Learning the Follower's Objective Function in Sequential Bilevel Games

Ioana Molan, Martin Schmidt, Johannes Thürauf

Abstract. We consider bilevel optimization problems in which the leader has no or only partial knowledge about the objective function of the follower. The studied setting is a sequential one in which the bilevel game is played repeatedly. This allows the leader to learn the objective function (values) of the follower over time. We focus on two methods: a multiplicative weight update (MWU) method and one based on the lower-level's KKT conditions that are used in the sense of inverse optimization. The MWU method requires less assumptions but the convergence guarantee is also only on the follower's objective function values, whereas the inverse KKT method requires stronger assumptions but actually allows to learn objective functions that are consistent with the already observed interactions between the two players. Although the theory we present is only related to the lower-level and not to the upper-level problem, we show that the gained information are practically useful for the leader by illustrating that, over time, the leader's objective function values tend to those that would be obtained under full information. The applicability of the proposed methods is shown using two case studies. First, we study a repeatedly played continuous knapsack interdiction problem and, second, a sequential bilevel pricing game in which the leader needs to learn the utility function of the follower. For both problems, we further illustrate the impact of this learning on the leader's decisions.

1. Introduction

Bilevel optimization is a very active field of research and the interest of the optimization and operations research community significantly increased over the last years and decades. The reason is that these models allow to capture hierarchical decision-making processes. However, bilevel optimization problems are very hard to solve. NP-hardness is shown in Jeroslow (1985) for the easiest instance of a bilevel problem, which is linear and continuous at both levels. Moreover, Hansen et al. (1992) showed that the problem is actually strongly NP-hard. For a more general overview, we refer the interested reader to the books by Dempe (2002) and Dempe et al. (2015) as well as to the recent survey by Kleinert et al. (2021).

Most of the research on bilevel optimization deals with the case that the leader, who acts first in the given hierarchical setting, has complete knowledge about the optimization problem of the follower. This means that she knows the objective function and all the constraints of the follower. In such situations, research mainly focuses on theoretical questions such as existence of solutions or optimality conditions as well as on algorithms for solving these problems. However, in practice, the follower's optimization problem is often not (fully) known by the leader. In this paper, we deal with such situations and develop methods that allow to learn the objective function (values) of the follower. To this end, we consider sequential bilevel problems, i.e., bilevel optimization problems that

Date: September 22, 2025.

2020 *Mathematics Subject Classification.* 90Cxx, 90C90, 91A26, 91A65.

Key words and phrases. Bilevel optimization, Inverse optimization, Unknown follower data, Multiplicative weight update, Inverse KKT method, Sequential games.

are "played" repeatedly. This allows the leader to collect information about the replies of the follower for the given decisions of the leader, which makes it possible to get some insights into the decision criteria (i.e., the objective function) of the follower, which then can finally be used to derive better leader decisions.

Our main contribution are two methods to learn the objective function (values) of the follower. To this end, we restrict ourselves to quadratic and concave lower-level objective functions in the follower's maximization problem. We make this assumption for being able to use single-level reformulations that are based on optimality conditions for the lower-level problem and for being able to use standard solvers such as Gurobi in our case studies. Moreover, we focus on the optimistic variant of the bilevel problem throughout the paper. First, we consider a specialization of the multiplicative weight update (MWU) method (Arora et al. 2012) to learn the objective function values of the follower in a sequential bilevel setting. The method is mainly based on Bärmann et al. (2018, 2017), where the MWU method is used to learn the objective function values of single-level combinatorial optimization problems. We extend their method to the bilevel case by allowing for a combination of known and unknown objective function coefficients that do not all need to be strictly positive, and by considering quadratic objective functions. We also prove asymptotic results that illustrate the convergence properties of the method. Second, we embed the ideas of Keshavarz et al. (2011) in the bilevel setting. In their paper, the authors study how to impute unknown objective functions of convex and parametric optimization problems by means of inverse optimization and the Karush–Kuhn–Tucker (KKT) conditions. This method, called inverse KKT method in what follows, delivers stronger results compared to the MWU method but also requires stronger assumptions. In a nutshell, the inverse KKT method allows for computing a so-called consistent objective function of the follower, which perfectly explains the already observed solutions of the follower in terms of inverse optimization. We discuss this in more detail later when we present the methods. To show the applicability of our approaches, we study two different bilevel problems—first, a repeatedly played continuous knapsack interdiction problem and, second, a repeatedly played bilevel pricing game, in which the leader needs to learn the utility function of the follower. Let us mention here that our theoretical results are all "only" concerned with the lower-level problem. However, the bilevel problem itself is the problem of the leader and the gained information should, of course, be useful for the leader herself. Although we do not have theoretical results regarding the outcome of the leader, we empirically show that the decisions of the leader, over time, tend to the outcome that she would obtain under full information. Further theoretical investigations regarding the leader's outcome over time are out of the scope of this paper and may be hard to obtain due to recent results about the upper-level sensitivity when approximating follower decisions, see, e.g., Beck et al. (2023c).

As already mentioned above, our approaches are strongly connected to the field of inverse optimization, see, e.g., Ahuja and Orlin (2001) for a primer and the references given in the literature overview in Bärmann et al. (2017) for further reading. In addition, we refer to Iraj and Terekhov (2021) for a recent comparison of inverse optimization and machine learning approaches to learn a convex objective function of a single-level optimization problem. Moreover, our MWU approach can be considered as an extension of the recent research on inverse optimization through online learning as it is considered in Bärmann et al. (2018). In addition to the MWU approach, other online learning frameworks, e.g., based on gradient methods, are used in the literature to learn unknown convex objective functions or constraints even with noisy data; see, e.g., Dong et al. (2018) and the references therein. Moreover, we refer to Besbes et al. (2025) for a theoretical analysis of offline (based on inverse optimization) and online learning methods with a focus on minimizing the regret of single-level optimization problems with unknown costs. Finally, let us also mention Tan et al. (2020), who use bilevel optimization to learn linear programs from observed optimal decisions.

Further, the considered setup is related to the field of robust optimization and bilevel optimization under uncertainty since the objective function of the follower is not (fully) known to the leader and can thus be considered as an uncertain parameter of the overall model. For a general overview of the field of bilevel optimization under uncertainty, we refer to the recent articles by Beck et al. (2022, 2023a) and for its connection to robust optimization to Goerigk et al. (2025). This field is rather young and not many papers actually consider the task of learning problem data of the lower level. In particular, although there have been very many applications of bilevel optimization problems for machine learning, see, e.g., Bennett et al. (2008) or Table 1 in Khanduri et al. (2021), the other way around is still in its infancy. Let us further mention the recent paper by Kwon et al. (2024), in which the authors make use of deep learning methods to solve bilevel problems. More exactly, the authors use graph neural networks to heuristically tackle knapsack interdiction problems.

In Molan and Schmidt (2023), the authors consider unknown lower-level problems and propose a learning method based on neural networks for the best-reply function of the follower, which is then introduced as a constraint in the problem of the leader. A similar approach is followed by Vlah et al. (2022), where the authors use convolutional neural networks for tackling bilevel bidding problems arising in power markets. In Fajemisin et al. (2024) the authors provide a survey on constraint learning methods, where the missing constraint is replaced by an estimating black-box model in the optimization problem. Also very recently, Dumouchelle et al. (2024) present a neural-network based heuristic for hard bilevel problems in a data-driven environment.

A rather similar setting compared to ours is studied by Borrero et al. (2022); see also Borrero et al. (2016, 2019) and Yang et al. (2021) for former papers that paved the way for the methods and results in Borrero et al. (2022). However, the techniques used to tackle the problem are rather different to the ones used in this paper as Borrero et al. (2022) use different forms of information feedback to update an uncertainty set that contains the actual follower's objective function. Let us also mention Kwon and Park (2022), where the authors study single-level reformulations of bilevel problems and where the leader's decision is predicted with the help of graph neural networks. Hence, they study a kind of opposite situation in which the uncertainty is in the upper- and not in the lower-level problem. In a recent paper, Li and Han (2023) consider a very related setup of a Stackelberg game in which the follower's objective function is unknown as well. Their approach is based on gradient methods using inexact best responses of the follower for solving the problem. Hence, the studied methods are very different from what we propose. A similar setting is considered by Sessa et al. (2020), but the authors use kernel-based approaches. Finally, related questions have been studied in sequential games as well; see, e.g., Clarke et al. (2023) and the references therein.

The remainder of the paper is structured as follows. In Section 2, we formally define sequential bilevel problems and introduce the necessary notation. Afterward, in Section 3, we introduce and analyze the MWU method, whereas the inverse KKT method is presented in Section 4. The case studies on continuous knapsack interdiction and on bilevel pricing are discussed in Section 5 before we close the paper in Section 6 by summarizing our findings and by sketching some ideas for potential future research directions.

2. Problem Statement

We consider the hierarchical interaction between a leader and a follower that takes place repeatedly over time. This is modeled in a way that, for every point in time $T \in [T^{\max}] := \{1, \dots, T^{\max}\}$, the following bilevel problem is solved:

$$\begin{aligned} \max_{x^T, y^T} \quad & F(x^T, y^T) \\ \text{s.t.} \quad & G(x^T, y^T) \leq 0, \\ & y^T \in \arg\max_y \left\{ y^\top Q y + c^\top y \colon y \in Y(x^T) \right\}. \end{aligned} \tag{$P^T(c, Q)$}$$

Here, $x^T \in \mathbb{R}^m$, $y^T \in \mathbb{R}^n$, $F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $G : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, and $Y(x^T) \subseteq \mathbb{R}^n$ is the feasible set of the follower's problem. Note that we consider the optimistic variant of the bilevel problem and that we use the symbol $\top$ to denote transposition to not confuse with the superscript T representing a certain point in time.

The quadratic lower-level objective function is defined by the matrix $Q \in \mathbb{R}^{n \times n}$ and the vector $c \in \mathbb{R}^n$. If the leader had full information about the follower's problem and if the follower's problem was uniquely solvable, the globally optimal solution would always be the same, i.e., $(x^{T_1}, y^{T_1}) = (x^{T_2}, y^{T_2})$ for all T_1, T_2 since the problem's data does not change over time. However, we consider the situation in which Q and c are assumed to be only partially known to the leader. We denote by $\mathcal{U}^c \subseteq [n]$ and $\mathcal{U}^Q \subseteq [n]^2 := [n] \times [n]$ the index sets of the unknown entries in c and Q, respectively. As a consequence, within every interaction between the two players at time point T, the leader must make a decision x^T while being unaware of the lower-level objective function. Afterward, the leader can observe the corresponding reaction y^T of the follower. However, for her decision making at time point T, the leader can always use the observed data from the past, that is all pairs (x^t, y^t), $t \in [T-1]$. For the timing, we use the following notation throughout the paper: $T^{\max}$ is the final time period of the sequential game. A certain time point is always denoted by $T \in [T^{\max}]$ and all time periods before this specific one are denoted by $t \in [T-1]$.

While Q and c remain partially unknown to the leader, the follower has full knowledge about them. Hence, in round T, the follower's response y^T is a solution of the parametric lower-level problem

$$\max_{y^T} \quad (y^T)^\top Q y^T + c^\top y^T \tag{1a}$$

$$\text{s.t.} \quad y^T \in Y(x^T). \tag{1b}$$

Under the assumption that the feasible set $Y(x^T)$ is known to the leader for all T, we use the set of bilevel solutions (x^t, y^t), $t \in [T-1]$, and present two inverse optimization methods to learn the objective function (value) of the follower.

The overall goal of the leader is to obtain a guess about the lower-level problem so that the outcome $F(x^T, y^T)$ of the bilevel problem is as close as possible to the optimal outcome $F(x^*, y^*)$ that the leader would obtain under full information. More formally, the overall goal of the leader is that the relative measure

$$\frac{\left| F(x^T, y^T) - F(x^*, y^*) \right|}{|F(x^*, y^*)|}$$

is getting smaller for T getting larger, i.e., for increasing number of rounds the repeated bilevel game has been played. In doing so, the leader plays bilevel optimal solutions in each point in time $\{1, \dots, T\}$ and does not strategically play potentially suboptimal ones in earlier time points to obtain a better outcome in round T.

