{x}) := g(\mathbf{y})^\top \mathbf{x} + t(\mathbf{y}) \tag{1a}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{R}(\mathbf{y}) := \arg\max\left\{c(\mathbf{y})^\top \hat{\mathbf{x}} \mid \hat{\mathbf{x}} \in \mathbb{X}(\mathbf{y})\right\}, \tag{1b}$$

where $\mathcal{Y} \subseteq \{0,1\}^{n_1-k_1} \times \mathbb{R}_+^{k_1}$ denotes the leader's feasible region, while $g, c : \mathcal{Y} \to \mathbb{R}^{n_2}$ and $t : \mathcal{Y} \to \mathbb{R}$. Set $\mathbb{X}(\mathbf{y}) \subseteq \{0,1\}^{n_2-k_2} \times \mathbb{R}_+^{k_2}$ denotes the follower's feasible region given leader's decision $\mathbf{y} \in \mathcal{Y}$.

We refer to the minimization in (1a) as the upper-level problem, which is solved by the leader, and to the maximization on the right-hand-side of (1b) as the lower-level problem, which is solved by the follower. For each $\mathbf{y} \in \mathcal{Y}$, the set $\mathcal{R}(\mathbf{y}) \subseteq \{0,1\}^{n_2-k_2} \times \mathbb{R}_+^{k_2}$ is known as the *follower's rational* *reaction set.* The leader's and the follower's problems are general mixed integer programs (MIPs). Furthermore, if $k_1 = n_1$ and $k_2 = n_2$, g and c are constant vectors of appropriate dimensions, $t(\mathbf{y})$ is a linear function, and $\mathcal{Y} \times \mathbb{X}(\mathbf{y})$ is a polyhedral set, then **BMIP** reduces to a bilevel linear program (BLP). In contrast to the classical linear programming (LP) solvable in polynomial time, BLPs are NP-hard (Colson et al. 2007, Deng 1998).

When $k_1 = k_2 = 0$ and the set $\{(\mathbf{y},\mathbf{x}) \mid \mathbf{y} \in \mathcal{Y},\ \mathbf{x} \in \mathbb{X}(\mathbf{y})\} \subseteq \{0,1\}^{n_1+n_2}$ is non-empty, it can be shown that **BMIP** has an optimal solution (Vicente et al. 1996). However, in general, **BMIP** may fail to have an optimal solution because the *inducible region* $\Omega := \{(\mathbf{y},\mathbf{x}) \mid \mathbf{y} \in \mathcal{Y}, \mathbf{x} \in \mathcal{R}(\mathbf{y})\}$ may be a non-compact set (Vicente et al. 1996). To address this issue in the remainder of the paper we assume that:

Assumption A1: BMIP has an optimal solution.

In the bilevel optimization literature, various assumptions on the structure of the objective functions and feasible regions in **BMIP** are made so as to guarantee the existence of an optimal solution (i.e., an equilibrium solution) (Colson et al. 2007). We discuss some of them in Section 4.1 for a class of the bilevel knapsack problems used in our computational experiments. For more details on the existence of optimal solution in bilevel problems, we refer the readers to Dempe (2002), Vicente et al. (1996), Harker and Pang (1988), Mersha and Dempe (2006), Aboussoror et al. (2017), Aboussoror and Mansouri (2005).

Finally, whenever the lower-level problem has multiple optimal solutions for a given leader's decision, a "collaborative" follower might implement the solution that is the most favorable to the leader. On the other hand, an "adversarial" follower might select the most disadvantageous (to the leader) solution. These two situations are respectively referred to as the *optimistic* and *pessimistic* formulations of bilevel problems (Colson et al. 2007).

2.1. Inexact Follower

As outlined in Section 1, a key assumption in most studies in the bilevel optimization literature is that the follower's rational reaction set includes only optimal solutions of the lower-level problem (1b), see, e.g., some recent results in Beheshti et al. (2015), Caramia and Mari (2015), DeNegre and Ralphs (2009), Tang et al. (2015). However, in many application areas, given a leader's decision $\mathbf{y} \in \mathcal{Y}$, the resulting lower-level problem is NP-hard, which means that, in practice, an exact solution might not be found in a timely manner. Hence, in such settings the follower might use approximate or heuristic solutions instead, that are typically much faster to find.

Consider, for example, the aforementioned case of bilevel knapsack problems, where the lower-level problem takes the form of the linear 0–1 knapsack problem. Specifically, assume that for a given upper-level decision $\mathbf{y} \in \mathcal{Y}$, the lower-level problem is of the form:

$$\max\Big\{\sum_i c_i x_i \mid \sum_i w_i x_i \le b,\ \mathbf{x} \in \{0,1\}^{n_2}\Big\}. \tag{2}$$

While there exist multiple exact solution methods for the linear and nonlinear integer knapsack problems (e.g., dynamic programming or branch-and-bound based algorithms), there are also various approximation and heuristic algorithms (Kellerer et al. 2004). One prime example is the popular *greedy method* that simply sorts the items in the non-increasing order of the ratio c_i/w_i, and selects the items prioritizing them according to said ratio, subject to the budgetary constraint.[2]

While heuristic and approximate approaches (as the greedy method above) for NP-hard problems (like the knapsack problem) are ubiquitous in practice (see, e.g., examples in Gendreau and Potvin (2010), Pardalos and Resende (2001)), most research studies in bilevel optimization typically ignore the follower's practical considerations. More importantly, ignoring such a choice might prevent the leader from anticipating the follower's actions and thus have profound consequences. To illustrate this point, we provide the following example.

Example 1. Consider a simple bilevel problem, which is an instance of **BMIP**:

$$\begin{aligned}
\min_{\mathbf{y}\in\{0,1\}^2}\ & f(\mathbf{y},\mathbf{x}) = y_1 + My_2 + x_1 + 2Mx_2 \\
\text{subject to}\ & y_1 + y_2 = 1 \\
& \mathbf{x} \in \arg\max_{\hat{\mathbf{x}}\in\{0,1\}^2} \{2M\hat{x}_1 + M\hat{x}_2\ : M\hat{x}_1 + \hat{x}_2 \le My_1\},
\end{aligned}$$

where M is a sufficiently large constant. Observe that the leader has two feasible solutions given by $\mathbf{y}^1 = (1,0)^\top$ and $\mathbf{y}^2 = (0,1)^\top$. If the follower solves his problem to optimality, then the leader's optimal solution is $\mathbf{y}^1 = (1,0)^\top$, which triggers the follower's optimal reaction $\mathbf{x}^1 = (1,0)^\top$, resulting in the upper-level objective function value $f(\mathbf{y}^1,\mathbf{x}^1) = 2$. Next, consider a scenario where the follower uses the greedy heuristic based on the cost-to-weight ratio. If the leader implements $\mathbf{y}^1 = (1,0)^\top$, then the follower's response is $\mathbf{x}^2 = (0,1)^\top$. Consequently, the upper-level objective function value is $f(\mathbf{y}^1,\mathbf{x}^2) = 1 + 2M$. On the other hand, if the leader is aware of the fact that the follower applies the greedy heuristic, then she implements $\mathbf{y}^2 = (0,1)^\top$ resulting in the upper-level objective function value $f(\mathbf{y}^2,\mathbf{x}^3) = M$, where $\mathbf{x}^3 = (0,0)^\top$. Note that $f(\mathbf{y}^1,\mathbf{x}^2) - f(\mathbf{y}^2,\mathbf{x}^3) = M + 1$ and this difference between the resulting objective functions values can be made arbitrarily large. ■

[2] It is worth mentioning that while the worst-case performance guarantee for the greedy method can be made arbitrarily close 0, a small variation of the algorithm provides a 1/2-approximation. (Kellerer et al. 2004, Chapter 2.5)

In the remainder of the paper, we assume that in order to find a solution to the lower-level problem the follower uses one of the algorithms from a pre-defined set $\mathcal{H}$. Formally, algorithm $h \in \mathcal{H}$ maps upper-level decisions to lower-level decisions and hence, is characterized by the set of responses it produces (thus, two algorithms that reacts in the same way to all upper-level decisions are indistinguishable). Specifically:

Definition 1. A follower's *reaction algorithm* $h \in \mathcal{H}$ is given by function $\mathbf{x}^h(\cdot) : \mathcal{Y} \to \mathbb{R}^{n_2}$ that maps an upper-level decision $\mathbf{y} \in \mathcal{Y}$ to a feasible lower-level decision $\mathbf{x}^h(\mathbf{y})$ in $\mathbb{X}(\mathbf{y})$.

Note that by its definition, algorithm $h \in \mathcal{H}$ returns a *unique* feasible solution to the lower-level problem for every $\mathbf{y} \in \mathcal{Y}$, which is consistent with assumption **A1**. A key assumption in this work, which we formalize next, is that the leader does not know the algorithm that is used by the follower, but knows that it belongs to the set $\mathcal{H}$.

Assumption A2: The reaction algorithm, h, used by the follower is not known to the leader in advance. However, the leader is aware of the set of possible reaction algorithms, $\mathcal{H}$.

Furthermore, in what follows, we make the distinction between *exact* and *inexact* algorithms. We say that $h \in \mathcal{H}$ is an exact algorithm if for any $\mathbf{y} \in \mathcal{Y}$ it returns an optimal solution to the lower-level problem, i.e., $\mathbf{x}^h(\mathbf{y}) \in \mathcal{R}(\mathbf{y})$. Similarly, we say $h \in \mathcal{H}$ is an inexact algorithm if it might return a suboptimal solution to the lower-level problem, i.e., there exists $\mathbf{y} \in \mathcal{Y}$ such that $\mathbf{x}^h(\mathbf{y}) \notin \mathcal{R}(\mathbf{y})$. For the linear knapsack 0–1 problem, for example, the aforementioned greedy method is, in general, an inexact algorithm. In addition, we say that algorithm h is *distinct* from h' if they return different solutions to some instance of the lower-level problem, i.e., there exists $\mathbf{y} \in \mathcal{Y}$ such that $\mathbf{x}^h(\mathbf{y}) \neq \mathbf{x}^{h'}(\mathbf{y})$. Note that both h and h' might be exact but distinct at the same time, which is possible whenever $\mathcal{R}(\mathbf{y})$ is not a singleton, for some $\mathbf{y} \in \mathcal{Y}$.

2.2. Approaches to the Lower-level Algorithmic Uncertainty

Next, we introduce three modeling approaches to handle the lower-level algorithmic uncertainty.

Robust Model (RBP). In this approach, the leader anticipates the worst possible outcome over $\mathcal{H}$. That is, the leader assumes that for any given upper-level decision, an adversarial follower uses the algorithm that damages the leader the most. Thus, the leader solves

$$[\textbf{RBP}] \quad z^*_{\mathcal{H}} := \min_{\mathbf{y} \in \mathcal{Y}} \quad \max_{h \in \mathcal{H}} \quad f(\mathbf{y}, \mathbf{x}^h(\mathbf{y})) = g(\mathbf{y})^\top \mathbf{x}^h(\mathbf{y}) + t(\mathbf{y}). \tag{3}$$

Note that **RBP** can be viewed as a generalization of the pessimistic and optimistic cases of the standard bilevel optimization. For example, let h and h' be both exact algorithms for the follower's problem and assume that, for every $\mathbf{y} \in \mathcal{Y}$, algorithm h yields solutions that are most favorable for the leader, while h' returns the least favorable one: by setting $\mathcal{H} = \{h\}$ and $\mathcal{H} = \{h'\}$ in (3) we reduce **RBP** to either optimistic or pessimistic bilevel problems, respectively.

Γ-Robust Model (Γ-RBP). The approach in **RBP** can be seen as too conservative, especially when $\mathcal{H}$ contains several solution methods. Next, we propose a more flexible model that enables the leader to control her level of conservatism. Let Γ be a positive integer representing the number of algorithms that the leader wishes to "hedge" against, i.e., $\Gamma \in \{1, \ldots, |\mathcal{H}|\}$. Then for a fixed value of Γ, the leader solves

$$[\mathbf{\Gamma\text{-}RBP}]\ z^*_\Gamma := \min_{S \subseteq \mathcal{H}, \mathbf{y} \in \mathcal{Y}, z} \quad \left\{ z : f(\mathbf{y}, \mathbf{x}^h(\mathbf{y})) \leq z,\ \forall h \in S,\ \ |S| = \Gamma \right\}. \tag{4}$$

Note that in contrast to **RBP**, here the leader anticipates the Γ-th smallest realization among the follower's algorithms. Thus, **RBP** is a special case of **Γ-RBP** with $\Gamma = |\mathcal{H}|$. In other words, the leader minimizes the Γ-th smallest objective function among all values generated by the algorithms in $\mathcal{H}$. For example, for $\Gamma = 1$, the leader is effectively selecting the algorithm that the follower uses, while for $\Gamma = |\mathcal{H}|$, the leader anticipates that the follower uses the algorithm that hurts her the most, for any given upper-level decision. Simply speaking, in the **Γ-RBP** model the leader hedges against Γ algorithms from $\mathcal{H}$ and ignores $|\mathcal{H}| - \Gamma$ worst possible outcomes for her.

Hence, **Γ-RBP** can be re-written as the following mathematical program:

$$
\begin{aligned}
z^*_\Gamma := \min\ & \rho && \text{(5a)}\\
\text{subject to}\ \ & \rho + M(1 - \sigma_h) \geq f(\mathbf{y}, \mathbf{x}^h(\mathbf{y})) \qquad \forall h \in \mathcal{H}, && \text{(5b)}\\
& \sum_{h=1}^{|\mathcal{H}|} \sigma_h = \Gamma, && \text{(5c)}\\
& \mathbf{y} \in \mathcal{Y},\ \ \sigma_h \in \{0, 1\}\ \forall h \in \mathcal{H}, && \text{(5d)}
\end{aligned}
$$

where M is a sufficiently large constant parameter, e.g., $M := \max\limits_{h \in \mathcal{H}, \mathbf{y} \in \mathcal{Y}} \{f(\mathbf{y}, \mathbf{x}^h(\mathbf{y}))\}$. In (5) we assume that such M exists. Note that the expression $f(\mathbf{y}, \mathbf{x}^h(\mathbf{y}))$ might not be readily available and itself might be a solution of another mathematical program.