Let us finally comment on why we restrict ourselves here to quadratic lower-level functions. In this case, the parameterization of the lower-level's objective function by a finite number of parameters to be learned is direct because it is the data of the quadratic objective itself: the entries of the matrix Q and the entries of the vector c. As said above, we need to use optimality conditions for the lower-level problem later on, which is possible if the quadratic objective has the right curvature, i.e., if Q is negative-definite because the problem of the follower is a maximization problem. The method we present in the following section can also be extended to general nonlinear but concave functions if they are parameterized by a finite number of parameters. However, for the ease of presentation, we restrict ourselves to quadratic functions.

3. A Multiplicative Weight Update Method

We now present a method to learn optimal objective function values of the follower based on the multiplicative weight update (MWU) algorithm (Arora et al. 2012). To this end, at the time point $T \in [T^{\max}]$, we consider a quadratic lower-level objective function (1a), which is not or only partially known to the leader, and we use past observations (x^t, y^t), $t \in [T-1]$, from leader-follower interactions to iteratively update the leader's guess of the unknown follower's objective function.

Our analysis follows the one by Bärmann et al. (2018, 2017), who use an MWU algorithm to learn objective function values of single-level problems with linear objective functions. We now extend this method (i) to quadratic functions, (ii) to only partially instead of completely unknown objective functions, as well as (iii) to the case in which we allow for negative coefficients. In order to do so, we need the following assumptions.

Assumption 1. *It holds*

$$\sum_{i \in \mathcal{U}^c} |c_i| + \sum_{(i,j) \in \mathcal{U}^Q} |Q_{ij}| = 1.$$

This assumption is mild in the sense that it only excludes the zero objective since all other quadratic objective functions can be scaled so that the unknown coefficients satisfy the assumption.

For the next assumption, for a matrix $A \in \mathbb{R}^{n \times m}$, we use the matrix norm

$$\|A\|_{\max} := \max_{i \in [n], j \in [m]} |A_{ij}|.$$

Moreover, we denote with

$$\Omega := \{x \in \mathbb{R}^m : \exists y \in Y(x) \text{ with } G(x, y) \le 0\}$$

a set of upper-level decisions that forms a superset of all feasible upper-level bilevel feasible points. We note that the set of feasible upper-level points of $(P^T(c, Q))$ may change if the lower-level objective is modified—in particular due to the presence of coupling constraints. However, the set Ω is always a superset of all feasible upper-level feasible points independent from the lower-level objective coefficients. Since we change the lower-level objective in the following algorithm, it is necessary to consider Ω instead of all feasible upper-level decision.

Assumption 2. *There exists a constant $K \ge 0$ such that*

$$\max_{y_1, y_2 \in Y(x)} \left\{ \|y_1 - y_2\|_\infty , \left\|y_1 {y_1}^\top - y_2 {y_2}^\top\right\|_{\max} \right\} \le K$$

holds for all upper-level decisions $x \in \Omega$.

Algorithm 1 A Multiplicative Weight Update Method

Input: Index sets $\mathcal{U}^c$,$\mathcal{U}^Q$, $\mathcal{P}^c$, $\mathcal{N}^c$, $\mathcal{P}^Q$, $\mathcal{N}^Q$, and known coefficients $c_i, i \in [n] \setminus \mathcal{U}^c$, Q_{ij}, $(i,j) \in [n]^2 \setminus \mathcal{U}^Q$.
Output: A sequence of objectives $(\bar{c}^1, \bar{Q}^1), (\bar{c}^2, \bar{Q}^2), \dots$
1: Set $T \leftarrow 1$.
2: Set $w^T_{c_i} \leftarrow 1$ for all $i \in \mathcal{U}^c$ and $w^T_{Q_{ij}} \leftarrow 1$ for all $(i,j) \in \mathcal{U}^Q$.
3: **for** $T = 1, 2, \dots$ **do**
4: Set $f \leftarrow \sum_{i \in \mathcal{U}^c} |w^T_{c_i}| + \sum_{(i,j) \in \mathcal{U}^Q} |w^T_{Q_{ij}}|$.
5: Set $c^T_i \leftarrow w^T_{c_i}/f$ for all $i \in \mathcal{U}^c$ and $Q^T_{ij} \leftarrow w^T_{Q_{ij}}/f$ for all $(i,j) \in \mathcal{U}^Q$.
6: Set $\bar{c}^T_i \leftarrow -c^T_i$ for all $i \in \mathcal{U}^c \cap \mathcal{N}^c$ and $\bar{c}^T_i \leftarrow c^T_i$ for all $i \in \mathcal{U}^c \cap \mathcal{P}^c$.
7: Set $\bar{Q}^T_{ij} \leftarrow -Q^T_{ij}$ for all $(i,j) \in \mathcal{U}^Q \cap \mathcal{N}^Q$ and $\bar{Q}^T_{ij} \leftarrow Q^T_{ij}$ for all $(i,j) \in \mathcal{U}^Q \cap \mathcal{P}^Q$.
8: Compute the bilevel solution $(x^T, \bar{y}^T)$ by solving the bilevel problem $P^T(\bar{c}^T, \bar{Q}^T)$.
9: Play x^T and observe the corresponding (true) follower's solution y^T.
10: Set $d \leftarrow \max \left\{ \max_{i \in \mathcal{U}^c} \left\{ \left| \bar{y}^T_i - y^T_i \right| \right\}, \max_{(i,j) \in \mathcal{U}^Q} \left\{ \left| \bar{y}^T_i \bar{y}^T_j - y^T_i y^T_j \right| \right\} \right\}$.
11: **if** $d = 0$ **then**
12: Set $e^T_{c_i} \leftarrow 0$ for all $i \in \mathcal{U}^c$ and $e^T_{Q_{ij}} \leftarrow 0$ for all $(i,j) \in \mathcal{U}^Q$.
13: **else**
14: Set $e^T_{c_i} \leftarrow (\bar{y}^T_i - y^T_i)/d$ for all $i \in \mathcal{P}^c \cap \mathcal{U}^c$.
15: Set $e^T_{c_i} \leftarrow (y^T_i - \bar{y}^T_i)/d$ for all $i \in \mathcal{N}^c \cap \mathcal{U}^c$.
16: Set $e^T_{Q_{ij}} \leftarrow (\bar{y}^T_i \bar{y}^T_j - y^T_i y^T_j)/d$ for all $(i,j) \in \mathcal{P}^Q \cap \mathcal{U}^Q$.
17: Set $e^T_{Q_{ij}} \leftarrow (y^T_i y^T_j - \bar{y}^T_i \bar{y}^T_j)/d$ for all $(i,j) \in \mathcal{N}^Q \cap \mathcal{U}^Q$.
18: **end if**
19: Set $\eta^T \in (0, 1/2]$.
20: Set $w^{T+1}_{c_i} \leftarrow w^T_{c_i}(1 - \eta^T e^T_{c_i})$ for all $i \in \mathcal{U}^c$.
21: Set $w^{T+1}_{Q_{ij}} \leftarrow w^T_{Q_{ij}}(1 - \eta^T e^T_{Q_{ij}})$ for all $(i,j) \in \mathcal{U}^Q$.
22: Set $T \leftarrow T + 1$.
23: **end for**
24: **return** $(\bar{c}^1, \bar{Q}^1), (\bar{c}^2, \bar{Q}^2), \dots$

Note that for Assumption 2 to hold it is sufficient that for each feasible upper-level decision x, the feasible set $Y(x)$ of the follower's problem is bounded. We finally assume that we can partition the coefficients of the objective function into sets of negative and nonnegative coefficients.

Assumption 3. *The index sets*

$$\mathcal{N}^c \coloneqq \{i \in [n] \colon c_i < 0\}, \quad \mathcal{P}^c \coloneqq \{i \in [n] \colon c_i \geq 0\}$$

and

$$\mathcal{N}^Q \coloneqq \left\{(i,j) \in [n]^2 \colon Q_{ij} < 0\right\}, \quad \mathcal{P}^Q \coloneqq \left\{(i,j) \in [n]^2 \colon Q_{ij} \geq 0\right\}$$

are known.

It depends on the specific problem at hand if the latter assumption is strong. We will discuss this again when considering the specific case studies in Section 5.

By adapting the MWU method, we now present an iterative method, see Algorithm 1, that is capable of learning the objective function values of the follower's problem over time. We now explain the main steps of the method in detail. Note that each iteration of the algorithm corresponds to one round in the sequential bilevel setting. In every iteration T, the algorithm first updates the unknown objective coefficients in c^T and Q^T, while the known ones remain the same; see Lines 4 and 5. Note that the update in Line 5 assigns every unknown coefficient c_i^T, $i \in \mathcal{U}^c$, and Q_{ij}^T, $(i,j) \in \mathcal{U}^Q$, a positive value. Thus, we use the index sets $\mathcal{N}^c$ and $\mathcal{N}^Q$ to re-introduce the missing negative signs into the objective function in Lines 6 and 7. Afterward, the bilevel problem $P^T(\bar{c}^T, \bar{Q}^T)$ is solved in Line 8 using the computed coefficients for the follower's objective function.

As a final step of each iteration, we compute weights in Lines 20 and 21, which are then used to update the guess of the unknown coefficients at the beginning of the next iteration. These weights are based on the difference between the anticipated solution $\bar{y}^T$ and the actual solution y^T of the follower's problem, that can be observed by the leader in iteration T. Thus, from a bilevel perspective, the update of the weights exploits the difference between the solution that the leader obtains by guessing the unknown objective function and the actual decision of the follower.

Rather than producing one objective function, the presented method yields a sequence of quadratic objective functions parameterized by $(\bar{c}^T, \bar{Q}^T)$. These learned objective functions satisfy the following guarantees for the objective function values.

Theorem 1. *Let a maximum iteration $T^{\max}$ be given and suppose that Assumptions 1–3 hold and let $\eta^T \leq 1/2$ for all $T \in [T^{\max}]$. Then, the output of Algorithm 1 satisfies*

$$\begin{aligned} 0 &\leq \frac{1}{T^{\max}} \sum_{T=1}^{T^{\max}} \eta^T \left(\sum_{i\in[n]} (\bar{c}_i^T - c_i)(\bar{y}_i^T - y_i^T) + \sum_{(i,j)\in[n]^2} (\bar{Q}_{ij}^T - Q_{ij})(\bar{y}_i^T \bar{y}_j^T - y_i^T y_j^T) \right) \\ &\leq \frac{K}{T^{\max}} \left(\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|) + \sum_{T=1}^{T^{\max}} (\eta^T)^2 \right). \end{aligned} \tag{2}$$

Proof. In order to prove the theorem, we apply Corollary 3 (see Appendix A), which is based on Corollary 2.2 in Arora et al. (2012), to the values (c^T, Q^T), $T \in [T^{\max}]$, computed by Algorithm 1. Note that this is the output of the algorithm but without the respective sign corrections. To apply the general theory, we have to show that all assumptions are fulfilled.

First, the errors computed in Lines 14 and 17 of the algorithm both satisfy $e_{c_i}^T, e_{Q_{ij}}^T \in [-1,1]$ for all $i \in \mathcal{U}^c$ and $(i,j) \in \mathcal{U}^Q$ for all T. For $T \in [T^{\max}]$, let c_i^T, $i \in [n]$, and $Q_{i,j}^T$, $(i,j) \in [n]^2$, be the coefficients computed in Algorithm 1 before re-introducing the negative sign in Lines 6 and 7. The corresponding unknown coefficients are nonnegative and they sum up to one due to Lines 4, 5 and 20, 21. In these lines we see that the weights never get negative and that the used coefficients are divided by the positive sum of their absolute values. Thus, the computed coefficients c_i^T, Q_{ij}^T for $i \in \mathcal{U}^c$, $(i,j) \in \mathcal{U}^Q$, and $T \in [T^{\max}]$ can be interpreted as a probability distribution.

Furthermore, due to Assumption 1, the sum of the absolute true but unknown coefficients adds up to one. Nevertheless, the coefficients c_i with $i \in \mathcal{U}^c \cap \mathcal{N}^c$ and Q_{ij} with $(i,j) \in \mathcal{U}^Q \cap \mathcal{N}^Q$ are in fact negative. In order to be able to re-interpret the probability distribution used in the Corollary 2.2 in Arora et al. (2012) as the true unknown coefficients, we consider for now the absolute values $|c_i|$ for $i \in \mathcal{U}^c$ and $|Q_{ij}|$ for $(i,j) \in \mathcal{U}^Q$ of the true coefficients.