Γ-RBP model is inspired by the robust optimization approach to matrix-data uncertainty in Bertsimas and Sim (2003, 2004). In their work, the decision-maker hedges against any change to the matrix-data as long as it involves at most Γ changes to uncertain elements in the data matrix. In our work, the decision-maker instead hedges against the lower-level algorithmic uncertainty. In particular, in the **Γ-RBP** model, for a given upper-level decision, the leader anticipates that the follower uses the Γ-th most favorable (to her) algorithm in $\mathcal{H}$. According to intuition, the next result shows that the solution to **Γ-RBP** is non-decreasing in Γ.

Proposition 1. *The optimal objective function value of* **Γ-RBP**, z^*_Γ, *is non-decreasing in* Γ.

Proposition 1 formalizes the intuition that the leader's optimal objective function value deteriorates as she hedges against an increasing number of algorithms. As mentioned earlier, $\Gamma = |\mathcal{H}|$ corresponds to the most conservative follower, which is formalized by the following corollary.

Corollary 1. *For any integer Γ such that $1 \le \Gamma \le |\mathcal{H}|$:*

$$z^*_\Gamma \le z^*_{|\mathcal{H}|} = z^*_{\mathcal{H}}.$$

We illustrate **RBP** and **Γ-RBP** by means of the following example.

Example 2. Consider the following instance of a bilevel knapsack problem (see further discussion in Section 4):

$$\begin{aligned}
\min_{\mathbf{y}\in\mathbb{R}^4_+} \quad & (5-y_1)x_1 + (6-y_2)x_2 + (12-1.5y_3)x_3 + (17-2y_4)x_4 \\
\text{subject to} \quad & y_1+y_2+y_3+y_4 \le 10 \\
& \mathbf{x} \in \arg\max_{\hat{\mathbf{x}}\in\{0,1\}^4} \ (3M-y_1)\hat{x}_1 + (100-0.5y_2)\hat{x}_2 + (90-y_3)\hat{x}_3 + (20-y_4)\hat{x}_4 \\
& \qquad \text{subject to} \quad M\hat{x}_1 + 50\hat{x}_2 + 30\hat{x}_3 + 21\hat{x}_4 \le M+50.
\end{aligned}$$

Note that for a given upper-level decision $\mathbf{y}$, the lower-level problem reduces to a knapsack binary problem of the form (2). Let $\mathcal{H}$=$\{h_1,\ h_2,\ h_3\}$ be three algorithms that the follower might use, where h_1 is an exact algorithm and h_2 is a greedy algorithm for solving 0–1 knapsack problem based on the c_i/w_i ratio (recall our earlier discussion in Section 1); furthermore, h_3 is also a greedy algorithm but the follower prioritizes items based on the value of w_i.

For sufficiently large value of M, regardless of the upper-level decision, the response of each algorithm in $\mathcal{H}$ is given by $\mathbf{x}^{h_1} = (1,1,0,0)^\top$, $\mathbf{x}^{h_2} = (1,0,1,0)^\top$ and $\mathbf{x}^{h_3} = (0,1,1,1)^\top$. Thus, for upper-level decision $\mathbf{y}^1 = (10,0,0,0)^\top$ we have that $f(\mathbf{y}^1,\mathbf{x}^{h_1}(\mathbf{y}^1)) = 1$, $f(\mathbf{y}^1,\mathbf{x}^{h_2}(\mathbf{y}^1)) = 7$ and $f(\mathbf{y}^1,\mathbf{x}^{h_3}(\mathbf{y}^1)) = 35$. Similarly, upper-level decision $\mathbf{y}^2 = (0,0,10,0)^\top$ results in $f(\mathbf{y}^2,\mathbf{x}^{h_1}(\mathbf{y}^2)) = 11$, $f(\mathbf{y}^2,\mathbf{x}^{h_2}(\mathbf{y}^2)) = 2$ and $f(\mathbf{y}^2,\mathbf{x}^{h_3}(\mathbf{y}^2)) = 20$. According to the **RBP** model, $\mathbf{y}^2$ is a better solution for the leader compared with $\mathbf{y}^1$ because $\max_h\{f(\mathbf{y}^2,\mathbf{x}^h(\mathbf{y}^2))\} = 20 \le \max_h\{f(\mathbf{y}^1,\mathbf{x}^h(\mathbf{y}^1))\} = 35$. If the leader applies the **Γ-RBP** model for $\Gamma = 2$, then $z_\Gamma(\mathbf{y}^1) = 7$ and $z_\Gamma(\mathbf{y}^2) = 11$, which implies that $\mathbf{y}^1$ is a better solution than $\mathbf{y}^2$ when $\Gamma = 2$ in **Γ-RBP**.

The optimal solution of **RBP** is $\mathbf{y}^*_{\mathcal{H}} = (0,0,1,9)^\top$, $z^*_{\mathcal{H}} = 15.5$ and z^*_Γ for different values of Γ is as follows: $z^*_{\Gamma=1} = 1$, $z^*_{\Gamma=2} = 5$ and $z^*_{\Gamma=3} = z^*_{\mathcal{H}}$. As expected from Proposition 1 and Corollary 1, we have: $z^*_{\Gamma=1} \le z^*_{\Gamma=2} \le z^*_{\Gamma=3} = z^*_{\mathcal{H}}$. ■

Probabilistic model (PBP). In **RBP** and **Γ-RBP** the leader does not use any prior information about the likelihood that the follower uses a particular algorithm. This type of information, if available, may help the leader to identify a better solution. Next, we suppose that the likelihood that the follower applies any given algorithm is available to the leader. We assume that the leader uses this information to minimize her expected objective function value. That is, the leader solves

$$[\mathbf{PBP}] \ z^*_p = \min_{\mathbf{y}\in\mathcal{Y}} \quad \mathbb{E}_h[f(\mathbf{y},\mathbf{x}^h(\mathbf{y}))] := \sum_{h\in\mathcal{H}} p_h f(\mathbf{y},\mathbf{x}^h(\mathbf{y})), \tag{6}$$

where p_h denotes the known probability that the follower uses algorithm $h \in \mathcal{H}$, i.e., $0 \leq p_h \leq 1$ and $\sum_{h \in \mathcal{H}} p_h = 1$.

Recall our earlier observation that the optimistic and pessimistic models of **BMIP** can be viewed as special cases of **RBP**. A similar generalization holds for **PBP**. Specifically, let methods h and h' be both exact algorithms for the follower's problem, but suppose that while h returns the solution that is most favorable for the leader, h' returns the least favorable one; then, for $\mathcal{H} = \{h, h'\}$ model **PBP** corresponds to a bilevel model, where the leader optimizes a convex combination of the leader's objective functions in the optimistic and pessimistic cases. Such setting has been considered in the bilevel optimization literature, mostly for bilevel linear programs, and is known as a *strong-weak* bilevel problem (Aboussoror and Loridan 1995, Cao and Leung 2002, Zare et al. 2018, Zheng et al. 2015).

In a sense, the strong-weak approach attempts to model settings with a partially collaborative follower, where the decision-maker knows the probabilities of cooperation or non-cooperation of the follower, respectively, i.e., the leader is not certain if the follower is either collaborative or adversarial, and thus attempts to make a robust decision by taking into account both situations. We note that the strong-weak model is a special case of **PBP**.

Proposition 2. *Let $z^*_{\mathcal{H}}$ and z^*_p be the optimal values of* **RBP** *and* **PBP**, *respectively. Then $z^*_p \leq z^*_{\mathcal{H}}$.*

Example 3. Let $p = (p_{h_1}, p_{h_2}, p_{h_3}) = (0.3, 0.2, 0.5)$ in Example 2. Then, $\mathbf{y}^*_p = (0, 0, 10, 0)^\top$ and $z^*_p = 13.7$. Thus, we have that $z^*_p = 13.7 \leq z^*_{\mathcal{H}} = 15.5$ which is aligned with Proposition 2. ■

The results of Proposition 2 have a simple intuitive interpretation. If the leader has some initial information (i.e., p_h for all $h \in \mathcal{H}$), then this information can be used to decrease her expected objective function value in comparison to the case when she needs to hedge against the worst possible outcome.

With regard to the existence of optimal solutions for the formulations above, in the same spirit as Assumption **A1** in the sequel we assume that:

Assumption A3: RBP, **Γ-RBP** for any $\Gamma \in \{1, \ldots, |\mathcal{H}|\}$ and **PBP** have optimal solutions.

In Section 4 we discuss the validity of Assumptions **A1** and **A3** for the class of bilevel knapsack problems used in our numerical illustration in Section 5.

3. Quantifying Leader's Loss

In this section we explore the consequences, for the leader, of making erroneous assumptions about the follower's reaction method. For this, below we formally define the notion of the *leader's loss* and then explore its properties.

Let $\mathbf{y}^h$ denote the leader's optimal decision when the follower uses algorithm h, and let f_h^* be the leader's corresponding objective function value, i.e,

$$\mathbf{y}^h \in \arg\min_{\mathbf{y}\in\mathcal{Y}}\{f(\mathbf{y},\mathbf{x}^h(\mathbf{y}))\}, \quad f_h^* := f(\mathbf{y}^h,\mathbf{x}^h(\mathbf{y}^h)).$$

Definition 2. The leader's loss for a decision $\mathbf{y}\in\mathcal{Y}$ and algorithm $h\in\mathcal{H}$, is given by:

$$\Delta_h(\mathbf{y}) := f(\mathbf{y},\mathbf{x}^h(\mathbf{y})) - f_h^*. \tag{7}$$

Note that $\Delta_h(\mathbf{y}) \geq 0$ and $\Delta_h(\mathbf{y}^h) = 0$. Also, we define $\Delta_{hh'}$ as the leader's loss when the leader acts assuming that the follower uses algorithm h while he instead uses algorithm h'. That is,

$$\Delta_{hh'} := \Delta_{h'}(\mathbf{y}^h).$$

Using Definition 2, we can think of **PBP** as a model in which the leader minimizes her expected total loss. Indeed, we have that

$$\min_{\mathbf{y}\in\mathcal{Y}} \ \mathbb{E}_{h\in\mathcal{H}}[\Delta_h(\mathbf{y})] = \min_{\mathbf{y}\in\mathcal{Y}} \ \mathbb{E}_{h\in\mathcal{H}}[f(\mathbf{y},\mathbf{x}^h(\mathbf{y})) - f_h^*] = \min_{\mathbf{y}\in\mathcal{Y}} \sum_{h\in\mathcal{H}} p_h f(\mathbf{y},\mathbf{x}^h(\mathbf{y})) - \sum_{h\in\mathcal{H}} p_h f_h^*, \tag{8}$$

which is equivalent to **PBP** as the second term in the right-hand side of (8) is a constant that is independent of the upper-level decision.

An alternative view of the leader's loss follows from comparing the realized upper-level objective function value with that anticipated by the leader.

Definition 3. The ex-post (leader's) loss $\Delta^{\mathrm{A}}_{hh'}$ from anticipating the use of algorithm $h\in\mathcal{H}$ when the follower's response is actually computed using algorithm $h'\in\mathcal{H}$ is given by

$$\Delta^{\mathrm{A}}_{hh'} = \max\left\{f(\mathbf{y}^h,\mathbf{x}^{h'}(\mathbf{y}^h)) - f_h^*, 0\right\}. \tag{9}$$

Loosely speaking, the ex-post loss compares the objective function value attained with that expected. Thus, in some situations the follower might not react as anticipated but this results in an improvement in the leader's objective function value (relative to the value that would have been obtained were the follower to react as anticipated). In such situations the ex-post loss $\Delta^{\mathrm{A}}_{hh'}$ is zero. This situation is illustrated in the next example.

Example 4. Table 1 reports the leader's and the follower's optimal solutions for Example 2 given the same set of possible follower's solution methods. We compute the values of $\Delta^{\mathrm{A}}_{hh'}$ and $\Delta_{hh'}$ and represent them in matrices Δ^{A} and Δ. For example, we have that $\mathbf{y}^{h_1} = (10,0,0,0)^\top$, $\mathbf{x}^{h_2}(\mathbf{y}^{h_1}) = (1,0,1,0)^\top$, and $f(\mathbf{y}^{h_1},\mathbf{x}^{h_2}(\mathbf{y}^{h_1})) = 7$. Thus, $\Delta^{\mathrm{A}}_{h_1h_2} = 7-1 = 6$, and $\Delta_{h_1h_2} = 7-2 = 5$. Furthermore:

$$\Delta^{\mathrm{A}} = \begin{bmatrix} 0 & 6 & 34 \\ 9 & 0 & 18 \\ 0 & 2 & 0 \end{bmatrix}; \qquad \Delta = \begin{bmatrix} 0 & 5 & 20 \\ 10 & 0 & 5 \\ 10 & 15 & 0 \end{bmatrix}.$$

■

Table 1 Optimal solutions for different follower's reaction methods in Example 2.

h	$\mathbf{x}^h$	$\mathbf{y}^h$	f_h^*
h_1	$(1,1,0,0)^\top$	$(10,0,0,0)^\top$	1
h_2	$(1,0,1,0)^\top$	$(0,0,10,0)^\top$	2
h_3	$(0,1,1,1)^\top$	$(0,0,0,10)^\top$	15

The following lemmas establish a series of logical relationships between the loss and ex-post loss.

LEMMA 1. *For any pair of algorithms $h, h' \in \mathcal{H}$ one has that $f_h^* \geq f_{h'}^* \Leftrightarrow \Delta_{hh'}^{\mathrm{A}} \leq \Delta_{hh'}$, and $f_h^* \leq f_{h'}^* \Rightarrow \Delta_{hh'}^{\mathrm{A}} \geq \Delta_{hh'}$.*

LEMMA 2. *For any pair of algorithms $h, h' \in \mathcal{H}$, if $\Delta_{hh'}^{\mathrm{A}} = 0$, then $f_{h'}^* \leq f_h^*$ and $\Delta_{hh'} \leq f_h^* - f_{h'}^*$.*

Lemma 2 provides an upper bound for $\Delta_{hh'}$ when the upper-level objective function value is not larger than anticipated. Note that $\Delta_{hh'}^{\mathrm{A}} = 0$ does not necessarily imply $\Delta_{hh'} = 0$ (which is illustrated further in Example 5 below). In other words, when the leader's objective function value is smaller than or equal to what she anticipated, then we cannot necessarily conclude that the leader has implemented the best possible decision.

The next proposition provides a bound on the objective function value attained by the leader when implementing the solution prescribed by **RBP** for $|\mathcal{H}| = 2$.

PROPOSITION 3. *Suppose that $\mathcal{H} = \{h, h'\}$. Then*

$$z_{\mathcal{H}}^* \leq \min\left\{f_h^* + \Delta_{hh'}^{\mathrm{A}}, f_{h'}^* + \Delta_{h'h}^{\mathrm{A}}\right\}. \tag{10}$$

Moreover, if $\Delta_{hh'}^{\mathrm{A}} = 0$, then $z_{\mathcal{H}}^* = f_h^*$.

Proposition 3 allows us to compare **RBP** with **BMIP** when $|\mathcal{H}| = 2$. In particular, we see that when $\Delta_{hh'}^{\mathrm{A}} = 0$ both problems have the same optimal objective function value. However, if $\Delta_{hh'}^{\mathrm{A}} > 0$, then inequality (10) reduces to $f(\mathbf{y}_{\mathcal{H}}^*, \mathbf{x}^{h'}(\mathbf{y}_{\mathcal{H}}^*)) \leq f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h))$. Thus, if the follower uses algorithm h', then the leader is better off implementing an optimal solution of **RBP**, $\mathbf{y}_{\mathcal{H}}^*$ rather than $\mathbf{y}^h$. This observation is rather natural as it is implied by the intuition behind model **RBP**. Note that Example 5 below illustrates the fact that $z_{\mathcal{H}}^* = f_h^*$ does not necessarily indicate that $f_h^* = f_{h'}^*$. Also, some of the structural properties are illustrated in this example.

The next corollary, which we state without proof, establishes an upper bound for $z_{\mathcal{H}}^*$ in terms of $\Delta_{hh'}$ instead of $\Delta_{hh'}^A$.

COROLLARY 2. *Suppose that $\mathcal{H} = \{h, h'\}$. If $\Delta_{hh'}^{\mathrm{A}} \geq 0$ and $\Delta_{h'h}^{\mathrm{A}} \geq 0$, then*

$$z_{\mathcal{H}}^* \leq \min\left\{f_{h'}^* + \Delta_{hh'}, f_h^* + \Delta_{h'h}\right\}.$$

Example 5. Based on the information provided in Table 1 and Example 4, $f^*_{h_1} \leq f^*_{h_2}$ and $\Delta^{\mathrm{A}}_{h_1h_2} \geq \Delta_{h_1h_2}$. This is consistent with Lemma 1. Moreover, $\Delta^{\mathrm{A}}_{h_3h_1} = 0$ and $\Delta_{h_3h_1} = 10$ which reflects the fact that $\Delta^{\mathrm{A}}_{h_3h_1} = 0$ does not necessarily result in $\Delta_{h_3h_1} = 0$. In addition, $\Delta_{h_3h_1} = 10 \leq f^*_{h_3} - f^*_{h_1} = 15 - 1$ and it illustrates Lemma 2. Next, we show that $z^*_{\mathcal{H}} = f^*_{h_1}$ does not necessarily imply that $f^*_{h_1} = f^*_{h_2}$. Let $\mathcal{H} = \{h_1, h_2\}$, and if the coefficient of x_3 in the leader's objective function is changed from $(12 - 1.5y_3)$ to $(6 - 1.1y_3)$, then $\Delta^{\mathrm{A}}_{h_1h_2} = 0$ and $z^*_{\mathcal{H}} = f^*_{h_1} = 1$. However, $f^*_{h_2} = 0 \neq f^*_{h_1}$.

Finally, for $\mathcal{H} = \{h_1, h_2\}$ in Example 2, $\mathbf{y}^*_{\mathcal{H}} = (6, 0, 4, 0)^\top$ and $z^*_{\mathcal{H}} = 5$. According to Proposition 3, $f(\mathbf{y}^*_{\mathcal{H}}, \mathbf{x}^{h_2}(\mathbf{y}^*_{\mathcal{H}})) \leq f^*_{h_1} + \Delta^{\mathrm{A}}_{h_1h_2}$, i.e., $5 \leq 1 + 6$, and $f(\mathbf{y}^*_{\mathcal{H}}, \mathbf{x}^{h_1}(\mathbf{y}^*_{\mathcal{H}})) \leq f^*_{h_2} + \Delta^{\mathrm{A}}_{h_2h_1}$, i.e., $5 \leq 2 + 9$. Similarly, based on Corollary 2, $f(\mathbf{y}^*_{\mathcal{H}}, \mathbf{x}^{h_2}(\mathbf{y}^*_{\mathcal{H}})) \leq f^*_{h_2} + \Delta_{h_1h_2}$, i.e., $5 \leq 2 + 5$, and $f(\mathbf{y}^*_{\mathcal{H}}, \mathbf{x}^{h_1}(\mathbf{y}^*_{\mathcal{H}})) \leq f^*_{h_1} + \Delta_{h_2h_1}$, i.e., $5 \leq 1 + 10$. ■

The results in Proposition 3 can be extended for the optimal solution of **PBP**, as shown next.

Proposition 4. *Suppose that $\mathcal{H} = \{h, h'\}$ and that $p_{h'} > 0$ and $p_h > 0$. We have that*

$$z^*_p \leq p_{h'}(f^*_{h'} + \Delta_{hh'}) + p_h(f^*_h + \Delta_{h'h}).$$

4. Bilevel Knapsack Problem

In this section we apply the framework developed in Sections 2 and 3 to a special class of **BMIP** known as the bilevel knapsack problem (**BKP**), which can be written in the following form:

$$[\mathbf{BKP}] \ \min_{\mathbf{y}} \ f(\mathbf{y}, \mathbf{x}) := \sum_{i=1}^{n_2} \big(g_{i0} + \sum_{j=1}^{n_1} g_{ij} y_j\big) x_i + \sum_{j=1}^{n_1} t_j y_j \tag{11a}$$

$$\text{subject to} \quad \mathbf{y} \in \mathcal{Y} := \{\mathbf{y} : \ A\mathbf{y} \leq b, \mathbf{y} \in \{0,1\}^k \times \mathbb{R}^{n_1 - k}_+\}, \tag{11b}$$

$$\mathbf{x} \in \mathcal{R}(\mathbf{y}) := \arg\max_{\hat{\mathbf{x}}} \left\{ \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij} y_j\big) \hat{x}_i : \ w^\top \hat{\mathbf{x}} \leq d, \hat{\mathbf{x}} \in \{0,1\}^{n_2} \right\}, \tag{11c}$$

where $A \in \mathbb{R}^{m \times n_1}$, $b \in \mathbb{R}^{m \times 1}$, $w \in \mathbb{R}^{n_2 \times 1}_{>0}$, $d \in \mathbb{R}^1_{>0}$. Note that the functions $g_i(\cdot)$, $c_i(\cdot)$ and $t(\cdot)$ are affine with respect to $\mathbf{y}$, where $g_{i0}, \ g_{ij}, \ c_{i0}, \ c_{ij}, \ t_j \in \mathbb{R}$ for all $i \in \{1, \ldots, n_2\}$ and $j \in \{1, \ldots, n_1\}$. In order to simplify the notation, our definition of **BKP** contemplates a single constraint in the follower's knapsack problem. However, the prescriptive approach offered in this section admits a rather straightforward generalization to multiple constraints.

In the remainder of the paper we make the following assumption:

Assumption A4: Set $\mathcal{Y}$ is non-empty and bounded.

Assumption **A4** simply states that the leader's feasible set is non-empty and bounded. Later in this section, we show that **A4** together with the observation that $\mathbf{x} = \mathbf{0}$ is a feasible solution for the follower's problem (and thus (11c) has an optimal solution for any $\mathbf{y} \in \mathcal{Y}$) imply Assumption **A1** for the case of **BKP**.

BKP is used to illustrate the proposed framework for two reasons. First, the bilevel knapsack problem (11) and its variants form a well-known class of bilevel optimization problems (see, e.g., Beheshti et al. (2015), Brotcorne et al. (2009), Caprara et al. (2013, 2014, 2016), Dempe and Richter (2000), Özaltın et al. (2010)). In particular, our numerical examples in Section 5 are based on a special class of **BKP** that has military and law-enforcement applications in the context of the defender-attacker models. The second reason is that, for any upper-level decision $\mathbf{y} \in \mathcal{Y}$, the follower's problem reduces to a linear 0–1 knapsack problem, which is known to be NP-hard (Garey and Johnson 1979). The 0–1 knapsack problem is one of the most studied combinatorial optimization problems, mainly because of its simple integer programming formulation, its recurrent appearance in the study of more complex problems, and its capability of representing various real-life decision situations (Martello and Toth 1990). More importantly, in practical settings, the 0–1 knapsack problem is often solved by applying greedy heuristic approaches (recall our earlier examples in the previous sections): see Section 4.2 for a detailed discussion.

In what follows, we assume that each algorithm in $\mathcal{H}$ fulfills minimum *local optimality* conditions.

Assumption A5: For any $h \in \mathcal{H}$ we have that: (i) if the i^{th} component of $c(\mathbf{y})$ is non-positive, then $x_i^h(\mathbf{y}) = 0,\ i = 1, \ldots, n_2$; and (ii) if $x_i^h(\mathbf{y}) = 0$, then $\tilde{\mathbf{x}}^h(\mathbf{y}) = (x_1^h, \ldots, x_{i-1}^h, 1, x_{i+1}^h, \ldots, x_{n_2}^h)$ is infeasible, i.e., $w^\top \tilde{\mathbf{x}}^h(\mathbf{y}) > d$.

Assumption **A5** states that algorithms in $\mathcal{H}$ do not pack items with clearly unfavorable costs, and do not generate solutions that can be easily improved by adding a single item. Next, we show that when Assumption **A5** holds, **BKP** remains NP-hard even if the follower applies an inexact solution algorithm from $\mathcal{H}$.

Proposition 5. **BKP** *remains NP–hard when the maximization on the r.h.s. of* (11c) *is solved using any algorithm h for which* **A5** *holds.*

It is worth noting that Proposition 5 holds for any set of approximation and heuristic algorithms, as long as Assumption **A5** holds. This result can be extended to the other proposed formulations. We formalize this in the following corollary, which we state without proof.

Corollary 3. **RBP**, **PBP***and* **BMIP** *for any fixed* $\Gamma \in \{1, \ldots, |\mathcal{H}|\}$, *are NP-hard.*

Next, we introduce a family of greedy algorithms for solving the linear 0–1 knapsack problem and present a single-level formulation of **BKP** for selecting the upper-level decisions when the follower uses one of such greedy methods. Before that, we briefly discuss the case when the follower uses an exact algorithm to solve the lower-level problem (we use such a model in the numerical experiments in Section 5).

4.1. BKP with an exact follower

Consider the case of an exact follower, i.e., we assume that $\mathcal{H} = \{h\}$, with h exact. We briefly describe a cutting plane algorithm for solving **BKP** based on its single-level relaxation. Specifically, the latter is given by the problem of the form:

$$\begin{aligned}
[\mathbf{SKP}]\ \min_{\mathbf{y},\mathbf{x}}\ & f(\mathbf{y},\mathbf{x}) := \sum_{i=1}^{n_2} \big(g_{i0} + \sum_{j=1}^{n_1} g_{ij} y_j\big) x_i + \sum_{j=1}^{n_1} t_j y_j \\
\text{subject to}\quad & A\mathbf{y} \le b, \\
& w^\top \mathbf{x} \le d, \\
& \mathbf{y} \in \{0,1\}^k \times \mathbb{R}_+^{n_1-k},\ \mathbf{x} \in \{0,1\}^{n_2},
\end{aligned}$$

where a single decision-maker controls both sets of decision variables and the follower's objective function is completely disregarded. Thus, **SKP** is a single-level mathematical program and referred to as a single-level relaxation of **BKP**.

Clearly, the optimal objective function value of **SKP** provides a lower bound for the optimal objective function value of **BKP**. Note that solution approaches based on exploiting single-level relaxations as bounding mechanisms are among the most common approaches in the bilevel optimization literature, see, e.g., a recent example in Caramia and Mari (2015) for solving bilevel linear integer problems. In this section, we demonstrate an application of this approach for a class of nonlinear bilevel problem given by **BKP**.

In particular, we observe that **SKP** is a nonlinear mixed integer problem due to the presence of nonlinear terms $y_j x_i$ in its objective function. However, these terms can be linearized by introducing new variables z_{ij} and additional set of linear constraints (see, further details and discussion in Adams and Forrester (2005)):

$$\begin{aligned}
&\big\{(x_i, y_j, z_{ij})\ :\ z_{ij} = y_j x_i,\ x_i \in \{0,1\},\ y_j^L \le y_j \le y_j^U\big\} = \\
&\quad \big\{(x_i, y_j, z_{ij})\ :\ x_i \in \{0,1\},\ z_{ij} \le y_j^U x_i,\ z_{ij} \le y_j + y_j^L x_i - y_j^L,\ z_{ij} \ge y_j^L x_i,\ z_{ij} \ge y_j + y_j^U x_i - y_j^U\big\},
\end{aligned}$$

where we assume that the lower (y_j^L) and upper (y_j^U) bounds on y_j for each $j \in \{1,\dots,n_1\}$ are either readily available or can be easily computed. (The bounds exist due to Assumption **A4**: note that tighter bounds could improve solution times). Hence, **SKP** can be re-written as an equivalent linear MIP that can be solved by a standard solver. This observation also implies that **SKP** has a finite optimal solution.