We now apply Corollary 3 to the nonnegative output (c^T, Q^T), $T \in [T^{\max}]$, of our algorithm and obtain

$$\begin{aligned}
&\sum_{T=1}^{T^{\max}} \eta^T \left(\sum_{i \in \mathcal{U}^c} c_i^T e_{c_i}^T + \sum_{(i,j) \in \mathcal{U}^Q} Q_{ij}^T e_{Q_{ij}}^T \right) \\
\leq\ & \ln(|\mathcal{U}^c| + |\mathcal{U}^Q|) + \sum_{T=1}^{T^{\max}} \eta^T \left(\sum_{i \in \mathcal{U}^c} |c_i| (e_{c_i}^T + \eta^T |e_{c_i}^T|) + \sum_{(i,j) \in \mathcal{U}^Q} |Q_{ij}| (e_{Q_{ij}}^T + \eta^T |e_{Q_{ij}}^T|) \right).
\end{aligned}$$

We use $|e_{c_i}^T| \leq 1$ for all $i \in \mathcal{U}^c$ and $|e_{Q_{ij}}^T| \leq 1$ for all $(i,j) \in \mathcal{U}^Q$, divide by $T^{\max}$, and obtain

$$\begin{aligned}
&\frac{1}{T^{\max}} \sum_{T=1}^{T^{\max}} \eta^T \left(\sum_{i \in \mathcal{U}^c} c_i^T e_{c_i}^T + \sum_{(i,j) \in \mathcal{U}^Q} Q_{ij}^T e_{Q_{ij}}^T \right) \\
&\quad - \frac{1}{T^{\max}} \sum_{T=1}^{T^{\max}} \eta^T \left(\sum_{i \in \mathcal{U}^c} |c_i| e_{c_i}^T + \sum_{(i,j) \in \mathcal{U}^Q} |Q_{ij}| e_{Q_{ij}}^T \right) \\
\leq\ & \frac{1}{T^{\max}} \ln(|\mathcal{U}^c| + |\mathcal{U}^Q|) + \frac{1}{T^{\max}} \sum_{T=1}^{T^{\max}} (\eta^T)^2 \left(\sum_{i \in \mathcal{U}^c} |c_i| + \sum_{(i,j) \in \mathcal{U}^Q} |Q_{ij}| \right) \\
=\ & \frac{1}{T^{\max}} \left(\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|) + \sum_{T=1}^{T^{\max}} (\eta^T)^2 \right),
\end{aligned}$$

where the last equality above follows from Assumption 1. We use the definitions of $e_{c_i}^T$ and $e_{Q_{ij}}^T$, multiply by the constant K, and obtain

$$\begin{aligned}
&\frac{1}{T^{\max}} \sum_{T=1}^{T^{\max}} \eta^T \Bigg(\sum_{i \in \mathcal{U}^c \cap \mathcal{P}^c} (c_i^T - |c_i|)(\bar{y}_i^T - y_i^T) + \sum_{i \in \mathcal{U}^c \cap \mathcal{N}^c} (c_i^T - |c_i|)(y_i^T - \bar{y}_i^T) \\
&+ \sum_{(i,j) \in \mathcal{U}^Q \cap \mathcal{P}^Q} (Q_{ij}^T - |Q_{ij}|)(\bar{y}_i^T \bar{y}_j^T - y_i^T y_j^T) \\
&+ \sum_{(i,j) \in \mathcal{U}^Q \cap \mathcal{N}^Q} (Q_{ij}^T - |Q_{ij}|)(y_i^T y_j^T - \bar{y}_i^T \bar{y}_j^T) \Bigg) \\
\leq\ & K \frac{1}{T^{\max}} \left(\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|) + \sum_{T=1}^{T^{\max}} (\eta^T)^2 \right).
\end{aligned} \tag{3}$$

In our method, the known entries in c and Q remain unchanged from iteration to iteration. Thus we can include them in the left-hand side of Expression (3) leading to

$$\frac{1}{T^{\max}}\sum_{T=1}^{T^{\max}}\eta^T\left(\sum_{i\in\mathcal{U}^c\cap\mathcal{P}^c}(c_i^T-|c_i|)(\bar{y}_i^T-y_i^T)+\sum_{i\in\mathcal{U}^c\cap\mathcal{N}^c}(c_i^T-|c_i|)(y_i^T-\bar{y}_i^T)\right. \tag{4a}$$

$$+\sum_{i\in\mathcal{P}^c\cap([n]\setminus\mathcal{U}^c)}(c_i-c_i)(\bar{y}_i^T-y_i^T)+\sum_{i\in\mathcal{N}^c\cap([n]\setminus\mathcal{U}^c)}(|c_i|-|c_i|)(y_i^T-\bar{y}_i^T) \tag{4b}$$

$$+\sum_{(i,j)\in\mathcal{U}^Q\cap\mathcal{P}^Q}(Q_{ij}^T-|Q_{ij}|)(\bar{y}_i^T\bar{y}_j^T-y_i^Ty_j^T) \tag{4c}$$

$$+\sum_{(i,j)\in\mathcal{U}^Q\cap\mathcal{N}^Q}(Q_{ij}^T-|Q_{ij}|)(y_i^Ty_j^T-\bar{y}_i^T\bar{y}_j^T) \tag{4d}$$

$$+\sum_{(i,j)\in\mathcal{P}^Q\cap([n]^2\setminus\mathcal{U}^Q)}(Q_{ij}-Q_{ij})(\bar{y}_i^T\bar{y}_j^T-y_i^Ty_j^T) \tag{4e}$$

$$\left.+\sum_{(i,j)\in\mathcal{N}^Q\cap([n]^2\setminus\mathcal{U}^Q)}(|Q_{ij}|-|Q_{ij}|)(y_i^Ty_j^T-\bar{y}_i^T\bar{y}_j^T)\right). \tag{4f}$$

The newly added terms in Lines (4b), (4e), and (4f) equal zero. Thus, the expression above is equal to

$$\frac{1}{T^{\max}}\sum_{T=1}^{T^{\max}}\eta^T\left(\sum_{i\in\mathcal{P}^c}(|c_i^T|-|c_i|)(\bar{y}_i^T-y_i^T)+\sum_{i\in\mathcal{N}^c}(|c_i^T|-|c_i|)(y_i^T-\bar{y}_i^T)\right.$$
$$\left.+\sum_{(i,j)\in\mathcal{P}^Q}(|Q_{ij}^T|-|Q_{ij}|)(\bar{y}_i^T\bar{y}_j^T-y_i^Ty_j^T)+\sum_{(i,j)\in\mathcal{N}^Q}(|Q_{ij}^T|-|Q_{ij}|)(y_i^Ty_j^T-\bar{y}_i^T\bar{y}_j^T)\right).$$

In addition, the nonnegativity of each term follows from the optimality of $\bar{y}^T$ w.r.t. $\bar{c}^T$ and $\bar{Q}^T$, respectively of y^T w.r.t. c and Q. Using the variable mapping of Lines 6 and 7, we obtain the desired result. □

From the latter theorem, we immediately obtain the following corollary that we state for completeness although we do not directly exploit it later on.

Corollary 1. *Under the requirements of Theorem 1, the inequalities*

$$0\le\frac{1}{T^{\max}}\sum_{T=1}^{T^{\max}}\eta^T\left(\sum_{i\in[n]}\bar{c}_i^T(\bar{y}_i^T-y_i^T)+\sum_{(i,j)\in[n]^2}\bar{Q}_{ij}^T(\bar{y}_i^T\bar{y}_j^T-y_i^Ty_j^T)\right)$$
$$\le\frac{K}{T^{\max}}\left(\ln(|\mathcal{U}^c|+|\mathcal{U}^Q|)+\sum_{T=1}^{T^{\max}}(\eta^T)^2\right)$$

and

$$0\le\frac{1}{T^{\max}}\sum_{T=1}^{T^{\max}}\eta^T\left(\sum_{i\in[n]}c_i(y_i^T-\bar{y}_i^T)+\sum_{(i,j)\in[n]^2}Q_{ij}(y_i^Ty_j^T-\bar{y}_i^T\bar{y}_j^T)\right)$$
$$\le\frac{K}{T^{\max}}\left(\ln(|\mathcal{U}^c|+|\mathcal{U}^Q|)+\sum_{T=1}^{T^{\max}}(\eta^T)^2\right)$$

hold.

Theorem 1 provides bounds for the MWU error. However, for specific learning rates η^T, we can show that the upper bound on the error converges to zero for increasing number of iterations.

Corollary 2. *Let $T^{\max}$ be given, let the assumptions of Theorem 1 hold, and choose $\eta^T := \gamma/T$ with $\gamma \in (0, 1/2]$ and $T \in [T^{\max}]$. Then, it holds*

$$0 \leq \frac{1}{T^{\max}} \sum_{T=1}^{T^{\max}} \frac{\gamma}{T} \left(\sum_{i \in [n]} (\bar{c}_i^T - c_i)(\bar{y}_i^T - y_i^T) + \sum_{(i,j) \in [n]^2} (\bar{Q}_{ij}^T - Q_{ij})(\bar{y}_i^T \bar{y}_j^T - y_i^T y_j^T) \right)$$

$$\leq \frac{K}{T^{\max}} \left(\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|) + \gamma^2 \sum_{T=1}^{T^{\max}} \left(\frac{1}{T}\right)^2 \right) \leq \frac{K}{T^{\max}} \left(\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|) + \gamma^2 \frac{\pi^2}{6} \right),$$

which implies that

$$\frac{1}{T^{\max}} \sum_{T=1}^{T^{\max}} \frac{\gamma}{T} \left(\sum_{i \in [n]} (\bar{c}_i^T - c_i)(\bar{y}_i^T - y_i^T) + \sum_{(i,j) \in [n]^2} (\bar{Q}_{ij}^T - Q_{ij})(\bar{y}_i^T \bar{y}_j^T - y_i^T y_j^T) \right)$$

tends to 0 for $T^{\max}$ tending to infinity.

4. An Inverse KKT Method

In this section, we focus on learning a so-called consistent objective function of the follower's problem instead of the corresponding objective function value as in the previous section. Since learning the objective function itself is even more demanding than learning the objective function value, we have to impose stronger assumptions compared to the presented MWU approach. Thus, in addition to Assumption 3, we assume that the follower's objective function is concave-quadratic and that the x-parameterized feasible set is polyhedral, i.e., the lower-level problem is given by

$$\max_y \quad y^\top Q y + c^\top y \quad \text{s.t.} \quad A(x)y \geq b(x) \tag{5}$$

with $b(x) \in \mathbb{R}^m$ and $A(x) \in \mathbb{R}^{m \times n}$ both depending on the leader's decision x. Further, $Q \in \mathbb{R}^{n \times n}$ is, w.l.o.g., symmetric and negative semidefinite by the assumption that the objective function is concave. Moreover, being at time point T, let (x^t, y^t), $t \in [T-1]$, be all past pairs of observations consisting of leader decisions x^t and respective optimal follower replies y^t.

We now iteratively learn a consistent objective function of the follower's problem by adapting an approach for convex single-level problems by Keshavarz et al. (2011) to the considered sequential bilevel setting. As in Keshavarz et al. (2011), we say that an objective function is consistent if for a given x^t, the corresponding y^t is optimal for Problem (5) for all t in a considered data set. Note that by this property alone, the objective function does not need to be uniquely determined.

In the following, we describe the approach in detail. Since the feasible set of the follower's problem (5) is polyhedral and due to the concavity of the objective function, the Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient without any further constraint qualifications. For fixed x^t and y^t, these KKT conditions are given by

$$2Qy^t + c + A(x^t)^\top \lambda^t = 0, \tag{6a}$$

$$A(x^t)y^t - b(x^t) \geq 0, \tag{6b}$$

$$(\lambda^t)^\top (A(x^t)y^t - b(x^t)) = 0, \tag{6c}$$

$$\lambda^t \geq 0. \tag{6d}$$

If $y^t \in \mathbb{R}^n$ is optimal for Problem (5) given a leader's decision x^t, there exist Lagrange multipliers $\lambda^t \in \mathbb{R}^m$ such that the KKT conditions (6) are satisfied. In line with Keshavarz et al. (2011), we define a solution $y^t \in \mathbb{R}^n$ as approximately optimal for Problem (5) for given x^t, if (6) is satisfied approximately, i.e., for y^t there exists λ^t such that the residuals

$$r^t_{\text{stat}}(\lambda) \coloneqq 2Qy^t + c + A(x^t)^\top \lambda^t, \tag{7a}$$

$$r^t_{\text{ineq}} \coloneqq \left[(A(x^t)y^t - b(x^t))\right]^-, \tag{7b}$$

$$r^t_{\text{comp}}(\lambda) \coloneqq (\lambda^t)^\top (A(x^t)y^t - b(x^t)), \tag{7c}$$

$$r^t_{\text{pos}}(\lambda) \coloneqq \left[(\lambda^t)\right]^-. \tag{7d}$$

are close to zero. Here, the operator $[\alpha]^- := \max\{0, -\alpha\}$ in (7b) and (7d) is to be understood component-wise.

We now minimize these residuals subject to additional constraints to compute a consistent objective function. Since for each observation (x^t, y^t), the decisions y^t are optimal for the given x^t, the inequality $A(x^t)y^t \geq b(x^t)$ is satisfied. Consequently, $r^t_{\text{ineq}} = 0$ holds and we do not have to explicitly minimize this residual in the following. Moreover, the nonnegativity of the Lagrange multipliers can be easily ensured by bound constraints, which is why $r^t_{\text{pos}}(\lambda) = 0$ can always be achieved by imposing the respective constraints. At time T, the problem to compute consistent objective function data c^T, Q^T for the observed data (x^t, y^t), $t \in [T-1]$, is given by

$$\min_{c^T, Q^T, \lambda} \quad \sum_{t \in [T-1]} \|r^t_{\text{stat}}\|_2^2 + \sum_{t \in [T-1]} \|r^t_{\text{comp}}\|_2^2 \tag{8a}$$

$$\text{s.t.} \quad \lambda^t \geq 0, \quad t \in [T-1], \tag{8b}$$

$$Q^T_{ij} = Q_{ij}, \quad (i,j) \in [n]^2 \setminus \mathcal{U}^Q, \tag{8c}$$

$$Q^T_{ij} \geq 0, \quad (i,j) \in \mathcal{U}^Q \cap \mathcal{P}^Q, \tag{8d}$$

$$Q^T_{ij} \leq 0, \quad (i,j) \in \mathcal{U}^Q \cap \mathcal{N}^Q, \tag{8e}$$

$$c^T_i = c_i, \quad i \in [n] \setminus \mathcal{U}^c, \tag{8f}$$

$$c^T_i \geq 0, \quad i \in \mathcal{U}^c \cap \mathcal{P}^c, \tag{8g}$$

$$c^T_i \leq 0, \quad i \in \mathcal{U}^c \cap \mathcal{N}^c, \tag{8h}$$

$$Q^T \preceq 0, \tag{8i}$$

Problem (8) minimizes the sum of the squared residuals r^t_{stat} and r^t_{comp}. The residual r^t_{stat} is linear in Q_{ij}, $(i,j) \in [n]^2$, and in c_i, $i \in [n]$, and r^t_{comp} is linear in the variables λ^t_i, $i \in [m]$, $t \in [T-1]$. Hence, the objective function is a convex-quadratic function. Constraints (8b) ensure that the Lagrange multipliers are nonnegative. By constraints (8c)–(8h), we incorporate the a priori given knowledge about the objective coefficients: We fix the known objective coefficients and impose the a priori known sign of the unknown coefficients. By the semidefinite-programming constraint (8i), we ensure that the computed objective function is negative semidefinite like the original objective function of the follower's problem (5). Overall, Problem (8) is a quadratic program with a single additional semidefinite constraint.

Furthermore, the complementarity condition (6c) allows us to further tighten the feasible set of Problem (8). As discussed, the primal constraint $A(x^t)y^t \geq b(x^t)$ is satisfied for all samples (x^t, y^t), $t \in [T-1]$, and the value of the slack of this constraint is known

Algorithm 2 Inverse KKT Method

Input: An initial point (x^0, y^0), index sets $\mathcal{U}^c$,$\mathcal{U}^Q$, $\mathcal{P}^c$, $\mathcal{N}^c$, $\mathcal{P}^Q$, $\mathcal{N}^Q$, and all known coefficients $c_i, i \in [n] \setminus \mathcal{U}^c$ and Q_{ij}, $(i,j) \in [n]^2 \setminus \mathcal{U}^Q$.
Output: A sequence of objectives $(\bar{c}^1, \bar{Q}^1), (\bar{c}^2, \bar{Q}^2), \ldots$

```
1: Initialize the data set of observed points D^0 = {(x^0, y^0)}.
2: for T = 1, 2, ... do
3:     Solve the Problem (9) using D^{T-1} and obtain (c̄^T, Q̄^T).
4:     Solve the bilevel problem P^T(c̄^T, Q̄^T) and obtain (x^T, ȳ^T).
5:     Play x^T and observe the corresponding, true follower's solution y^T.
6:     Update D^T ← D^{T-1} ∪ {(x^T, y^T)}.
7: end for
8: return (c̄^1, Q̄^1), (c̄^2, Q̄^2), ...
```

a priori. Using this knowledge we define the index sets of active and inactive constraints

$$I_0^t \coloneqq \left\{\ell \in [m] \colon (A(x^t)y^t - b(x^t))_\ell = 0\right\},$$
$$I_+^t \coloneqq \left\{\ell \in [m] \colon (A(x^t)y^t - b(x^t))_\ell > 0\right\}.$$

Thus, $\lambda_\ell^t = 0$ holds for all $\ell \in I_+^t$ and $\lambda_\ell^t \geq 0$ holds for all $\ell \in I_0^t$ and all $t \in [T-1]$. Introducing these equalities and inequalities as constraints, we obtain $r_{\text{comp}}^t = 0$ for all $t \in [T-1]$. Consequently, we can equivalently reformulate Problem (8) as

$$\min_{c^T, Q^T, \lambda} \quad \sum_{t \in [T-1]} \|r_{\text{stat}}^t\|_2^2 \tag{9a}$$
$$\text{s.t.} \quad \lambda_\ell^t = 0, \quad \ell \in I_+^t,\ t \in [T-1], \tag{9b}$$
$$\lambda_\ell^t \geq 0, \quad \ell \in I_0^t,\ t \in [T-1], \tag{9c}$$
$$\text{(8c)–(8i)}. \tag{9d}$$

The iterative method using this inverse KKT approach is outlined in the Algorithm 2. Note that we initialize the method with a single given data point of leader-follower interactions do avoid a first "dummy" iteration. We finally discuss the presented approach in the light of the considered sequential bilevel setting. The inverse KKT method enables the leader to compute a consistent objective function for the unknown objective of the follower's problem. More precisely, in every round T, the leader can apply the inverse KKT method to the previously observed optimal decisions $\{y^0, y^1, \ldots, y^{t-1}\}$ of the follower in order to obtain a consistent objective function. For this reason, the Model (9) is built and solved repeatedly for a data set $\{(x^t, y^t) \colon t \in \{0\} \cup [T-1]\}$, which increases by a single pair in every iteration.

Remark 2. *For the special case that the matrix Q of the follower's problem (5) is diagonal, we can replace the semidefinite-programming constraint (8i) by the linear constraints $Q_{ii} \leq 0$ for $i \in [n]$. These constraints together with setting the non-diagonal entries of Q to 0 ensures that the computed objective function is concave.*

Remark 3. *A nice feature of the inverse KKT approach is that Model (9) can be extended by any further problem specific knowledge. Let us give two examples for this. First, it might be possible that the leader does not only observe the follower's responses but also the respective objective function values f^t, $t \in [T-1]$, of the follower. If this is the case, one could, e.g., add the constraint*

$$(y^t)^\top Q^T y^t + (y^t)^\top c^T + s^t = f^t, \quad t \in [T-1],$$

where $s^t \geq 0$, $t \in [T-1]$, are slack variables that are minimized by an ℓ_1-penalty term $\sum_{t\in[T-1]} s^t$ in an extended objective function of (9).

Second, if Q is a diagonal matrix and if y^+ is an upper bound on the lower-level variables, the constraint

$$2Q^T y^+ + c^T \geq 0$$

can be added to ensure a component-wise monotonicity of the objective function; see, e.g., Keshavarz et al. (2011), where this has been used as well.

In our numerical experiments, we use both additional constraints as discussed in this remark.

5. Case Studies

In this section, we apply the presented learning methods to two different applications, in which the follower's quadratic objective function is only partially known to the leader. In Section 5.1, we consider a continuous knapsack interdiction problem. In general, the class of interdiction problems is of special interest in the context of learning the objective function (values) of the follower since usually the leader, often referred to as attacker or defender, does not know the exact objective function of the adversarial player. Further, due to the entirely opposing objectives of the players in interdiction problems, the follower has no incentive to share any information about his objective function. In addition, we consider bilevel pricing problems, in which the leader's and follower's objective function do not coincide; see Section 5.2.

Both methods, the MWU and the inverse KKT approach, are implemented in Python 3.9.17 and the corresponding optimization problems are solved with Gurobi 11.0.1. All computations have been executed on a Intel© Core™i7-10510U CPU with 8 cores of 1.8 GHz each and 32 GB RAM.

5.1. Continuous Knapsack Interdiction Problems.

We start by demonstrating the performance of the presented learning methods using the example of the continuous knapsack interdiction instance BKIP_35_1, which has 35 items and which has been considered before in Beck et al. (2023b). For the origin of this instance, we refer to Caprara et al. (2016) and Martello et al. (1999). We focus here on an exemplary instance to show the behavior of the learning methods in detail, while an extensive numerical study ranging over many different instances is out of scope of this work. In doing so, we adapt the instance BKIP_35_1 by relaxing the integrality constraints and considering a quadratic objective function instead of the linear objective function determined by the vector c in the original instance. Based on this vector, we randomly sample entries for the diagonal matrix Q between the minimum and maximum entry of c. We then switch the sign of these matrix diagonal entries, and re-scale c and Q in order to satisfy Assumption 1. This leads to a continuous knapsack interdiction problem of the form

$$\min_{x\in[0,1]^n} \quad y^\top Q y + c^\top y = \sum_{i=1}^{n} q_{ii} y_i^2 + c_i y_i \tag{10a}$$

$$\text{s.t.} \quad v^\top x \leq B, \tag{10b}$$

$$y \in \arg\max_{\bar{y}\in[0,1]^n} \left\{ \bar{y}^\top Q \bar{y} + c^\top \bar{y} \colon w^\top \bar{y} \leq C,\ \bar{y}_i \leq 1 - x_i,\ i \in [n] \right\}, \tag{10c}$$

where $v, w \in \mathbb{R}^n$ are the weight vectors of the leader and the follower, respectively, and $B, C \in \mathbb{R}$ are the leader's and the follower's budget. Furthermore, the lower-level objective function is concave-quadratic with $c \in \mathbb{R}^n$ and a negative-definite diagonal matrix

$$Q \coloneqq \operatorname{Diag}(\alpha_1, \dots, \alpha_n) \in \mathbb{R}^{n\times n}, \quad \alpha_i < 0, \quad i \in [n].$$

Let (x^*, y^*) be the solution of the bilevel problem (10) under full model information and let $F(x^*, y^*) = F(y^*)$ be the leader's objective function value. We note that all variables are bounded. Consequently, Assumption 2 is satisfied. Moreover, the follower's objective is a quadratic utility function, in which the quadratic term $y^\top Qy$ may, e.g., represent saturation effects. Consequently, it is rather natural that the entries of $c \in \mathbb{R}^n$ are positive and that the entries of Q are negative, which leads to Assumption 3 being satisfied as well.

The overall goal of the leader in this setting is thus to learn the $2n$ unknown parameters of the diagonal matrix Q and the vector c.

We note that the considered interdiction problem (10) slightly deviates from the original bilevel formulation ($P^T(c, Q)$), because the upper-level objective function also depends on the unknown objective coefficients. However, this does not affect the corresponding theoretical results, which concern the lower-level objective function. Specifically, in the MWU algorithm 1 (Line 8) the bilevel solutions are always computed for a fixed guess of these objective coefficients. Analogously, this is true for the inverse KKT approach; see Algorithm 2.

5.1.1. *Learning with the Inverse KKT Method.* The feasible set of the lower-level Problem (10c) is polyhedral and the objective function to be maximized is concave. Furthermore, for the data set of observed points (x^t, y^t), y^t is optimal w.r.t. the corresponding x^t for all $t \in [T-1]$. Thus, we can apply the inverse KKT method to the modified BKIP_35_1 instance to compute the unknown coefficients c and Q so as to minimize the violation of the stationarity condition; see Section 4. Using this data set, we can build and solve Problem (9) including the additional constraints of Remark 3 and also make use of Remark 2.