The pseudo-code of the exact cutting-plane based approach for solving **BKP** is provided in Algorithm 1, whose convergence is established in the next result under Assumption **A4**.

Proposition 6. *Algorithm 1 outputs an optimal solution of* **BKP** *in a finite number of steps.*

Algorithm 1 Exact Algorithm for solving **BKP**

```
Step 1. Solve SKP and denote by $(\hat{\mathbf{y}}, \hat{\mathbf{x}})$ its optimal solution.
Step 2. Solve linear binary problem (11c) for $\mathbf{y} = \hat{\mathbf{y}}$. Let $\check{\mathbf{x}}$ and $z_f^*$ denote its optimal solution and the optimal objective function value, respectively.
if $z_f^* = \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j\big)\hat{x}_i$ then
    $(\hat{\mathbf{y}}, \hat{\mathbf{x}})$ is an optimal solution of BKP; STOP.
end if
if $z_f^* > \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j\big)\hat{x}_i$ then
    Go to Step 3.
end if
Step 3. Add a linear constraint of the form:
$\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}y_j\big)x_i \geq \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}y_j\big)\check{x}_i$ to SKP and go to Step 1.
```

This result implies that Assumption **A1** holds for **BKP** (which is a class of **BMIP**) under Assumption **A4**. Finally, we note that exact solution approaches based on single-level relaxations along with cutting plane and/or branch-and-bound based ideas are very common in the bilevel literature (see, e.g. Caramia and Mari (2015), Audet, Savard and Zghal (2007), Audet, Haddad and Savard (2007), DeNegre and Ralphs (2009), Tahernejad et al. (2018)). While the vast majority of such approaches focus on linear or linear mixed integer bilevel problems, our model contains non-linear terms, which can be trivially handled for our models using standard linearization techniques. More importantly, this cutting plane algorithm can be extended to solve models from Section 2.2, see our discussion next.

4.2. BKP with a greedy follower

4.2.1. The greedy follower. The 0–1 knapsack problem, one of the most studied combinatorial problems, can not be solved in polynomial time (unless $P = NP$). However, because of its relevance to practice, multiple exact and approximate solution algorithms have been proposed, many of which are used in practice.

In this section, we focus our analysis on a simple version of the greedy algorithm, see Kellerer et al. (2004). In its simplest form, the greedy algorithm first ranks the available alternatives (referred to as items in the context of knapsack problems) based on their *cost-to-weight* ratio c_i/w_i, where

$$c_i \equiv c_i(\mathbf{y}) = c_{i0} + \sum_{j=1}^{n_1} c_{ij}y_j,$$

and then goes through the ranking (in decreasing order, so items with greater ranking are preferred), selecting items if their inclusion does not violate the capacity constraint $w^\top \mathbf{x} \leq d$.

More generally, the follower may use different *rating functions* (besides the cost-to-weight ratio) in a hierarchical fashion to rank items (so that ties in the overall ranking are broken using ratings hierarchically), see, e.g., Hogarth and Karelaia (2005). In what follows, we use such a generalization of the greedy algorithm.

Specifically, we say that a rating function $r := (r_i(c_i, w_i), i = 1, \ldots, n_2) \in \mathbb{R}^{n_2}$ assigns rating $r_i \equiv r_i(c_i, w_i) \in [0,1]$ to item i, which is a function of its cost and weight. We consider a set of rating functions $\{r^1, \ldots, r^K\}$ for some positive integer K. Define $\mathcal{K} := \{1, \ldots, K\}$ and

$$k_{ij} := \arg\max\{k \ : \ r_i^\ell = r_j^\ell \text{ for all } \ell < k, \ k \in \mathcal{K}\} \tag{13}$$

for any pair of distinct items (i, j), where $r_i^0 = 0$ for all i.

Let "$\succ$" denote the *preference* relation among items, i.e., $i \succ j$ denotes that i is preferred (regarding selection) over j. The preference relationship is such that

$$r_i^{k_{ij}} > r_j^{k_{ij}} \quad \Longleftrightarrow \quad i \succ j.$$

Note that by (13) there may exist a tie in rankings between i and j if and only if $k_{ij} = K$ and $r_i^K = r_j^K$. For simplicity of exposition we assume that such scenarios do not occur.

The above discussion implies that ratings are used in a lexicographic fashion to define the overall ranking. That is, rankings r^1, …, r^K are ordered according to the follower's preferences. With the above, we assume that, for $\mathbf{y}$ given, the follower solves (11c) using the *Greedy Heuristic* described below.

Algorithm 2 Greedy Heuristic

```
Step 1: Order items according to "$\succ$." Relabel items so that $i \succ i+1$ for all $i$. Let $i \leftarrow 1$.
Step 2: Pick item $i$ if its selection does not violate the knapsack constraint. Let $i \leftarrow i+1$.
Step 3: If $i \le n_2$, go to Step 2. Otherwise, return the obtained solution.
```

4.2.2. Single-level MIP reformulation.

Single-level reformulations for bilevel programs are common in settings where the lower-level problem admits an LP formulation (see, e.g., Audet et al. (1997) and Zare et al. (2019)). In particular, the strong duality property of LPs is usually exploited to derive single-level MIP reformulations that can be handled by standard MIP solution methods. For more complex lower-level problems (e.g., general MIPs at the lower level) single-level reformulations are not typically available (at least not polynomially sized) when the follower uses an exact algorithm. Next, we leverage the structure of the follower's greedy heuristic to provide a single-level reformulation of **BKP** with irrational follower.

Decision variables. Let $N = \{1,\ldots,n_2\}$. For item $i \in N$, let binary variable x_i represent the selection of item i, i.e.

$$x_i = \begin{cases} 1 & \text{if item } i \text{ selected,} \\ 0 & \sim . \end{cases}$$

We consider various auxiliary (binary) variables for distinct items i and j: α_{ij} represents whether item i is preferred to item j; q_{ij} denotes that $i \succ j$ and item i is selected; and z_{ij}^k signals whether there is a tie in the k-th rating function between items i and j. This is,

$$\alpha_{ij} = \begin{cases} 1 & \text{if } i \succ j, \\ 0 & \sim, \end{cases}, \quad q_{ij} = \alpha_{ij}\, x_i, \quad z_{ij}^k = \begin{cases} 1 & \text{if } r_i^k = r_j^k, \\ 0 & \sim . \end{cases}$$

Constraints. First, we force preferences to align with the set of rating functions. For this, we begin ordering items using ranking function r^1 by considering the following set of constraints.

$$r_i^1 \le r_j^1 + \alpha_{ij} \qquad \forall i,j \in N \tag{14a}$$

$$\alpha_{ij} + \alpha_{ji} = 1 \qquad \forall i,j \in N \tag{14b}$$

$$q_{ij} \le \alpha_{ij} \qquad \forall i,j \in N \tag{14c}$$

$$\sum\nolimits_{j=1}^{n} q_{ij} \le n x_i \qquad \forall i \in N \tag{14d}$$

$$\alpha_{ij} + x_i \le q_{ij} + 1 \qquad \forall i,j \in N \tag{14e}$$

Constraints (14a)-(14b) provide consistency to the value of α_{ij}, when $r_i^1 \neq r_j^1$. Constraints (14c)–(14e) assure that q_{ij} is equal to x_i when $i \succ j$, and zero otherwise.

When $r_i^1 = r_j^1$, we require α_{ij} to be consistent with the remaining rating functions. We do this via the following set of constraints.

$$-(1 - z_{ij}^k) \le r_i^k - r_j^k \le (1 - z_{ij}^k) \qquad \forall i,j \in N, k \in \mathcal{K} \tag{15a}$$

$$r_j^k - r_i^k \le \sum\nolimits_{h=1}^{k-1}(1 - z_{ij}^h) + z_{ij}^k + (1 - \alpha_{ij}) - \delta^k \qquad \forall i,j \in N, k \in \mathcal{K} \tag{15b}$$

$$z_{ij}^k = z_{ji}^k \qquad \forall i,j \in N, k \in \mathcal{K}, \tag{15c}$$

where $\delta^k = \min\{|r_i^k - r_j^k| \text{ s.t. } r_i^k \neq r_j^k\}$. Note that $z_{ij}^k = 1$ in (15a) implies that $r_i^k = r_j^k$. Moreover, if $r_i^k = r_j^k$ and $k < k_{ij}$, then (15b)-(15c) imply that $z_{ij}^k = 0$ is infeasible. Also, note that (15b)-(15c) are trivially satisfied for $k \ge k_{ij}$ because $z_{ij}^h = 1$ for some $h < k_{ij}$.

Finally, we consider the knapsack constraints limiting item selection.

$$\sum\nolimits_{i=1}^{n} w_i x_i \le d \tag{16a}$$

$$w_i \le d - \sum\nolimits_{\substack{t=1 \\ t \neq i}}^{n} w_t q_{ti} + M(1 - x_i) \qquad \forall i \in N \tag{16b}$$

$$w_i + M x_i \ge d - \sum\nolimits_{\substack{t=1 \\ t \neq i}}^{n} w_t q_{ti} + \delta \qquad \forall i \in N, \tag{16c}$$

where M is a sufficiently large constant. Constraint (16a) ensures that item selection satisfies the follower's knapsack constraint. In addition, constraints (16b)-(16c) enforce item i to be selected when there is enough space left by the selection of items preferred to i. In this constraint, we have that $\delta = \min_i \{w_i - u_i\}$ and u_i is as follows:

$$u_i = \max_S \{d - \sum_{t\in S} w_t \ : \ 0 \le d - \sum_{t\in S} w_t < w_i \quad \forall S \subseteq N, i \notin S\}.$$

The next proposition formalizes the correctness of the formulation (we omit its proof as it is embedded in the discussion above).

Proposition 7. *For any fixed* $\mathbf{y}$*,* $\mathbf{x}$ *is a greedy solution if and only if it is a feasible solution of inequalities* (14a)-(16c)*.*

From Proposition 7, **BKP** with a greedy follower admits the following single-level reformulation:

$$\begin{aligned}
[\textbf{g-BKP}] \quad \min_{\mathbf{y},\mathbf{x},\alpha,\mathbf{q},\mathbf{z}} \quad & f(\mathbf{y},\mathbf{x}) := \sum_{i=1}^{n_2} \big(g_{i0} + \sum_{j=1}^{n_1} g_{ij} y_j\big) x_i + \sum_{j=1}^{n_1} t_j y_j \\
\text{subject to} \quad & (14\text{a}) - (14\text{e}), \\
& (15\text{a}) - (15\text{c}), \\
& (16\text{a}) - (16\text{c}), \\
& \alpha_{ij}, q_{ij}, z_{ij}^k, x_i \in \{0,1\} \qquad\qquad \forall i,j \in N, \ \forall k \in \mathcal{K}
\end{aligned}$$

4.2.3. BKP with exact and greedy followers. From the above derivations, we conclude that if $\mathcal{H}$ includes only greedy-like algorithmic methods, then we can reformulate **RBP** and **Γ-RBP** as a single-level MIP problem. In addition, if $\mathcal{H}$ also includes an exact method, then model (5) and Algorithm 1 can be exploited to find solutions for **RBP** and **Γ-RBP**.

Specifically, assume, for example, that $\mathcal{H}$ in **Γ-RBP** includes an exact (optimistic) method, h_1, and several greedy approaches, h_2, …, $h_{|\mathcal{H}|}$. We define extra binary variables $\mathbf{x}^h$ for all $h \in \mathcal{H}$. To solve **Γ-RBP**, we need to add the following constraints to problem (5): the leader's constraint (11b), inequalities (14a)–(16c) for $\mathbf{x}^{h_2}, \ldots, \mathbf{x}^{h_{|\mathcal{H}|}}$, and $w^\top \mathbf{x}^{h_1} \le d$. Because $\mathbf{x}^{h_1}$ reflects the follower's exact method, then we simply apply Algorithm 1, where in Step 1 we solve the modified problem (5) as a single-level relaxation of the original problem. In Step 2 the stopping criteria is evaluated to verify whether $\mathbf{x}^{h_1}$ is an exact solution of the follower's problem. Finally, in Step 3 we add cuts to ensure that the algorithm converges to the appropriate values of $\mathbf{x}^{h_1}$, i.e., an optimal follower's solution given leader's decision $\mathbf{y}$.

Finally, we note that the above discussion implies that Assumption **A3** holds for the **BKP** models considered in this section, which can be shown using the approach similar to the one used for Proposition 6 under Assumption **A4**.

5. Numerical Illustration

In this section we illustrate the modeling framework and structural results established in Sections 2 and 3 by a series of numerical experiments.

5.1. Defender-Attacker Problem (DAP)

We consider a class of the defender-attacker problems that can be formulated as **BKP**, where we apply the solution techniques in Section 4. **DAP** is an important and well studied problem in bilevel optimization, see, e.g. Brown et al. (2006), Zare et al. (2018) and references therein. Interesting results on this class and other related classes of bilevel problems can be found in Borrero et al. (2016), Guan et al. (2017), Samuel and Guikema (2012).

We consider a **DAP** variant in which a defender, as the leader, allocates defensive resources among the various facilities in a set I to reduce a total restoration cost (subject to a defense budget B), and the attacker, as the follower, selects facilities to attack. More precisely, on the upper-level, the defender incurs on a cost of $g_{i0} - g_i\ y_i$ to restore facility $i \in I$ after an attack, where g_{i0} represents the cost of restoring facility i if unprotected when attacked, g_i denotes a marginal cost reduction per unit of defensive resource, and y_i denotes the defensive resources allocated to facility $i \in I$. In addition, we let b_i the marginal cost of allocating a unit of defensive resource to facility i. The defender's objective is to minimize the total recovery cost.