After solving Model (9), we check whether the objective function determined in every iteration by the solution $\bar{c}^T$, $\bar{Q}^T$ is consistent with the observed data. To do so, we compute the relative consistency measure

$$\left| \frac{\left((y^T)^\top \bar{Q}^T y^T + (\bar{c}^T)^\top y^T\right) - \left((\bar{y}^T)^\top \bar{Q}^T \bar{y}^T + (\bar{c}^T)^\top \bar{y}^T\right)}{(\bar{y}^T)^\top \bar{Q}^T \bar{y}^T + (\bar{c}^T)^\top \bar{y}^T} \right|, \tag{11}$$

where y^T is the observation, and $\bar{y}^T$ is an optimizer computed using the solution $\bar{c}^T$, $\bar{Q}^T$, for the corresponding x^T. In what follows, we also call $\bar{y}^T$ the anticipated solution of the follower. For a consistent objective function, the relative consistency measure is zero. Indeed, in our computational results, all relative consistency measure values are below 10^{-3}.

Let us note again that the computed objective function coefficients $(\bar{c}^T, \bar{Q}^T)$, do not necessarily need to coincide with or converge to the unknown c and Q. The inverse KKT method outputs in every iteration an objective function, which is consistent with the input data. However, the observed data points (x^T, y^T) can be explained by different objective functions. Hence, the solutions of the method are not necessarily unique. In fact, Figure 1 shows that we do not learn the true coefficients.

Similarly, Figure 2 (left) shows that the anticipated $(\bar{y}^T)$ and the observed (y^T) follower's solutions do not necessarily match. In the right figure we see, however, that the observed lower-level solutions y^T tend to coincide with the solution y^* under full model information. Although not theoretically guaranteed, after a few iterations the relative error between the observations and y^* is very close to zero.

If the leader is able to solve the Problem (10) under full information, she obtains a bilevel solution (x^*, y^*) and a corresponding objective function value $F(x^*, y^*) = F(y^*)$. In our case, the leader updates missing problem data iteratively, and achieves the

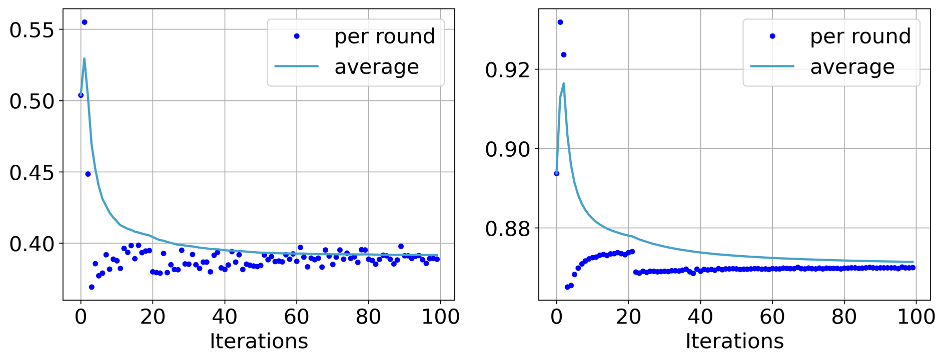

FIGURE 1. Left: $\|c^T - c\|_2/\|c\|_2$. Right: $\|Q^T - Q\|_F/\|Q\|_F$. Both per inverse KKT iteration when applied to the considered continuous knapsack interdiction instance for 100 iterations.

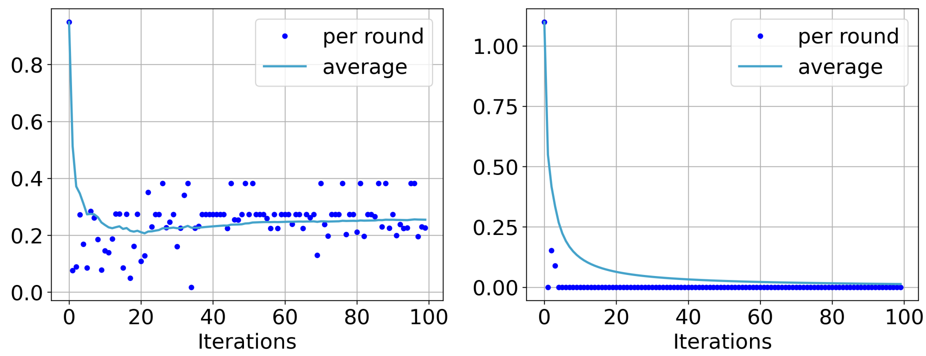

FIGURE 2. Left: Relative error w.r.t. the anticipated $\bar{y}^T$ and the observed y^T follower's solution $\|\bar{y}^T - y^T\|_2/\|y^T\|_2$. Right: Relative error w.r.t. the observed y^T and the true y^* follower's solution $\|y^T - y^*\|_2/\|y^*\|_2$. Both per inverse KKT iteration when applied to the considered continuous knapsack interdiction instance for 100 iterations.

profits $F(x^T, y^T) = F(y^T)$ in every sequential interaction with the follower. Figure 3 (left) shows that after a few interactions with the follower, the upper-level decision x^T is very similar to the solution x^* of the bilevel problem under full information. In Figure 3 (right) we see that the relative error between the objective values $F(x^*, y^*) = F(y^*)$ and $F(x^T, y^T) = F(y^T)$ drops to values close to zero after only a few iterations. Hence, using our approach, the leader is able to achieve profits very similar to $F(x^*, y^*)$, and even take decisions x^T that are similar to x^*. This is not guaranteed in theory but is experimentally the best outcome that one could observe.

To shed some more light on the computational performance of the method, we present runtimes of the inverse KKT approach for 9 different continuous knapsack instances in Table 1. The first number in the name of the instance represents the number of items of the knapsack problem and the second number N is used to compute the capacity C of the follower's knapsack via $\sigma N/11$, where σ is the sum of weights of the follower. We see rather expectable results: The runtimes are increasing with the size of the instances.

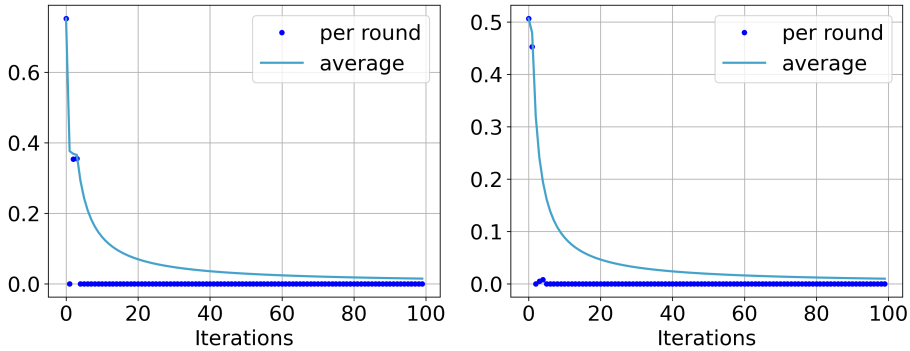

FIGURE 3. Left: Relative error $\|x^T - x^*\|_2/\|x^*\|_2$. Right: Relative error $|F(x^T, y^T) - F(x^*, y^*)|/|F(x^*, y^*)|$ between the computed and the leader's objective function under full information. Both per inverse KKT iteration when applied to the considered continuous knapsack interdiction instance for 100 iterations.

TABLE 1. Runtimes for the inverse KKT method for 9 continuous knapsack interdiction instances

Instance	Runtime (in seconds)
BKIP_35_1	7.07×10^2
BKIP_35_5	1.01×10^4
BKIP_35_10	5.22×10^2
BKIP_20_1	3.14×10^2
BKIP_20_5	3.41×10^2
BKIP_20_10	2.96×10^2
BKIP_10_1	1.50×10^2
BKIP_10_5	1.49×10^2
BKIP_10_10	1.51×10^2

5.1.2. *Learning with the MWU Method.* We now apply the MWU method to the same continuous knapsack instance. While doing so, we successively compute new guesses for the unknown coefficients c_i, $i \in \mathcal{U}^c = [n]$, and Q_{ii}, $(i, i) \in \mathcal{U}^Q = \{(i, i) : i \in [n]\}$. The method outputs a sequence of objective functions $(\bar{c}^T, \bar{Q}^T)$.

Figure 4 (left) shows the MWU error (2) in every iteration of the MWU method when applied to Model (10) for the modified instance BKIP_35_1. In line with the theoretical result, the figure illustrates that this error is decreasing on average for an increasing number of sequential interactions between the leader and the follower. Consequently, also the errors bounded by Corollary 1 converge to zero.

Further, we empirically check if the computed objective functions are consistent w.r.t. the observations of the previous rounds in Figure 4 (right). Unlike for the solutions of the inverse KKT method, consistency is not theoretically guaranteed for the outputs of the MWU method. Nevertheless, we observe that the consistency measure (11) decreases with increasing number of iterations.

In theory, the results for the MWU error (2) do not extend to the anticipated $\bar{y}^T$ and the observed y^T solutions of the follower; see Lines 8 and 9 of Algorithm 1. However, as illustrated in Figure 5 (left), the anticipated follower's solutions match the observations well

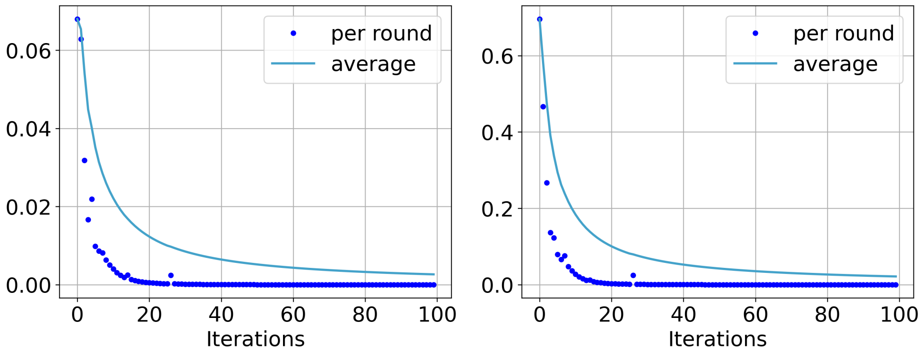

FIGURE 4. Left: MWU error (2), average and per round. Right: Relative consistency measure (11). Both per MWU iteration applied to the considered continuous knapsack interdiction instance for 100 iterations.

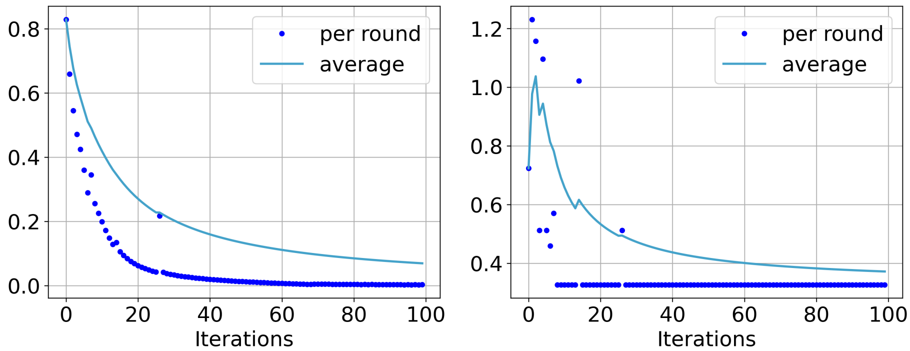

FIGURE 5. Left: Relative error between the anticipated and the observed solutions $\|\bar{y}^T - y^T\|_2/\|y^T\|_2$. Right: Relative error between the observed and the true bilevel solution $\|y^T - y^*\|_2/\|y^*\|_2$. Both per MWU iteration applied to the considered continuous knapsack interdiction instance for 100 iterations.

over time. In practice, this means that we are able to anticipate the follower's reaction to the decision x^T. Nevertheless, it is not implied that the follower's anticipated or observed solutions coincide with the solution (x^*, y^*) of the original continuous knapsack problem under full information; see Figure 5 (right). In addition, Figure 6 shows a qualitatively similar development for the computed coefficients $\bar{c}^T, \bar{Q}^T$.

Moreover, under full model information, the leader solves the knapsack interdiction problem (10), and obtains a bilevel solution (x^*, y^*) and an objective function value $F(x^*, y^*) = F(y^*)$. With the MWU approach, the leader iteratively changes her strategy x^T, and obtains in every iteration T an objective value $F(x^T, y^T) = F(y^T)$. The relative error between the decisions x^T and x^* remains rather large (see Figure 7 left), indicating that the leader's sequential interdiction choices differ from the bilevel optimal choice under full model information. However, the relative error between the objective function values related to these choices $F(x^T, y^T)$, and $F(x^*, y^*)$ decreases iteratively on average, and reaches values very close to zero the longer the MWU method progresses; see Figure 7 (right). Hence, although the leader's decision itself differs, its outcome is of the same quality in our experiments.

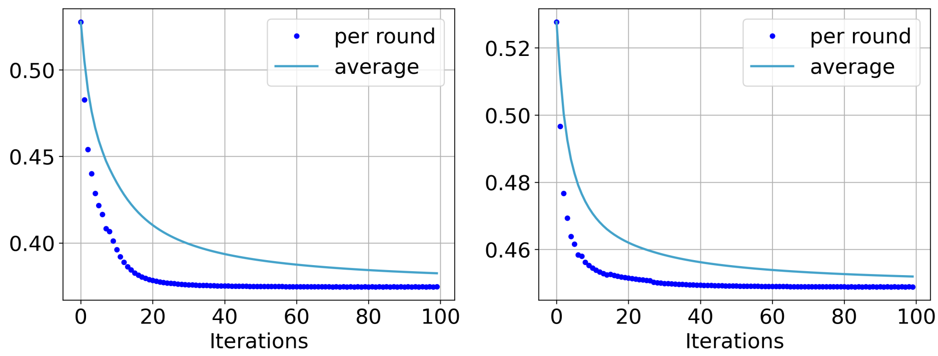

FIGURE 6. Left: $\|c^T - c\|_2/\|c\|_2$. Right: $\|Q^T - Q\|_F/\|Q\|_F$. Both per MWU iteration when applied to the considered continuous knapsack instance for 100 iterations.

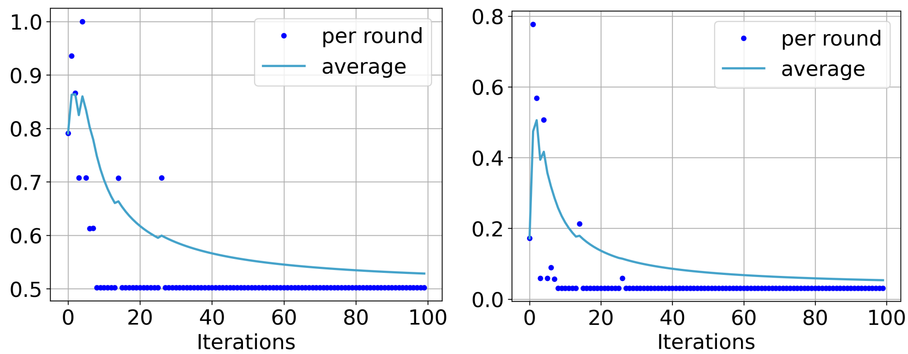

FIGURE 7. Left: Relative error $\|x^T - x^*\|_2/\|x^*\|_2$. Right: Relative error $|F(x^T, y^T) - F(x^*, y^*)|/|F(x^*, y^*)|$ of the true leader's objective function evaluated at the iteratively observed solutions (x^T, y^T) and the true bilevel solution (x^*, y^*) under full information. The iterative solutions are the outputs of the MWU method when applied to the continuous knapsack interdiction instance BKIP_35_1 for 100 iterations.

In order to shed some more light on how the MWU method performs on a broader set of instances, we also applied it to other continuous knapsack interdiction problems. In Table 2, we display the runtimes as well as the relative consistency measure and the MWU error after 100 iterations. First of all, it can be seen that the relative consistency measure and the MWU error are coherent and small. Moreover, we see that the runtimes also seem reasonable. The first number in the ID of the instance refers to the number of items of the knapsack problem. Hence, the larger the knapsack problem, the larger the runtimes.

TABLE 2. Statistics for the MWU method for 9 continuous knapsack interdiction instances

Instance	Runtime (for 100 iter.)	Rel. consistency measure	MWU Error
BKIP_35_1	1.27×10^{3}	1.55×10^{-2}	1.85×10^{-3}
BKIP_35_5	5.75×10^{3}	1.65×10^{-2}	2.04×10^{-3}
BKIP_35_10	3.90×10^{1}	1.92×10^{-2}	3.62×10^{-4}
BKIP_20_1	2.85×10^{1}	7.67×10^{-3}	8.34×10^{-4}
BKIP_20_5	1.17×10^{1}	1.90×10^{-2}	1.37×10^{-3}
BKIP_20_10	8.35×10^{0}	7.91×10^{-2}	7.31×10^{-4}
BKIP_10_1	5.04×10^{0}	9.43×10^{-3}	7.70×10^{-4}
BKIP_10_5	5.31×10^{0}	5.36×10^{-3}	1.00×10^{-3}
BKIP_10_10	4.50×10^{0}	1.61×10^{-2}	7.10×10^{-4}

5.2. **Bilevel Pricing Problems.** As a second application, we consider a bilevel pricing model of the form

$$\begin{aligned} \max_{x,y} \quad & x^\top y \\ \text{s.t.} \quad & x \in [x^-, x^+], \\ & y \in \arg\max_{\bar{y}} \left\{ u(\bar{y}) - x^\top \bar{y} \colon A\bar{y} \geq b,\ 0 \leq \bar{y} \leq y^+ \right\}, \end{aligned} \tag{12}$$

where $x \in \mathbb{R}^n$ is the vector of prices (the upper-level decisions) and $y \in \mathbb{R}^n$ are the purchase decisions of the follower. Moreover, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ describe the polyhedral feasible set of the follower for which we additionally assume that the bounds on the purchase decisions are given explicitly. Further, $u(y) \coloneqq y^\top Q y + c^\top y$ is a concave-quadratic utility function with $c \in \mathbb{R}^n$ and

$$Q \coloneqq \operatorname{Diag}(\alpha_1, \dots, \alpha_n) \in \mathbb{R}^{n \times n}, \quad \alpha_i < 0, \quad i \in [n],$$

is a negative diagonal matrix. While the leader maximizes her revenue by setting prices x, the follower makes a purchase decision y in order to maximize his utility (in which we subtract, as usual, costs). Consequently, as in the previous application, Assumptions 1–3 are rather mild and easy to satisfy. For more information on bilevel pricing models we refer to Bialas and Karwan (1984), Labbé et al. (1998), and Labbé and Violin (2013). As before, the overall goal of the leader is to learn the $2n$ unknown parameters of the diagional matrix Q and the vector c.

For our computations, we randomly sample the entries in the vector c in $[0, 1]$ and the entries on the main diagonal of Q in $[-1, 0]$. Furthermore, we use a randomly chosen subset of the data from the popular Stigler diet problem (Stigler 1945). To be more precise, the vector b represents the minimum recommended amounts of daily nutrients (e.g., protein, calcium, etc.), and the matrix A consists of the amount of nutrients contained in each of the $n = 15$ food items considered.

Thus, the leader sets 15 prices (x_i) for the 15 items that the follower can purchase (y_i). In addition, we use the lower and upper bounds $x_i^- = 0$ and $x_i^+ = 1$ for all $i \in [n]$. The upper bound on y is given by

$$y_i^+ \coloneqq \max\{b_j / A_{ji} \colon j \in [m] \text{ with } A_{ji} \neq 0\} \quad \text{for} \quad i \in [n].$$

We now equivalently reformulate the follower's problem in (12) such that the objective function is independent from products of upper- and lower-level variables, i.e., $x^\top \bar{y}$. The latter is necessary since, from a leader's perspective, x are no objective coefficients that are to be learned. Consequently, we move these products to the feasible set by an epigraph reformulation and consider the objective coefficient of the corresponding new variable as known to the leader. More precisely, we introduce a new variable μ and we obtain the model

$$\max_{y,\mu} \quad y^\top Q y + c^\top y - \mu \quad \text{s.t.} \quad (y, \mu) \in Y(x),$$

where the x-parameterized feasible set $Y(x)$ is given by

$$Y(x) := \left\{ (y, \mu) \in \mathbb{R}^n \times \mathbb{R} \colon Ay \geq b,\ 0 \leq y \leq y^+,\ \mu \geq \sum_{i \in [n]} x_i y_i \right\}.$$

It holds $\mu = \sum_{i \in [n]} x_i y_i$ in every optimal solution. Hence, using $x_i^- = 0$ and $x_i^+ = 1$ for all $i \in [n]$ leads to

$$\mu^- = 0 = \sum_{i \in [n]} x_i^- y_i^- \leq \mu \leq \sum_{i \in [n]} y_i^+ =: \mu^+.$$

Using these lower and upper bounds on μ, we can compute the constant

$$K \geq \max_{(y_1,\mu_1),(y_2,\mu_2) \in Y(x)} \left\{ \|y_1 - y_2\|_\infty, \left\|y_1 y_1^\top - y_2 y_2^\top\right\|_{\max}, |\mu_1 - \mu_2| \right\},$$

which fulfills Assumption 2. Further, let

$$z := \begin{pmatrix} y \\ \mu \end{pmatrix} \in \mathbb{R}^{n+1}, \quad \tilde{c} := \begin{pmatrix} c \\ -1 \end{pmatrix} \in \mathbb{R}^{n+1}, \quad \tilde{b} := \begin{pmatrix} b \\ 0 \end{pmatrix} \in \mathbb{R}^{m+1},$$
$$\tilde{Q} := \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad \tilde{A} := \begin{bmatrix} A & 0 \\ -x^\top & 1 \end{bmatrix} \in \mathbb{R}^{(m+1)\times(n+1)}.$$

Then, the lower level can be re-written as

$$\begin{aligned} \max_z \quad & z^\top \tilde{Q} z + \tilde{c}^\top z \\ \text{s.t.} \quad & \tilde{A}(x) z \geq \tilde{b}, \\ & z_i^- \leq z_i \leq z_i^+,\ i \in [n+1], \end{aligned}$$

with $z^+ := (y^+, \mu^+)$ and $z^- := (y^-, \mu^-)$, so that it fits in the framework considered in the previous sections. However, in what follows, we only present results for the original variables y and coefficients c and Q.

5.2.1. *Learning with the Inverse KKT Method.* We now apply the inverse KKT approach to the bilevel pricing problem. Let us first note here that it turned out to be essential for obtaining good results to incorporate the constraints discussed in Remark 3. As expected due to the construction of the method, we obtain consistent objective functions with coefficients $\bar{c}^T$ and $\bar{Q}^T$ in every iteration T. More specifically, the relative consistency measure (11) is below 10^{-5} in every iteration.

Nevertheless, and as already discussed before, the coefficients of the obtained objective functions do not necessarily need to coincide with the true objective function coefficients c and Q. In this case, however, we obtain relative differences

$$\left| \frac{\bar{c}_i^{T^{\max}} - c_i}{c_i} \right|, \quad \left| \frac{\bar{Q}_{ii}^{T^{\max}} - Q_{ii}}{Q_{ii}} \right|, \quad i \in [n],$$

of the computed and real coefficients after the last iteration $T^{\max}$ that are all below 10^{-4}. In fact, the relative errors depicted in Figure 8 show that the computed coefficients $\bar{c}^T$, $\bar{Q}^T$ match the true after relatively few iterations.

Furthermore, we can see in Figure 9 (left) that as the iterations progress, we are able to anticipate the follower's solutions y^T increasingly better. Even more, after only a few

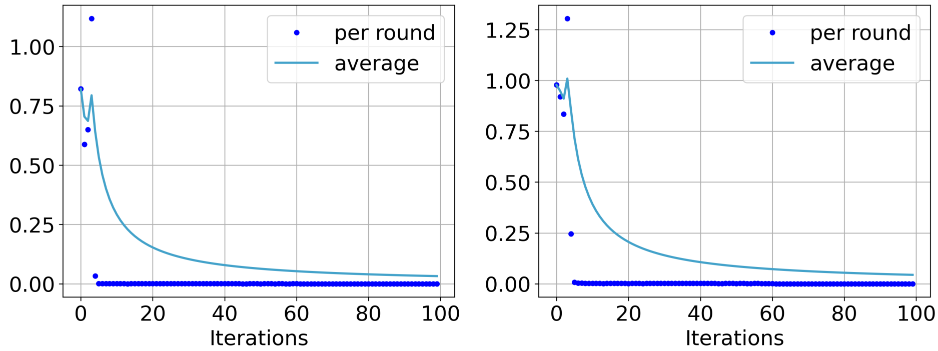

FIGURE 8. Left: $\|\bar{c}^T - c\|_2/\|c\|_2$. Right: $\|\bar{Q}^T - Q\|_F/\|Q\|_F$. Both per inverse KKT iteration when applied to the considered bilevel pricing instance for 100 iterations.