On the lower level, for a given defensive resource allocation, the attacker selects targets among the same various facilities, so as to maximize the total damage inflicted by attacking said facilities; the damage inflicted by attacking facility $i \in I$ (as perceived by the attacker) is given by $c_{i0} - c_i y_i$, where c_{i0} denotes the base damage inflicted to an unprotected facility i, and c_i is a marginal damage reduction per unit of defensive resource. The attacker aims at maximizing the total damage inflicted, subject to a total budget on attacking resources. In this regard, we let w_i denote the amount of said resources necessary to attack facility $i \in I$, and K the overall budget on attacking resources.

Assuming a rational follower (in the sequel, we refer to the defender (attacker) and leader (follower) interchangeably), the **DAP** described above can be formulated as follows.

$$[\textbf{DAP}] \quad \min_{\mathbf{y}\in\mathbb{Y}} \quad f_1(\mathbf{y},\mathbf{x}) := \sum_{i\in I}(g_{i0} - g_i y_i)x_i \tag{18a}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{R}(\mathbf{y}) := \operatorname*{arg\,max}_{\hat{\mathbf{x}}\in\{0,1\}^{|I|}} \left\{ f_2(\mathbf{y},\hat{\mathbf{x}}) = \sum_{i\in I}(c_{i0} - c_i y_i)\hat{x}_i \mid \sum_{i\in I} w_i \hat{x}_i \le K \right\}, \tag{18b}$$

where $\mathbb{Y} := \{y \in \mathbb{R}_+^{|I|} :\ g_{i0} - g_i y_i \ge 0,\ c_{i0} - c_i y_i \ge 0\ \forall i \in I,\ \sum_{i\in I} b_i y_i \le B\}$ denotes the feasible region of possible defensive resource allocations, while the follower's (attacker's) decision variable, $\hat{x}_i$, is equal to 1 if and only if facility i attacked. Note that **DAP** is a class of **BKP** and Assumption **A4** holds.

In our computational experiments we use randomly generated instances of **DAP** where all parameters are integers generated as follows: $g_{i0} \sim U[0,100]$, $c_{i0} \sim U[0,50]$, $g_i, c_i \sim U[0,2]$, $b_i, w_i \sim U[0,20]$ for all $i \in I$, where $U[\cdot,\cdot]$ denotes a discrete uniform distribution. Furthermore, we let $B = \sum_{i\in I} b_i$ and $K = 0.5\sum_{i\in I} w_i$. All experiments are conducted on an Intel Xenon PC with 3.7 GHz CPU and 32 GB of RAM, and MIPs are solved using CPLEX 12.4 (CPLEX (2016)).

5.2. Results and Discussion

Note that for a given upper-level decision (i.e., a defensive resource allocation), the attacker's problem in **DAP** reduces to the standard 0-1 knapsack problem. The attacker, solves his problem either exactly or by using a greedy approach outlined in Section 4.2. Specifically, in our experiments the attacker's set of alternative solution methods, $\mathcal{H}$, consists of an exact method, h_1, and two Greedy Heuristics, h_2 and h_3. Thus, $|\mathcal{H}| = 3$. We assume that for h_2 the rating functions are given by $r_i^1(c_i, w_i) = c_i$ and $r_i^2(c_i, w_i) = w_i$, while for h_3 the rating functions are $r_i^1(c_i, w_i) = c_i/w_i$ (i.e., the classical cost-to-weight ratio) and $r_i^2(c_i, w_i) = w_i$, where $i \in I$.

5.2.1. Exploring Γ-RBP and RBP. In this set of experiments we explore how the leader's optimal decisions both under **Γ-RBP** or **RBP** and the follower's responses affect the objective function values at both levels. Specifically, we consider instances of **DAP** with $|I| = 15$. To develop a better understanding of the proposed approaches, first we study two instances of **DAP** and then, we implement the same experiment over thirty instances of **DAP**. The results of our experiments are depicted in Figures 1(a)-(i).

The results for the first instance of **DAP** are given in Figures 1(a)-(c). Specifically, Figure 1(a) displays the leader's objective function values, f_1, when the leader implements $\mathbf{y}^*_\Gamma$, $\Gamma \in \{1,2,3\}$. The follower responds using methods h_1, h_2 and h_3; thus, for each $\mathbf{y}^*_\Gamma$ there are three bars in Figure 1(a), each corresponding to one of the follower's solution methods. Similarly, Figure 1(b) depicts the follower's objective function values, f_2, given his responses via one of the methods. The leader's loss values, $\Delta_h(\mathbf{y}^*_\Gamma)$, are illustrated in Figure 1(c) for $\Gamma \in \{1,2,3\}$ and different methods h_1, h_2 and h_3.

Recall that by the definition of **Γ-RBP**, the defender takes into account only Γ out of $|\mathcal{H}|$ possible solution methods of the attacker. Thus, for $\Gamma = 1$ in Figure 1(a), the defender takes into account only method h_2 and disregards h_1 and h_3. Consequently, as the defender's hedges only against the best possible outcome, her objective function attains the best possible value, $z_{\Gamma=1}$, if she implements $\mathbf{y}^*_{\Gamma=1}$ and the attacker responds using h_2. Note that in this case, the defender's loss, $\Delta_{h_2}(\mathbf{y}^*_{\Gamma=1})$, is equal to zero. On the other hand, if the defender's guess about the attacker's response is incorrect (i.e., the attacker's uses either h_1 or h_3) then her losses can be rather substantial, which can be observed by comparing $\Delta_{h_1}(\mathbf{y}^*_{\Gamma=1})$ and $\Delta_{h_3}(\mathbf{y}^*_{\Gamma=1})$ against $\Delta_{h_2}(\mathbf{y}^*_{\Gamma=1})$ in Figure 1(c). Also, it is quite intuitive that the attacker obtains his best possible objective function values (i.e., he inflicts the most damage to the defender) whenever the leader has an incorrect assumption about the attacker's method, see, e.g., the values of f_2 with h_1 and h_3 for $\Gamma = 1$ in Figure 1(b).

In Figure 1(a) for $\Gamma = 2$ the defender takes into account two out of three of the possible solution methods used by the attacker, which, in this instance, turn out to be h_1 and h_3. The defender's objective function value, z_Γ, increases, which is consistent with Proposition 1.

The case of $\Gamma = |\mathcal{H}| = 3$ corresponds to the most conservative defender, where **Γ-RBP** reduces to **RBP** as she hedges against all three possible solution methods used by the attacker. Clearly, as the defender hedges against all three solution methods her objective function in the worst case for $\mathbf{y}^*_{\Gamma=3}$ is better than the worst cases of $\mathbf{y}^*_{\Gamma=1}$ and $\mathbf{y}^*_{\Gamma=2}$. Note also that Corollary 1 is illustrated in Figures 1(a), as for any value of $\Gamma \in \{1,2,3\}$, $z^*_\Gamma \leq z^*_{\mathcal{H}} = z^*_{\Gamma=3}$.

Figure 1(c) depicts losses $\Delta_h(\mathbf{y}^*_\Gamma)$, $h \in \{h_1, h_2, h_3\}$. These losses are caused by lower-level uncertainty. Thus, we can interpret these values as the "value of information" for the defender regarding lower-level uncertainty.

Another instance is illustrated in Figures 1(d)-1(f). These results are consistent with those depicted in Figures 1(a)-1(c). Recall that whenever the defender solves model **Γ-RBP**, she does not hedge against a fixed subset of the attacker's methods, but rather ensures that Γ out of them are taken into account, while $|\mathcal{H}| - \Gamma$ worst outcomes for the defender are discarded. Thus, it is worth pointing out that for the same value of $\Gamma \in \{1,2\}$ in Figures 1(a) and 1(d) the defender takes into account different subsets of the attacker's solution methods.

Finally, in order to develop a deeper insight about the proposed models, we present a similar illustration in Figures 1(g)-1(i) where the results are obtained from thirty **DAP** instances. All values are normalized to interval $[0,1]$ and the average value is reported. These results are consistent with those depicted in Figures 1(a)-1(f). Note that, because we report the average values, $z^*_{\mathcal{H}}$ does not necessarily correspond to $\max_h \ f(\mathbf{y_{\Gamma=3}}^*, \mathbf{x}^h(\mathbf{y_{\Gamma=3}}^*))$. We show these averages by $\hat{z}^*_{\mathcal{H}}$ and $\hat{z}^*_\Gamma$.

5.2.2. The leader's loss analysis. In this set of experiments, we provide a more detailed exploration of the defender's losses under different scenarios. In Figure 2(a) we depict the defender's loss ratio, $\Delta_h(\mathbf{y})/f^*_h$, where the attacker selects a method from h_1, h_2 or h_3 to respond to the defender's decision, $\mathbf{y}$. The latter is assumed to be computed based on one of the following six methods. In the first three, the defender assumes that the attacker always selects a specific method h and thus, she implements $\mathbf{y}^h$. In the next three, she decides based on the **Γ-RBP** model, where $\Gamma \in \{1,2,3\}$ and implements $\mathbf{y}^*_\Gamma$. Furthermore, Figure 2(b), depicts the defender's ex-post average loss ratio, $\Delta^{\mathrm{A}}_{h'h}/f^*_h$, for $h, h' \in \{h_1, h_2, h_3\}$. The results for both figures are obtained for thirty **DAP** instances, where $|I| = 15$ and $|\mathcal{H}| = 3$, and the average loss ratio is reported. In each figure the error bars represent the confidence interval for the mean values for a significance level of $\alpha = 0.05$.

In Figure 2(a) the first three bars, for each of the defender's solution method h_1, h_2 and h_3, represent the leader's loss ratio due to her incorrect guess about the attacker's response. If her guess is correct, then by definition, $\Delta_h(\mathbf{y}^h) = 0$ (see Definition 2), and consequently her loss ratio is zero. For example, in Figures 2(a), there is no "blue bar" for attacker's method h_1, meaning that when the defender implements $\mathbf{y}^{h_1}$ and the attacker responds based on method h_1, then the defender's loss ratio is zero. Otherwise, the defender's loss can be rather significant when her guess about the attacker's behavior is incorrect. For example, see the "blue bar" for h_2 in Figures 2(a) which represents $\Delta_{h_2}(\mathbf{y}^{h_1})/f^*_{h_2}$.

On the other hand, Figures 2(a) illustrates how employing **Γ-RBP** to hedge against all attacker's potential responses, influences the defender's loss ratio. Note that, because in $\Gamma = 1$ the defender ignores two possible responses, her loss still can be large. See, for example the "gray bar" for methods h_1 and h_3. However, her loss ratio decreases by increasing the value of Γ to $\Gamma = 2$ and $\Gamma = 3$. For example, for any attacker's method, the defender's loss ratio for $\Gamma = 3$, "green bar," is among the smallest values of loss ratios.

Finally, Figures 2(b) displays the leader's ex-post loss ratio, $\Delta^{\mathrm{A}}_{h'h}/f^*_h$, under different situations. If the attacker applies a method which is anticipated by the defender, then by definition the defender's ex-post loss value is zero (see Definition 3). Note that, defender's ex-post loss ratio can be larger than her actual loss ratio. For example, compare the "blue bar" in Figures 2(b), for attacker's method h_3, with the corresponding value in Figures 2(a). In other words, even when the defender's objective functions is much smaller than she expected, in fact, her actual losses are not necessarily that substantial.

5.2.3. Comparing RBP, Γ-RBP and PBP.

Finally, in the last set of our experiments, we compare the defender's expected loss value, see (8), when she applies one of the **PBP**, **Γ-RBP** (for $\Gamma = 2$) or **RBP** models, i.e., she implements as her decisions $\mathbf{y}^*_p$, $\mathbf{y}^*_{\Gamma=2}$ or $\mathbf{y}^*_{\Gamma=3}$, respectively. These experiments illustrate the effect of incorporating p_h into our framework, on reducing the defender's expected loss value. In other words, if the leader has some additional information, then she can exploit it to implement decisions that potentially reduce her expected losses. In our context, this information consists of probabilities of implementing a particular solution method by the attacker.

We use a specific instance of **DAP** in which $|I| = 15$ and $\mathcal{H} = \{h_1, h_2, h_3\}$. The parameters of the defender's problem are $\mathbf{g}_0 = [110\ \ 30\ \ 35\ \ 3.5\ \ \ldots\ \ 3.5\ \ 5\ \ 10]^\top$, $\mathbf{g} = [3\ \ 3\ \ 2\ \ \ldots\ \ 2\ \ 5]^\top$, $b_i = 1$ for all $i \in I$ and $B = 5$. In addition, for the attacker's problem we have $\mathbf{c}_0 = [15M\ 15M\ 5000\ 720\ 605\ 500\ 405\ 320\ 245\ 180\ 125\ 80\ 45\ 20\ 0]$, $\mathbf{c} = [0\ 0\ 0\ 12\ 11\ 10\ 9\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 0]^\top$, $\mathbf{w} = [M\ \ M+1\ \ 5000\ \ 60\ \ 55\ \ 50\ \ 45\ \ 40\ \ 35\ \ 30\ \ 25\ \ 20\ \ 15\ \ 10\ \ 5]^\top$ and $K = M + 5001$, where M is a sufficiently large constant. The defender's optimal solutions for different attacker's responses is presented in Table 2.

Table 2 Leader's optimal solution and her corresponding objective function value for follower's different solution methods for the DAP instance used in Figure 3

h	$\mathbf{y}^h$	f_h^*
h_1	$\mathbf{y}^{h_1} = (0\ 5\ 0\ \ldots\ 0)^\top$	50
h_2	$\mathbf{y}^{h_2} = (5\ 0\ \ldots\ 0)^\top$	130
h_3	$\mathbf{y}^{h_3} = (0\ \ldots\ 0\ 5)^\top$	135

The defender's expected loss value, $\mathbb{E}_h[\Delta_h(\mathbf{y})]$, is illustrated in Figures 3(a)-(f) as a function of p_h, $h \in \{h_1, h_2, h_3\}$, where we fix the value of p_h for a specific method in each figure. For a fixed defender's decision $\mathbf{y}$, the value of $f(\mathbf{y}, \mathbf{x}^h(\mathbf{y}))$ can be computed for any method $h \in \mathcal{H}$, and then $\mathbb{E}_h[\Delta_h(\mathbf{y})]$ is a linear function of p_h, see equation (8). Note that for **Γ-RBP** and **RBP**, the defender's decisions do not depend on the information available to **PBP**, i.e., the probability distributions. Thus, given decisions $\mathbf{y}^*_{\Gamma=2}$ and $\mathbf{y}^*_{\Gamma=3}$, in **Γ-RBP** and **RBP**, the defender's expected losses are linear function of p_h, which can be observed in Figures 3(a)-(f).

On the other hand, for the **PBP** model the defender incorporates this additional information into her decision making, and thus, the value of $\mathbf{y}^*_p$ changes for different values of p_h, $h \in \mathcal{H}$. Consequently, $\mathbb{E}_h[\Delta_h(\mathbf{y})]$ is a piece-wise linear function of p_h in **PBP**, see Figures 3(a)-(f).

Furthermore, compared to **Γ-RBP**, model **PBP** always results in smaller values of expected losses. This observation is not surprising, given that the defender's objective function in **PBP** is equivalent to minimizing the expected loss value, see (8). This reduction in the expected losses can be interpreted as the value of additional information available to the defender.

Recall that for **Γ-RBP**, only Γ methods out $|\mathcal{H}|$ are taken into account by the defender. In Figure 3, for the **Γ-RBP** model, we set $\Gamma = 2$, and thus, one of the attacker's solution methods is disregarded by the defender. Consequently, whenever the probability of implementing this method is sufficiently high (small), the expected losses of **RBP** are smaller (higher) than those of **Γ-RBP**.

For example, in Figure 3(c), **Γ-RBP** hedges against h_1 and h_2, while h_3 is disregarded. The value of $p_{h_1} = 0$ in Figure 3(c). Thus, when the value of p_{h_2} is sufficiently small, $0 \le p_{h_2} \le 0.4$, the corresponding probability of implementing h_3 is rather high. Consequently, the expected losses of **Γ-RBP** are worse than those of both the **RBP** and **PBP** models. On the other hand, as the value of p_{h_2} increases, given that $p_{h_1} = 0$, the value of p_{h_3} decreases resulting in better and worse expected losses for **Γ-RBP** and **RBP**, respectively.

6. Conclusion

A traditional and key assumption in the standard bilevel optimization modeling framework is that the follower solves his problem optimally. However, there are many practical application settings where this assumption is not likely to hold. In this paper, we propose an approach for addressing this issue. By assuming that a set of possible follower's solution methods is known, we propose three modeling approaches, namely, **RBP**, **Γ-RBP** and **PBP**, that allow the leader to hedge against different response scenarios at the lower level, which we refer to as the lower-level algorithmic uncertainty.

Among the proposed approaches, the **RBP** model is the most conservative one as it hedges against all possible follower's solution methods. On the other hand, the **Γ-RBP** model allows the leader to control the level of her conservatism through a fixed parameter Γ. Finally, the **PBP** model assumes that some additional probabilistic information is available to the leader, who exploits it in the decision-making process.

We explore theoretical properties of the proposed models, and illustrate its potential applicability using a broad class of the bilevel knapsack problems in the context of the defender-attacker model. Our results indicate that the proposed approaches allow the leader to substantially reduce her losses whenever the follower's actual behaviour is not known precisely.

With respect to the future research directions, it would be valuable to derive additional single-level reformulations of bilevel problems with irrational followers where the lower-level algorithmic uncertainty extends beyond the use of greedy heuristics. Another interesting direction includes settings where the leader and the follower interact repeatedly over time (see, e.g., Borrero et al. (2016, 2019)), and hence the leader might infer information regarding the method used by the follower based on his response to the leader decisions.

Acknowledgements. This material is based upon work partially supported by the National Science Foundation [Grants CMMI-1400009 and CMMI-1634835], DoD DURIP grant FA2386-12-1-3032, the Air Force Research Laboratory (AFRL) Mathematical Modeling, Optimization Institute and the Air Force Office of Scientific Research (AFOSR), and the Complex Engineering Systems Institute, ISCI (CONICYT: PIA FB0816). Oleg Prokopyev would like to thank Prof. Ted Ralphs for discussions and comments on the earlier version of the manuscript. Finally, we authors would like to thank the Associate Editor and Anonymous Reviewers for their helpful and constructive comments, which have served to greatly improve the manuscript.

References

Aboussoror, A., Adly, S. and Saissi, F. E. (2017), 'Strong-weak nonlinear bilevel problems: existence of solutions in a sequential setting', *Set-Valued and Variational Analysis* **25**(1), 113–132.

Aboussoror, A. and Loridan, P. (1995), 'Strong-weak stackelberg problems in finite dimensional spaces', *Serdica Mathematical Journal* **21**(2), 151–170.

Aboussoror, A. and Mansouri, A. (2005), 'Weak linear bilevel programming problems: existence of solutions via a penalty method', *Journal of mathematical analysis and applications* **304**(1), 399–408.

Adams, W. and Forrester, R. (2005), 'A simple recipe for concise mixed 0-1 linearizations', *Operations Research Letters* **33**(1), 55–61.

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

Audet, C., Haddad, J. and Savard, G. (2007), 'Disjunctive cuts for continuous linear bilevel programming', *Optimization Letters* **1**(3), 259–267.

Audet, C., Hansen, P., Jaumard, B. and Savard, G. (1997), 'Links between linear bilevel and mixed 0-1 programming problems', *Journal of Optimization Theory and Applications* **93**(2), 273–300.

Audet, C., Savard, G. and Zghal, W. (2007), 'New branch-and-cut algorithm for bilevel linear programming', *Journal of Optimization Theory and Applications* **134**(2), 353–370.

Balas, E. and Zemel, E. (1980), 'An algorithm for large zero-one knapsack problems', *Operations Research* **28**(5), 1130–1154.

Bard, J. F. (1998), *Practical bilevel optimization: algorithms and applications. Nonconvex optimization and its applications.*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Bard, J. F., Plummer, J. and Sourie, J. C. (2000), 'A bilevel programming approach to determining tax credits for biofuel production', *European Journal of Operational Research* **120**(1), 30–46.

Beheshti, B., Özaltın, O., Zare, M. H. and Prokopyev, O. A. (2015), 'Exact solution approach for a class of nonlinear bilevel knapsack problems', *Journal of Global Optimization* **61**(2), 291–310.

Bertsimas, D. and Sim, M. (2003), 'Robust discrete optimization and network flows', *Mathematical Programming* **98**(1), 49–71.

Bertsimas, D. and Sim, M. (2004), 'The price of robustness', *Operations Research* **52**(1), 35–53.

Borrero, J. S., Prokopyev, O. A. and Saure, D. (2016), 'Sequential shortest path interdiction with incomplete information', *Decision Analysis* **13**(1), 68–98.

Borrero, J. S., Prokopyev, O. A. and Saure, D. (2019), 'Sequential interdiction with incomplete information and learning', *Operations Research* . To appear.

Brotcorne, L., Hanafi, S. and Mansi, R. (2009), 'A dynamic programming algorithm for the bilevel knapsack problem', *Operations Research Letters* **37**(3), 215–218.

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

Burgard, A., Pharkya, P. and Maranas, C. (2003), 'Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization', *Biotechnology and Bioengineering* **84**(6), 647–657.

Cao, D. and Leung, L. (2002), 'A partial cooperation model for non-unique linear two-level decision problems', *European Journal of Operational Research* **140**(1), 134–141.

Caprara, A., Carvalho, M., Lodi, A. and Woeginger, G. J. (2013), A complexity and approximability study of the bilevel knapsack problem, *in* 'Integer programming and combinatorial optimization', Springer, pp. 98–109.

Caprara, A., Carvalho, M., Lodi, A. and Woeginger, G. J. (2014), 'A study on the computational complexity of the bilevel knapsack problem', *SIAM Journal on Optimization* **24**(2), 823–838.

Caprara, A., Carvalho, M., Lodi, A. and Woeginger, G. J. (2016), 'Bilevel knapsack with interdiction constraints', *INFORMS Journal on Computing* **28**(2), 319–333.

Caramia, M. and Mari, R. (2015), 'Enhanced exact algorithms for discrete bilevel linear problems', *Optimization Letters* **9**(7), 1447–1468.

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

Côté, J.-P. and Savard, G. (2003), 'A bilevel modelling approach to pricing and fare optimization in the airline industry', *Journal of Revenue and Pricing Management* **2**(1), 23–26.

CPLEX, I. I. (2016). Available at http://www-01.ibm.com/software/info/ilog/. Accessed on January 7, 2016.

Dempe, S. (2002), *Foundations of bilevel programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Dempe, S. and Richter, K. (2000), 'Bilevel programming with knapsack constraints', *Central European Journal of Operations Research* **8**(2), 93–107.

DeNegre, S. T. and Ralphs, T. K. (2009), 'A branch-and-cut algorithm for bilevel integer programming', *Proceedings of the Eleventh INFORMS Computing Society Meeting* pp. 65–78.

Deng, X. (1998), Complexity issues in bilevel linear programming, *in* P. Pardalos, A. Migdalas and P. Varbrand, eds, 'Multilevel Optimization: Algorithms and Applications', Springer, pp. 149–164.

Fayard, D. and Plateau, G. (1982), 'An algorithm for the solution of the 0–1 knapsack problem', *Computing* **28**(3), 269–287.

Fudenberg, D. and Tirole, J. (1991), *Game Theory*, MIT Press, Cambridge, MA.

Garey, M. and Johnson, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco.

Gendreau, M. and Potvin, J.-Y. (2010), *Handbook of metaheuristics*, Vol. 2, Springer.

Guan, P., He, M., Zhuang, J. and Hora, S. C. (2017), 'Modeling a multitarget attacker–defender game with budget constraints', *Decision Analysis* **14**(2), 87–107.

Gzara, F. (2013), 'A cutting plane approach for bilevel hazardous material transport network design', *Operations Research Letters* **41**(1), 40–46.

Harker, P. T. and Pang, J.-S. (1988), 'Existence of optimal solutions to mathematical programs with equilibrium constraints', *Operations research letters* **7**(2), 61–64.

Hogarth, R. M. and Karelaia, N. (2005), 'Simple models for multiattribute choice with many alternatives: When it does and does not pay to face trade-offs with binary attributes', *Management Science* **51**(12), 1860–1872.

Horowitz, E. and Sahni, S. (1974), 'Computing partitions with applications to the knapsack problem', *J. ACM* **21**(2), 277–292.

Ibarra, O. H. and Kim, C. E. (1975), 'Fast approximation algorithms for the knapsack and sum of subset problems', *J. ACM* **22**(4), 463–468.

Kellerer, H., Pferschy, U. and Pisinger, D. (2004), *Knapsack Problems*, Springer, Berlin.

Martello, S. and Toth, P. (1977), 'An upper bound for the zero-one knapsack problem and a branch and bound algorithm', *European Journal of Operational Research* **1**(3), 169 – 175.

Martello, S. and Toth, P. (1988), 'A new algorithm for the 0-1 knapsack problem', *Management Science* **34**(5), 633–644.

Martello, S. and Toth, P. (1990), *Knapsack problems*, Wiley and Sons, Chichester, England.

Mersha, A. G. and Dempe, S. (2006), 'Linear bilevel programming with upper level constraints depending on the lower level solution', *Applied mathematics and computation* **180**(1), 247–254.

Nikoofal, M. E. and Zhuang, J. (2012), 'Robust allocation of a defensive budget considering an attacker's private information', *Risk Analysis: An International Journal* **32**(5), 930–943.

Nikoofal, M. E. and Zhuang, J. (2015), 'On the value of exposure and secrecy of defense system: First-mover advantage vs. robustness', *European Journal of Operational Research* **246**(1), 320–330.

Özaltın, O. Y., Propkopyev, O. A. and Schaefer, A. J. (2010), 'The bilevel knapsack problem with stochastic right-hand sides', *Operations Research Letters* **38**(4), 328–333.

Pardalos, P. M. and Resende, M. G. (2001), *Handbook of applied optimization*, Oxford university press.

Ren, S., Zeng, B. and Qian, X. (2013), 'Adaptive bilevel programming for optimal gene knockouts for targeted overproduction under phenotypic constraints', *BMC Bioinformatics* **14**(Suppl 2), S17.

Samuel, A. and Guikema, S. D. (2012), 'Resource allocation for homeland defense: Dealing with the team effect', *Decision Analysis* **9**(3), 238–252.

Shan, X. and Zhuang, J. (2013), 'Hybrid defensive resource allocations in the face of partially strategic attackers in a sequential defender–attacker game', *European Journal of Operational Research* **228**(1), 262–272.

Shen, S., Smith, J. C. and Goli, R. (2012), 'Exact interdiction models and algorithms for disconnecting networks via node deletions', *Discrete Optimization* **9**(3), 172–188.

Smith, J. C., Lim, C. and Sudargho, F. (2007), 'Survivable network design under optimal and heuristic interdiction scenarios', *Journal of Global Optimization* **38**(2), 181–199.

Tahernejad, S., DeNegre, S. and Ralphs, T. (2018), 'MiBS Version 1.1'.

Tang, Y., Richard, J. and Smith, J. (2015), 'A class of algorithms for mixed-integer bilevel min–max optimization', *Journal of Global Optimization* pp. 1–38.

Toth, P. (1980), 'Dynamic programming algorithms for the zero-one knapsack problem', *Computing* **25**(1), 29–45.