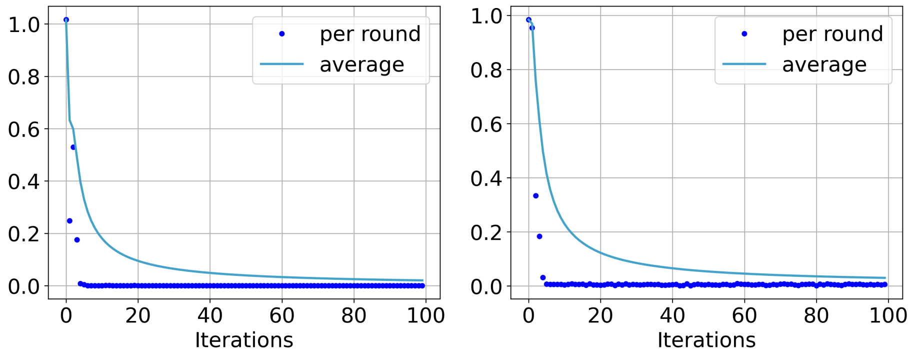

FIGURE 9. Left: Relative error between the anticipated and the observed solutions $\|\bar{y}^T - y^T\|_2/\|y^T\|_2$. Right: Relative error between the observed and the true bilevel solution $\|y^T - y^*\|_2/\|y^*\|_2$. Both per inverse KKT iteration applied to the considered bilevel pricing instance for 100 iterations.

iterations, the observed follower's solution y^T and the solution y^* of the bilevel problem's lower level under full information are relatively close to each other; see Figure 9 (right).

Lastly, under full model information, the leader is able to solve the Problem (12) and obtain a bilevel solution (x^*, y^*) with objective function value $F(x^*, y^*)$. In our case under missing problem data, the leader adapts her decisions iteratively and sets the prices x^T, and the follower responds with the purchase decisions y^T. Thus, the leader's objective function value after every sequential interaction with the follower is given by $F(x^T, y^T)$. Figure 10 shows that with the inverse KKT approach, the relative error between the objective function values $F(x^*, y^*)$ and $F(x^T, y^T)$, as well as the relative error between the leader's solution x^* and the sequentially played strategies x^T are both close to zero after a few iterations. In this case we were able to perfectly replicate not only the bilevel optimal solutions and objective function value, but also find the exact coefficients.

5.2.2. *Learning with the MWU Method.* Let $(\bar{c}^T, \bar{Q}^T)$, $T \in [T^{\max}]$, be the output of the MWU method applied to the considered bilevel pricing problem. We illustrate the MWU error (2) made in every iteration of the MWU method in the left plot of Figure 11. The

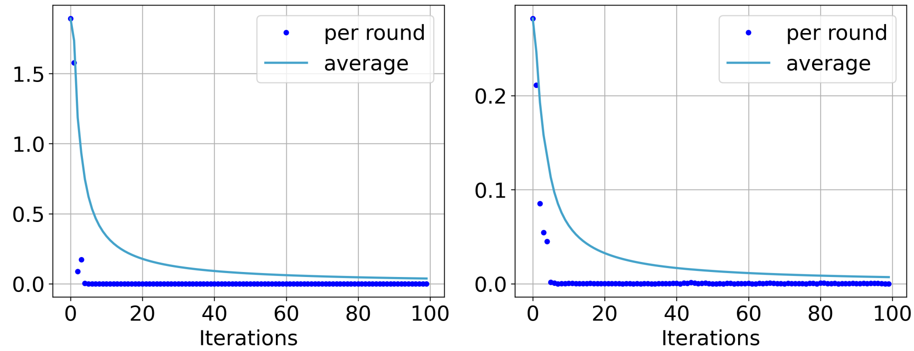

FIGURE 10. Left: Relative error $\|x^T - x^*\|_2/\|x^*\|_2$. Right: Relative error $|F(x^T, y^T) - F(x^*, y^*)|/|F(x^*, y^*)|$ of the true leader's objective function evaluated at the iteratively observed solutions (x^T, y^T) and the true bilevel solution (x^*, y^*). The solutions are the outputs of the inverse KKT method applied to the considered bilevel pricing instance for 100 iterations.

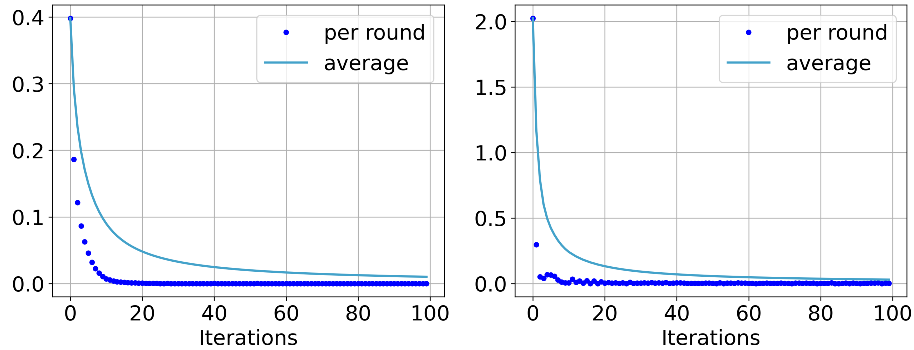

FIGURE 11. Left: MWU error (2), average and per round. Right: Relative consistency measure (11). Both per MWU iteration applied to the considered bilevel pricing instance for 100 iterations.

general behavior is the same as in the previous section, i.e., as the theory guarantees, the error is converging to zero. The right plot of the Figure 11 shows the relative consistency values (11) of the MWU outputs over iterations. Although these values are not theoretically guaranteed to be small, we observe that they decrease rapidly with increasing number of iterations.

The left plot of Figure 12 shows that the relative error of the anticipated $\bar{y}^T$ and the observed follower's solutions y^T obtained with the MWU method applied to our pricing instance converges to zero. While we cannot guarantee this behavior theoretically, in practice we are often able to anticipate the follower's solution relatively well over time. Furthermore, despite being able to anticipate the follower's response at the market better over iterations, the leader cannot guarantee that the anticipated $\bar{y}^T$ or the observed y^T purchasing patterns will match the solution (x^*, y^*) of the bilevel problem under full information; see Figure 12 (right). There can be multiple reasons for this such as ambiguous lower-level solutions or the different upper-level decisions that parameterize the lower-level problem. In addition, we compare the learned objective coefficients with

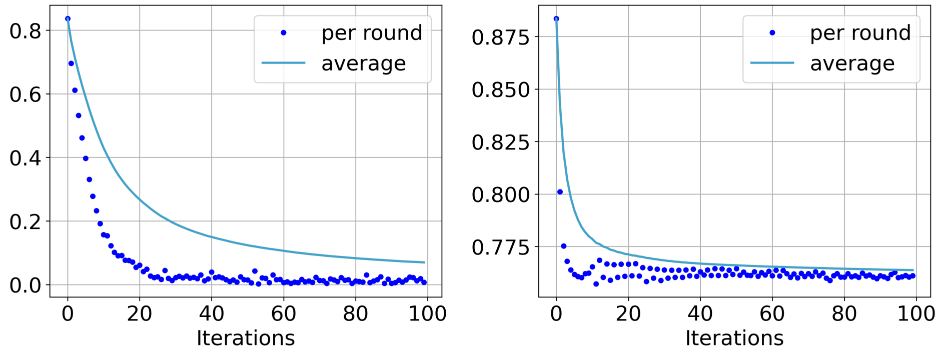

FIGURE 12. Left: Relative error between the anticipated and the observed follower's solutions $\|\bar{y}^T - y^T\|_2/\|y^T\|_2$. Right: Relative error between the observed and the true bilevel solution $\|y^T - y^*\|_2/\|y^*\|_2$. Both per MWU iteration applied to the considered bilevel pricing instance for 100 iterations.

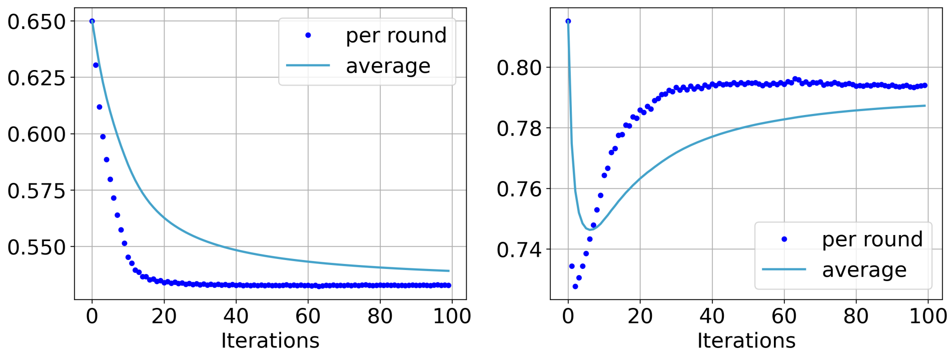

FIGURE 13. Left: Relative error between the computed and the true, unknown coefficients $\|\bar{c}^T - c\|_2/\|c\|_2$. Right: Relative error between the computed and the true, unknown coefficients $\|\bar{Q}^T - Q\|_F/\|Q\|_F$. Both per MWU iteration when applied to the considered bilevel pricing instance for 100 iterations.

the true coefficients in Figure 13. We see that low MWU errors and low consistency errors (see Figure 11) do not ensure low relative coefficient errors.

Finally, if the leader is able to solve the Problem (12) under full information, she obtains a bilevel solution (x^*, y^*) and a corresponding profit $F(x^*, y^*)$. Using our iterative MWU approach under missing problem data, the leader chooses the prices x^T iteratively, and earns a profit $F(x^T, y^T)$. Figure 14 (left) illustrates that the leader chooses prices relatively similar to x^* after a few MWU iterations. Furthermore, the profits resulting with our MWU method approach are close to the profit obtained solving the exact bilevel pricing instance after about 20 iterations. The corresponding relative error seems to stagnate at a relatively low value; see Figure 14 (right). Thus, without replicating the exact missing data c and Q, the leader is able to attain profits close to $F(x^*, y^*)$.

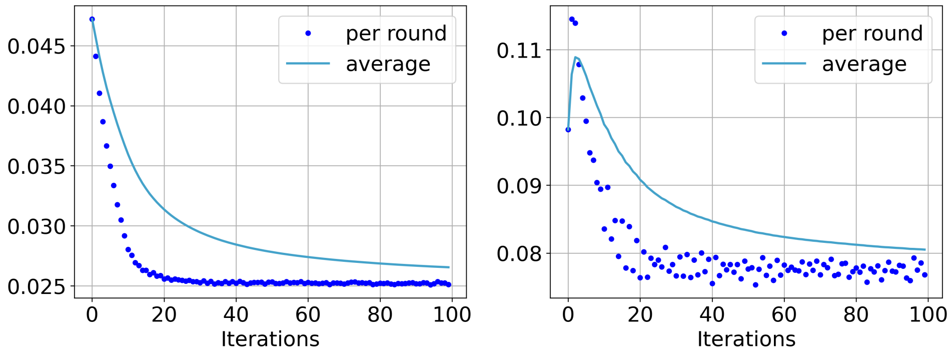

FIGURE 14. Left: Relative error $\|x^T - x^*\|_2/\|x^*\|_2$. Right: Relative error $|F(\bar{x}^T, y^T) - F(x^*, y^*)|/|F(x^*, y^*)|$, of the true leader's objective function evaluated at the iteratively observed solutions (x^T, y^T) and the true bilevel solution (x^*, y^*) under full information. The iterative solutions are the outputs of the MWU method applied to the considered bilevel pricing instance for 100 iterations.