Vicente, L. N., Savard, G. and Judice, J. (1996), 'Discrete linear bilevel programming problem', *Journal of Optimization Theory and Applications* **89**(3), 597–614.

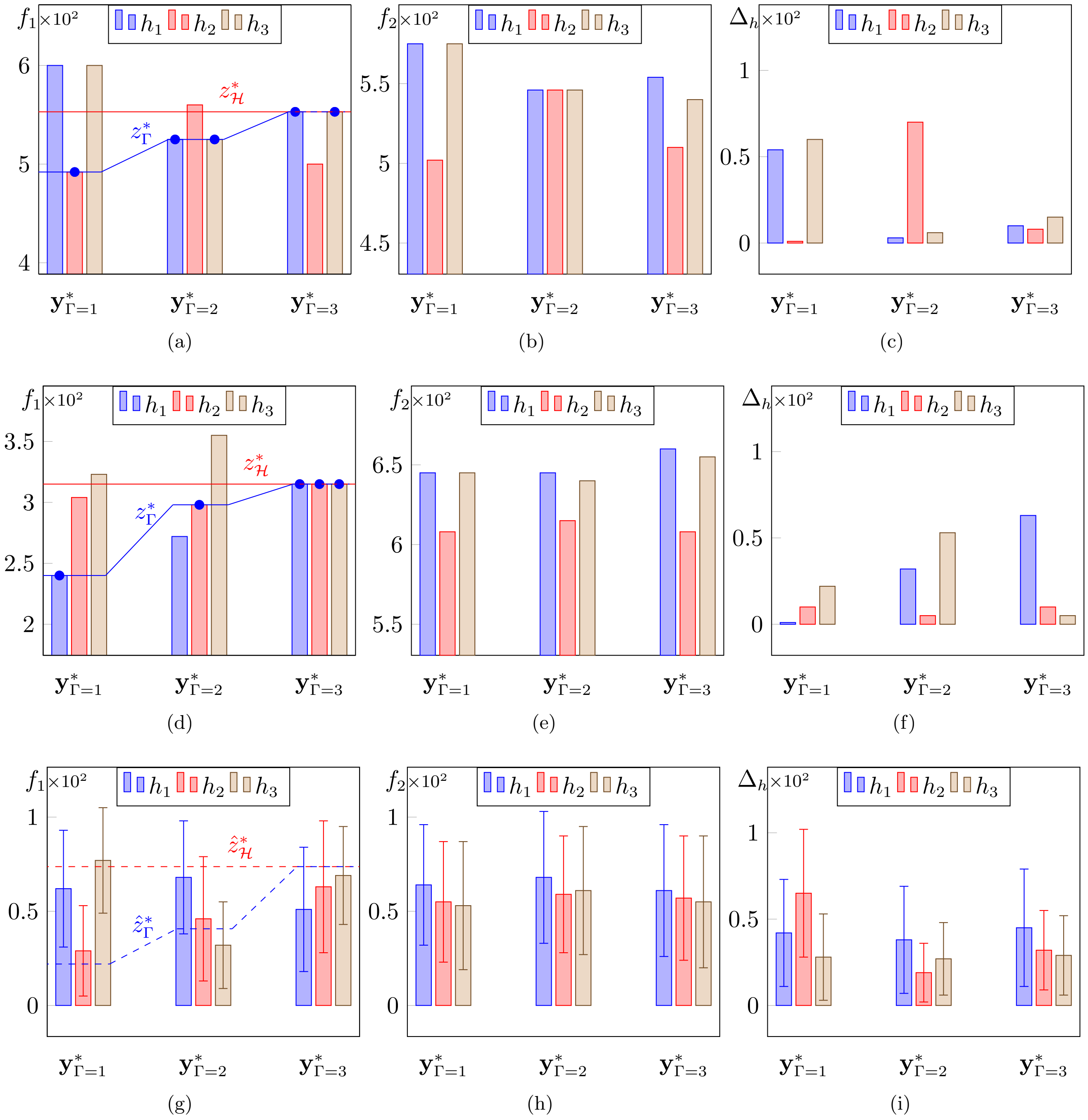
Wood, R. (1993), 'Deterministic network interdiction', *Mathematical and Computer Modelling* **17**(2), 1–18.

Zare, M. H., Borrero, J. S., Zeng, B. and Prokopyev, O. A. (2019), 'A note on linearized reformulations for a class of bilevel linear integer problems', *Annals of Operations Research* **272**(1-2), 99117.

Zare, M. H., Özaltın, O. Y. and Prokopyev, O. A. (2018), 'On a class of bilevel linear mixed-integer programs in adversarial settings', *Journal of Global Optimization* **71**(1), 91–113.

Zhang, J., Zhuang, J. and Behlendorf, B. (2018), 'Stochastic shortest path network interdiction with a case study of arizona–mexico border', *Reliability Engineering & System Safety* **179**, 62–73.

Zheng, Y., Wan, Z., Jia, S. and Wang, G. (2015), 'A new method for strong-weak linear bilevel programming problem', *Journal of Industrial and Management Optimization* **11**(2), 529–547.

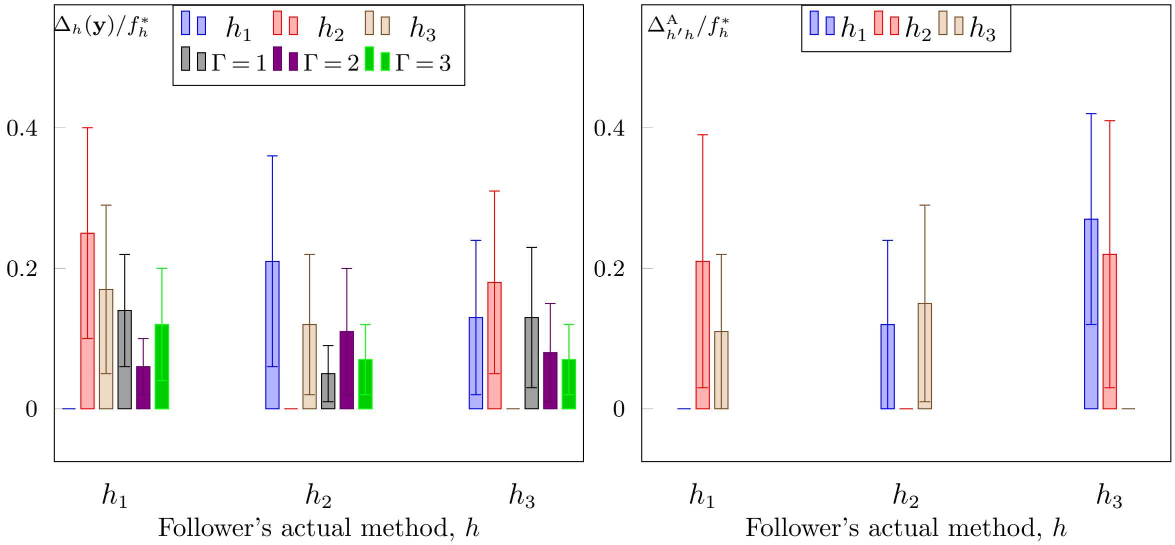


(a) (b) (c)

(d) (e) (f)

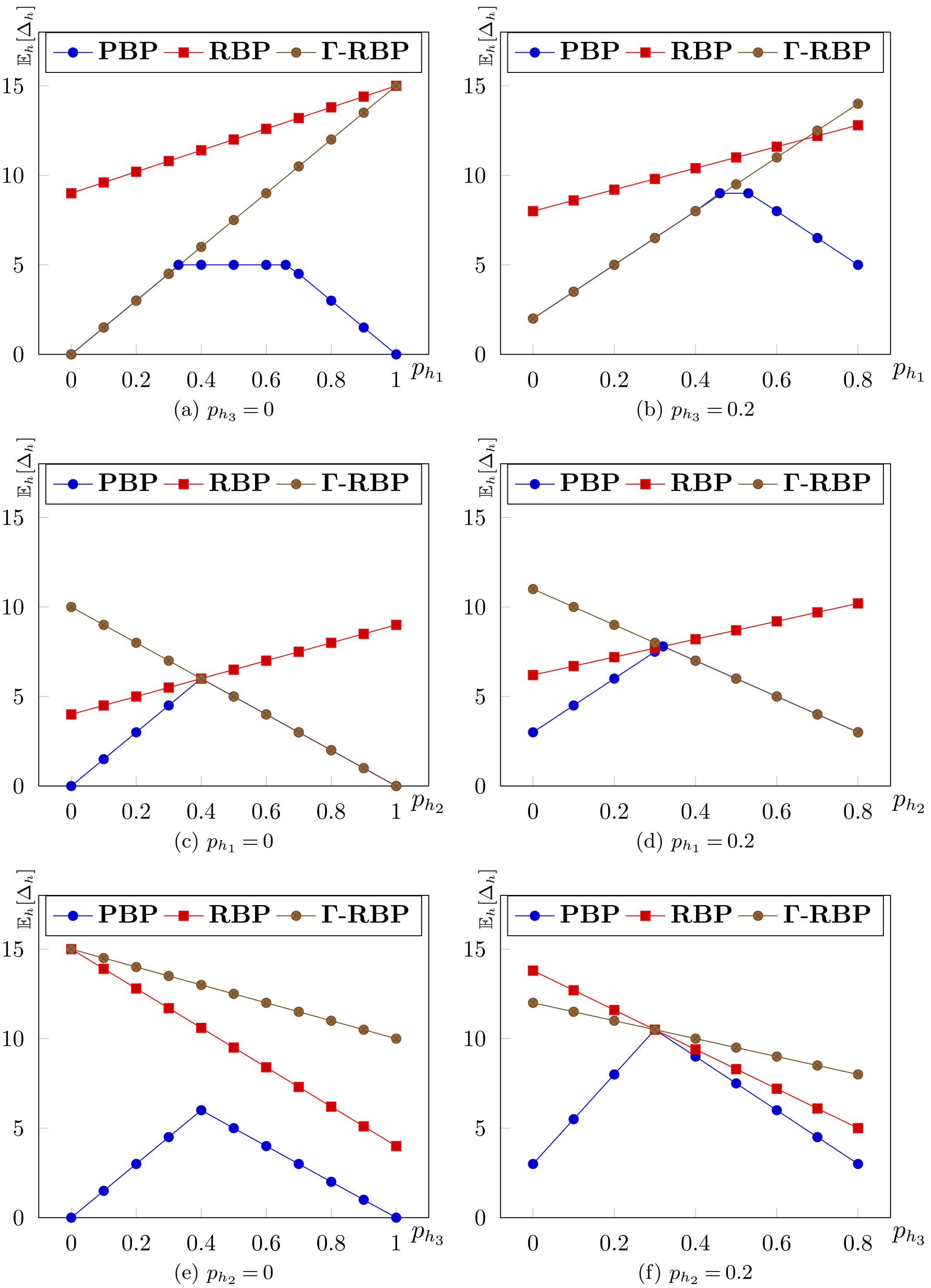
(g) (h) (i)

Figure 1 **Illustration of Proposition 1 and Corollary 1 for three DAP instances where $|I| = 15$ and $|\mathcal{H}| = 3$. The results for the first instance are given in Figures 1(a)-(c). Specifically, the defender implements $\mathbf{y}^*_\Gamma$, $\Gamma \in \{1,2,3\}$, and the attacker responds using methods h_1, h_2 and h_3. Figures 1(a) and 1(b) depict the defender's and the attacker's objective function values, f_1 and f_2, respectively, for each follower's method (thus, three different bars) given leader's decision $\mathbf{y}^*_\Gamma$, $\Gamma \in \{1,2,3\}$. The leader's loss values, $\Delta_h(\mathbf{y}^*_\Gamma)$, are illustrated in Figure 1(c) for $\Gamma \in \{1,2,3\}$ and different methods h. The value of z^*_Γ for $\Gamma \in \{1,2,3\}$ is shown in Figure 1(a). Figures 1(d)-1(f) display the same information for another instance of DAP. The same experiment is performed for thirty DAP instances, all values are scaled down to interval $[0,1]$ and the average value is displayed at Figures 1(g)-1(i). The error bars represent the confidence interval for a significance level of $\alpha = 0.05$.**



(a) (b)

Figure 2 **Illustration of the (defender's) loss ratios, $\Delta_h(\mathbf{y})/f_h^*$ and $\Delta^{\mathrm{A}}_{h'h}/f_h^*$, under different situations. The defender's loss is considered for each possible solution method of the attacker, i.e., h_1, h_2 and h_3. In (a) for each attacker's solution approach, the defender's decision, $\mathbf{y}$, is assumed to be computed based on one of the following six methods (depicted in different colors). In the first three, the defender assumes that the attacker always selects a specific method h and thus, she implements $\mathbf{y}^h$. In the next three, she decides based on the Γ-RBP model, where $\Gamma \in \{1,2,3\}$ and thus, she implements $\mathbf{y}^*_\Gamma$. In (b) we depict the defender's ex-post average loss ratio $\Delta^{\mathrm{A}}_{h'h}/f_h^*$ (relative to her expectations). The results are obtained for thirty DAP instances where $|I| = 15$ and $|\mathcal{H}| = 3$ and the average is reported. The error bars represent the corresponding confidence interval for a significance level of $\alpha = 0.05$.**



(a) $p_{h_3} = 0$

(b) $p_{h_3} = 0.2$

(c) $p_{h_1} = 0$

(d) $p_{h_1} = 0.2$

(e) $p_{h_2} = 0$

(f) $p_{h_2} = 0.2$

Figure 3 **Illustration of the defender's expected loss value as a function of p_h, $h \in \mathcal{H}$ for a DAP instance with $|I| = 15$ and $|\mathcal{H}| = 3$. For Γ-RBP, $\Gamma = 2$. Thus, the expected losses values are displayed for defender's decision $\mathbf{y}_p^*$, $\mathbf{y}_{\Gamma=2}^*$ and $\mathbf{y}_{\Gamma=3}^*$, i.e., when she uses PBP, Γ-RBP and RBP, respectively. In each figure, p_h is fixed for a specific method h and the defender's expected loss values for different approaches are compared.**

Appendix. Proofs

Proof of Proposition 1. Given integer Γ, $1 \leq \Gamma \leq |\mathcal{H}|$, and any leader's feasible decision $\mathbf{y} \in \mathcal{Y}$, define

$$z_\Gamma(\mathbf{y}) := \min_{S,z} \left\{ z : f(\mathbf{y}, \mathbf{x}^h(\mathbf{y})) \leq z \quad \forall h \in S,\, S \subseteq \mathcal{H},\, |S| = \Gamma \right\},$$

and $z^*_\Gamma := \min_{\mathbf{y}\in\mathcal{Y}}\{z_\Gamma(\mathbf{y})\}$. Also, let $\mathbf{y}^*_\Gamma \in \mathcal{Y}$ denote the optimal solution to **Γ-RBP**, so that $z^*_\Gamma = z_\Gamma(\mathbf{y}^*_\Gamma)$. It follows that $z^*_\Gamma \leq z_\Gamma(\mathbf{y}^*_{\Gamma+1})$. Note now that, for every $S \subseteq S'$, one has that

$$\min\left\{ z : f(\mathbf{y}, \mathbf{x}^h(\mathbf{y})) \leq z \quad \forall h \in S \right\} \leq \min\left\{ z : f(\mathbf{y}, \mathbf{x}^h(\mathbf{y})) \leq z \quad \forall h \in S' \right\}.$$

We conclude that $z_\Gamma(\mathbf{y}^*_{\Gamma+1}) \leq z_{\Gamma+1}(\mathbf{y}^*_{\Gamma+1}) = z^*_{\Gamma+1}$. The result follows from combining the above. ■

Proof of Proposition 2. We have that

$$z^*_p = \min_{\mathbf{y}\in\mathcal{Y}} \; \mathbb{E}_{h\in\mathcal{H}}[f(\mathbf{y}, \mathbf{x}^h(\mathbf{y}))] \leq \mathbb{E}_{h\in\mathcal{H}}[f(\mathbf{y}^*_\mathcal{H}, \mathbf{x}^h(\mathbf{y}^*_\mathcal{H}))] \leq \max_h \; f(\mathbf{y}^*_\mathcal{H}, \mathbf{x}^h(\mathbf{y}^*_\mathcal{H})) = z^*_\mathcal{H},$$

which implies the result. ■

Proof of Lemma 1. (⇒) First, consider the case when $f^*_h \leq f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h))$. In this case we have that $f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) - f^*_{h'} \geq f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) - f^*_h \geq 0$, which implies directly that $\Delta^{\mathrm{A}}_{hh'} \leq \Delta_{hh'}$. Suppose now that $f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) \leq f^*_h$. Then $\Delta^{\mathrm{A}}_{hh'} = 0$, and the results follows from the fact that $\Delta_{hh'} \geq 0$.
(⇐) First, consider the case when $\Delta^{\mathrm{A}}_{hh'} > 0$: we have that

$$\Delta^{\mathrm{A}}_{hh'} \leq \Delta_{hh'} \;\Rightarrow\; f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) - f^*_h \leq f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) - f^*_{h'} \;\Rightarrow\; f^*_{h'} \leq f^*_h.$$

Suppose now that $\Delta^{\mathrm{A}}_{hh'} = 0$, then $f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) \leq f^*_h$ and we have that $f^*_{h'} \leq f^*_h$ because of the optimality of $\mathbf{y}^{h'}$.

With regard to the second assertion in the statement of the lemma, we have

$$\Delta^{\mathrm{A}}_{hh'} \geq f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) - f^*_h \geq f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) - f^*_{h'} = \Delta_{hh'},$$

where the first inequality holds by the definition of $\Delta^{\mathrm{A}}_{hh'}$ and the second inequality follows from the assumption that $f^*_h \leq f^*_{h'}$. ■

Proof of Lemma 2. If $\Delta^{\mathrm{A}}_{hh'} = 0$, then $f^*_{h'} \leq f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) \leq f^*_h$ which proves the first part of the lemma. The second part follows directly from this and the definition of $\Delta_{hh'}$. ■

Proof of Proposition 3. Let $\mathbf{y}^*_\mathcal{H}$ denote an optimal solution to **RBP**. We have that

$$z^*_\mathcal{H} = \min_{\mathbf{y}\in\mathcal{Y}} \max\left\{ f(\mathbf{y}, \mathbf{x}^h(\mathbf{y})), f(\mathbf{y}, \mathbf{x}^{h'}(\mathbf{y})) \right\} \leq \max\{f^*_h, f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h))\},$$

where the inequality follows as $\mathbf{y}^h \in \mathcal{Y}$. Recalling that $z^*_\mathcal{H} = \max\{f(\mathbf{y}^*_\mathcal{H}, \mathbf{x}^h(\mathbf{y}^*_\mathcal{H})), f(\mathbf{y}^*_\mathcal{H}, \mathbf{x}^{h'}(\mathbf{y}^*_\mathcal{H}))\}$, we have that

$$f(\mathbf{y}^*_\mathcal{H}, \mathbf{x}^{h'}(\mathbf{y}^*_\mathcal{H})) \leq z^*_\mathcal{H} \leq \max\{f^*_h, f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h))\} = f^*_h + \Delta^{\mathrm{A}}_{hh'},$$

where the last equality holds by the definition of $\Delta^{\mathrm{A}}_{hh'}$. With regard to the second assertion in the statement, we have (from above) that $f(\mathbf{y}^*_{\mathcal{H}}, \mathbf{x}^h(\mathbf{y}^*_{\mathcal{H}})) \leq z^*_{\mathcal{H}}$. Because $\Delta^{\mathrm{A}}_{hh'} = 0$, we obtain $f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) \leq f^*_h$ from the first part of this proof. Hence, we have that

$$f(\mathbf{y}^*_{\mathcal{H}}, \mathbf{x}^h(\mathbf{y}^*_{\mathcal{H}})) \leq z^*_{\mathcal{H}} \leq \max\left\{f^*_h, f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h))\right\} = f^*_h \leq f(\mathbf{y}^*_{\mathcal{H}}, \mathbf{x}^h(\mathbf{y}^*_{\mathcal{H}})),$$

and the result follows. ■

Proof of Proposition 4. Let $\mathbf{y}^*_p$ denote an optimal solution of **PBP**. First, note that both terms $p_h\Big(f(\mathbf{y}^*_p, \mathbf{x}^h(\mathbf{y}^*_p)) - f^*_h\Big)$ and $p_{h'}\Big(f(\mathbf{y}^*_p, \mathbf{x}^{h'}(\mathbf{y}^*_p)) - f^*_{h'}\Big)$ are non–negative. Therefore, we have that

$$p_{h'}\Big(f(\mathbf{y}^*_p, \mathbf{x}^{h'}(\mathbf{y}^*_p)) - f^*_{h'}\Big) \leq p_h\Big(f(\mathbf{y}^*_p, \mathbf{x}^h(\mathbf{y}^*_p)) - f^*_h\Big) + p_{h'}\Big(f(\mathbf{y}^*_p, \mathbf{x}^{h'}(\mathbf{y}^*_p)) - f^*_{h'}\Big).$$

From the optimality of $\mathbf{y}^*_p$ to (8), we have that

$$\begin{aligned} &p_h\Big(f(\mathbf{y}^*_p, \mathbf{x}^h(\mathbf{y}^*_p)) - f^*_h\Big) + p_{h'}\Big(f(\mathbf{y}^*_p, \mathbf{x}^{h'}(\mathbf{y}^*_p)) - f^*_{h'}\Big) \\ &\leq p_h\Big(f(\mathbf{y}^h, \mathbf{x}^h(\mathbf{y}^h)) - f^*_h\Big) + p_{h'}\Big(f(\mathbf{y}^h, \mathbf{x}^{h'}(\mathbf{y}^h)) - f^*_{h'}\Big) = p_{h'}\Delta_{hh'}, \end{aligned}$$

which implies $p_{h'}\Big(f(\mathbf{y}^*_p, \mathbf{x}^{h'}(\mathbf{y}^*_p)) - f^*_{h'}\Big) \leq p_{h'}\Delta_{hh'}$, that is, $f(\mathbf{y}^*_p, \mathbf{x}^{h'}(\mathbf{y}^*_p)) \leq f^*_{h'} + \Delta_{hh'}$. The result follows from exchanging the role of h and h' above and considering the weighted sum. ■

Proof of Proposition 5. Consider the SUBSET SUM problem, which is known to be NP-complete (Garey and Johnson 1979). Given a set of positive integers $S = \{s_1, \ldots, s_n\}$ and a positive integer $k \leq \sum_{i=1}^n s_i$, the SUBSET SUM problem consists of deciding whether or not there exists a subset $\tilde{S} \subseteq S$ such that $\sum_{i \in \tilde{S}} s_i = k$. Consider the following instance of **BKP**:

$$f^* = \min_{\mathbf{y} \in \{0,1\}^n} f(\mathbf{y}, \mathbf{x}^h(y)) = -\sum_{i=1}^n s_i y_i x^h_i(y), \tag{19}$$

where $\mathbf{x}^h(y)$ denotes a solution provided by algorithm h for

$$\max_{\mathbf{x} \in \{0,1\}^n} \Big\{\sum_{i=1}^n s_i y_i x_i : \sum_{i=1}^n s_i x_i \leq k\Big\}.$$

The follower's constraint implies that $f^* \geq -k$. If SUBSET SUM problem has a solution, i.e., there exists subset $\tilde{S} \subseteq S$ such that $\sum_{i \in \tilde{S}} s_i = k$, then the leader's optimal solution is $y_i = 1$ for all $i \in \tilde{S}$ and $y_i = 0$, otherwise. In this case, under Assumption **A4**, the follower's response, based on any algorithm $h \in \mathcal{H}$, is $x^h_i = 1$ if $i \in \tilde{S}$ and $x^h_i = 0$, otherwise.

On the other hand, if an optimal solution of (19) results in $f^* = -k$, then $\tilde{S} = \{s_i : y_i = x_i = 1\}$ corresponds to a "yes" answer of the SUBSET SUM problem. Thus, SUBSET SUM has a solution *iff* $f^* = -k$. ■

Proof of Proposition 6.

First, based on Assumption **A4** and the fact that $\mathbf{x}$ is binary we observe that **SKP** solved in **Step 1** has a finite optimal solution. Similarly, **SKP** with additional linear constraints introduced in **Step 3** has a finite optimal solution in every iteration. Furthermore, because $\check{\mathbf{x}}$ is a binary vector, the number of cuts of the form presented at **Step 3** is finite. Therefore, in order to establish the required result it is sufficient to show that a cut at **Step 3** is never regenerated.

Let $(\hat{\mathbf{y}}^\kappa, \hat{\mathbf{x}}^\kappa)$ be the optimal solution of **SKP** after adding the κ-th cut and $\check{\mathbf{x}}^\kappa$ be the follower's optimal solution for $\mathbf{y} = \hat{\mathbf{y}}^\kappa$. If $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j\big)\check{x}_i = \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j\big)\hat{x}_i$, then Algorithm 1 stops according to **Step 2**. Otherwise, we add $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}y_j\big)x_i \geq \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}y_j\big)\check{x}_i^\kappa$ to **SKP**. Let $(\hat{\mathbf{y}}^{\kappa+1}, \hat{\mathbf{x}}^{\kappa+1})$ be its optimal solution in the next iteration. Then we have two possible situations:

(*i*) If $\hat{\mathbf{y}}^{\kappa+1} = \hat{\mathbf{y}}^\kappa$, then $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^\kappa\big)\hat{x}_i^{\kappa+1} \geq \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^\kappa\big)\check{x}_i^\kappa$. We also know that $\check{\mathbf{x}}^\kappa$ is the follower's optimal solution for $\mathbf{y} = \hat{\mathbf{y}}^\kappa$, i.e., $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^\kappa\big)\hat{x}_i^{\kappa+1} \leq \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^\kappa\big)\check{x}_i^\kappa$. Thus, $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^\kappa\big)\hat{x}_i^{\kappa+1} = \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^\kappa\big)\check{x}_i^\kappa$. That is $(\hat{\mathbf{y}}^{\kappa+1}, \hat{\mathbf{x}}^{\kappa+1})$ is an optimal solution for **BKP** and the algorithm stops.

(*ii*) If $\hat{\mathbf{y}}^{\kappa+1} \neq \hat{\mathbf{y}}^\kappa$, then let $\check{\mathbf{x}}^{\kappa+1}$ be the follower's optimal solution for $\mathbf{y} = \hat{\mathbf{y}}^{\kappa+1}$. We have $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^{\kappa+1}\big)\hat{x}_i^{\kappa+1} \leq \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^{\kappa+1}\big)\check{x}_i^{\kappa+1}$ and $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^{\kappa+1}\big)\hat{x}_i^{\kappa+1} \geq \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^{\kappa+1}\big)\check{x}_i^\kappa$. Thus, if $\check{\mathbf{x}}^{\kappa+1} = \check{\mathbf{x}}^\kappa$, then $\sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^{\kappa+1}\big)\hat{x}_i^{\kappa+1} = \sum_{i=1}^{n_2} \big(c_{i0} + \sum_{j=1}^{n_1} c_{ij}\hat{y}_j^{\kappa+1}\big)\check{x}_i^{\kappa+1}$, which implies that $(\hat{\mathbf{y}}^{\kappa+1}, \hat{\mathbf{x}}^{\kappa+1})$ is an optimal solution of **BKP** and the algorithm stops. Otherwise, $\check{\mathbf{x}}^{\kappa+1} \neq \check{\mathbf{x}}^\kappa$ and the algorithm generates a new cut. ■