6. Conclusion

In many practical applications of bilevel optimization it is not the case that the leader actually has full knowledge about the optimization problem of the follower. However, most of the literature makes this strong assumption. In this paper, we propose two approaches with which the leader can gather important information about the objective function (value) of the follower if the bilevel game is played repeatedly. We discuss the theoretical properties of the two methods and also show, using the example of two distinct case studies, that both methods work well in practice. The obvious question about which of the two methods is "better" and should be used is hard to answer in general. The two methods require different assumptions and they also lead to qualitatively different results. Hence, the decision about which method to use in practice should always be answered depending on the goal of its application and on the assumptions that are satisfied or not.

Several directions for future research are possible. First, it would be interesting to see how these approaches work in real-world applications. Moreover, we focus here on an exemplary instances to show the behavior of the learning methods in detail. Given the overall length of this paper, a more extensive numerical study is out of scope of this work but is, of course, a reasonable topic for future research. Second, the generalization to more complicated classes of lower-level objective functions is important and would be required for many real-world problems. Third, one could think about a leader that acts strategically in the sense that she might put more emphasis on better learning rates in early stages of the sequential game by making suboptimal decisions that reveal more information about the objective function of the follower. Fourth, we focused entirely on learning objective functions in this paper while assuming that the follower's feasible set is known to the leader. A natural next step would thus be to also learn the follower's constraints. Fifth and finally, we are still missing theoretical results regarding the outcomes of the leader. While our numerical results show that the leader's objective function value over time tends to the objective function value under full information, there are no guarantees for that. We think that it will be hard to obtain such theoretical results since very recent results (Beck et al. 2023c) show that approximate solutions of the lower-level can yield arbitrarily bad outcomes for the leader. Nevertheless, the investigation of the above mentioned guarantees would for sure be important in this field.

Acknowledgements

The authors thank the DFG for their support within the research training group 2126 "Algorithmic Optimization". Moreover, the authors acknowledge the use of DeepL and OpenAI's ChatGPT for partly editing and polishing the text and figures for spelling, grammar, and stylistic improvements.

Appendix A. Auxiliary Results

Based on the Theorem 2.1 in Arora et al. (2012) and using the same notation, we have a similar result for η^t, $t \in [T]$, for some given $T \in \mathbb{N}$.

Theorem 4. *Suppose that $m_i^t \in [-1, 1]$, $i \in [n]$, and $\eta^t \leq 1/2$ holds for all $t \in [T]$. Then, the multiplicative weights algorithm in Arora et al. (2012) guarantees that after T rounds, for any decision $i \in [n]$, we have*

$$\sum_{t=1}^{T} \eta^t \boldsymbol{m}^t \cdot \boldsymbol{p}^t \leq \ln(n) + \sum_{t=1}^{T} \eta^t m_i^t + \sum_{t=1}^{T} (\eta^t)^2 |m_i^t|.$$

Proof. Let $\Phi^t \coloneqq \sum_{i=1}^{n} w_i^t$ be the sum of all weights in iteration t. Then, in iteration $t+1$ we use the fact that $p_i^t = w_i^t/\Phi^t$ holds and obtain

$$\begin{aligned} \Phi^{t+1} &= \sum_{i=1}^{n} w_i^{t+1} = \sum_{i=1}^{n} w_i^t (1 - \eta^t m_i^t) = \sum_{i=1}^{n} w_i^t - \eta^t \sum_{i=1}^{n} w_i^t m_i^t \\ &= \Phi^t - \eta^t \Phi^t \sum_{i=1}^{n} m_i^t p_i^t = \Phi^t (1 - \eta^t \mathbf{m}^t \cdot \mathbf{p}^t) \leq \Phi^t \exp(-\eta^t \mathbf{m}^t \cdot \mathbf{p}^t). \end{aligned}$$

After T rounds, we have

$$\Phi^{t+1} \leq \Phi^1 \exp\left(-\sum_{t=1}^{T} \eta^t \mathbf{m}^t \cdot \mathbf{p}^t\right) = n \exp\left(-\sum_{t=1}^{T} \eta^t \mathbf{m}^t \cdot \mathbf{p}^t\right). \tag{13}$$

Now, we use

$$\begin{aligned} (1 - \eta^t x) &\geq (1 - \eta^t)^x, & x &\in [0, 1], \\ (1 - \eta^t x) &\geq (1 + \eta^t)^{-x}, & x &\in [-1, 0]. \end{aligned}$$

Hence, since $m_i^t \in [-1, 1]$ for all $i \in [n]$ and $t \in [T]$,

$$\Phi^{T+1} \geq w_i^{T+1} = \prod_{t \leq T} (1 - \eta^t m_i^t) \geq \prod_{t \geq 0} (1 - \eta^t)^{m_i^t} \prod_{t < 0} (1 + \eta^t)^{-m_i^t}, \tag{14}$$

holds, where the subscripts "$t \geq 0$" and "$t < 0$" in the products above indicate the rounds t in which it holds $m_i^t \geq 0$ or $m_i^t < 0$, respectively. We now take the logarithm in the inequalities (13) and (14) and obtain

$$\ln(n) - \sum_{t=1}^{T} \eta^t \mathbf{m}^t \cdot \mathbf{p}^t \geq \sum_{t \geq 0} m_i^t \ln(1 - \eta^t) - \sum_{t < 0} m_i^t \ln(1 + \eta^t),$$

which is equivalent to

$$\sum_{t=1}^{T} \eta^t \mathbf{m}^t \cdot \mathbf{p}^t \leq \ln(n) + \sum_{t \geq 0} m_i^t \ln\left(\frac{1}{1 - \eta^t}\right) + \sum_{t < 0} m_i^t \ln(1 + \eta^t).$$

For $\eta^t \leq 1/2$, it holds $\ln(1/(1-\eta^t)) \leq \eta^t + (\eta^t)^2$ and $\ln(1+\eta^t) \geq \eta^t - (\eta^t)^2$. Thus, we can write

$$\begin{aligned}
\sum_{t=1}^{T} \eta^t \mathbf{m}^t \cdot \mathbf{p}^t &\leq \ln(n) + \sum_{t\geq 0} m_i^t \ln\left(\frac{1}{1-\eta^t}\right) + \sum_{t<0} m_i^t \ln(1+\eta^t) \\
&\leq \ln(n) + \sum_{t\geq 0} m_i^t (\eta^t + (\eta^t)^2) + \sum_{t<0} m_i^t (\eta^t - (\eta^t)^2) \\
&= \ln(n) + \sum_{t\geq 0} \eta^t m_i^t + \sum_{t\geq 0} (\eta^t)^2 m_i^t + \sum_{t<0} \eta^t m_i^t - \sum_{t<0} (\eta^t)^2 m_i^t \\
&= \ln(n) + \sum_{t=1}^{T} \eta^t m_i^t + \sum_{t\geq 0} (\eta^t)^2 m_i^t - \sum_{t<0} (\eta^t)^2 m_i^t \\
&= \ln(n) + \sum_{t=1}^{T} \eta^t m_i^t + \sum_{t=1}^{T} (\eta^t)^2 |m_i^t|. \qquad \square
\end{aligned}$$

Therefore, we can, analogously to Corollary 2.2 in Arora et al. (2012), derive the following statement.

Corollary 3. *The MWU algorithm guarantees that after T rounds, for any distribution $\boldsymbol{p}$ on the decisions,*

$$\sum_{t=1}^{T} \eta^t \boldsymbol{m}^t \cdot \boldsymbol{p}^t \leq \ln(n) + \sum_{t=1}^{T} \eta^t (\boldsymbol{m}^t + \eta^t |\boldsymbol{m}^t|) \cdot \boldsymbol{p}, \tag{15}$$

holds where $|\boldsymbol{m}^t|$ is the vector obtained by taking the coordinate-wise absolute value of $\boldsymbol{m}^t$.

References

Ahuja, R. K. and J. B. Orlin (2001). "Inverse Optimization." In: *Operations Research* 49.5, pp. 771–783. DOI: 10.1287/opre.49.5.771.10607.

Arora, S., E. Hazan, and S. Kale (2012). "The multiplicative weights update method: a meta-algorithm and applications." In: *Theory of Computing* 8.1, pp. 121–164. DOI: 10.4086/toc.2012.v008a006.

Beck, Y., I. Ljubić, and M. Schmidt (2022). "A Brief Introduction to Robust Bilevel Optimization." In: *SIAG on Optimization Views and News. A Forum for the SIAM Activity Group on Optimization* 30.2, pp. 1–10. URL: https://siagoptimization.github.io/assets/views/ViewsAndNews-30-2.pdf.

– (2023a). "A survey on bilevel optimization under uncertainty." In: *European Journal of Operational Research* 311.2, pp. 401–426. DOI: 10.1016/j.ejor.2023.01.008.

– (2023b). "Exact methods for discrete Γ-robust interdiction problems with an application to the bilevel knapsack problem." In: *Mathematical Programming Computation* 15.4, pp. 733–782. DOI: 10.1007/s12532-023-00244-6.

Beck, Y., M. Schmidt, J. Thürauf, and D. Bienstock (2023c). "On a Computationally Ill-Behaved Bilevel Problem with a Continuous and Nonconvex Lower Level." In: *Journal of Optimization Theory and Applications* 198.1, pp. 428–447. DOI: 10.1007/s10957-023-02238-9.

Bennett, K. P., G. Kunapuli, J. Hu, and J.-S. Pang (2008). "Bilevel Optimization and Machine Learning." In: *Computational Intelligence: Research Frontiers: IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008, Plenary/Invited Lectures.* Ed. by J. M. Zurada, G. G. Yen, and J. Wang. Springer Berlin Heidelberg, pp. 25–47. DOI: 10.1007/978-3-540-68860-0_2.

Besbes, O., Y. Fonseca, and I. Lobel (2025). "Contextual Inverse Optimization: Offline and Online Learning." In: *Operations Research* 73.1, pp. 424–443. DOI: 10.1287/opre.2021.0369.

Bialas, W. F. and M. H. Karwan (1984). "Two-level linear programming." In: *Management Science* 30.8, pp. 1004–1020. DOI: 10.1287/mnsc.30.8.1004.

Borrero, J. S., O. A. Prokopyev, and D. Sauré (2016). "Sequential Shortest Path Interdiction with Incomplete Information." In: *Decision Analysis* 13.1, pp. 68–98. DOI: 10.1287/deca.2015.0325.

– (2019). "Sequential Interdiction with Incomplete Information and Learning." In: *Operations Research* 67.1, pp. 72–89. DOI: 10.1287/opre.2018.1773.

– (2022). "Learning in Sequential Bilevel Linear Programming." In: *INFORMS Journal on Optimization* 4.2, pp. 174–199. DOI: 10.1287/ijoo.2021.0063.

Bärmann, A., A. Martin, S. Pokutta, and O. Schneider (2018). "An online-learning approach to inverse optimization." In: *arXiv preprint arXiv:1810.12997*. URL: https://doi.org/10.48550/arXiv.1810.12997.

Bärmann, A., S. Pokutta, and O. Schneider (2017). "Emulating the Expert: Inverse Optimization through Online Learning." In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 400–410. URL: https://proceedings.mlr.press/v70/barmann17a.html.

Caprara, A., M. Carvalho, A. Lodi, and G. J. Woeginger (2016). "Bilevel Knapsack with Interdiction Constraints." In: *INFORMS Journal on Computing* 28.2, pp. 319–333. DOI: 10.1287/ijoc.2015.0676.

Clarke, S., G. Dragotto, J. F. Fisac, and B. Stellato (2023). "Learning Rationality in Potential Games." In: *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 4261–4266. DOI: 10.1109/CDC49753.2023.10383714.

Dempe, S. (2002). *Foundations of Bilevel Programming*. Springer. DOI: 10.1007/b101970.

Dempe, S., V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova (2015). *Bilevel Programming Problems*. Springer. DOI: 10.1007/978-3-662-45827-3.

Dong, C., Y. Chen, and B. Zeng (2018). "Generalized inverse optimization through online learning." In: *Advances in Neural Information Processing Systems* 31. URL: https://doi.org/10.48550/arXiv.1810.01920.

Dumouchelle, J., E. Julien, J. Kurtz, and E. B. Khalil (2024). *Neur2BiLO: Neural Bilevel Optimization*. URL: https://arxiv.org/abs/2402.02552.

Fajemisin, A. O., D. Maragno, and D. den Hertog (2024). "Optimization with constraint learning: A framework and survey." In: *European Journal of Operational Research* 314.1, pp. 1–14. DOI: 10.1016/j.ejor.2023.04.041.

Goerigk, M., J. Kurtz, M. Schmidt, and J. Thürauf (2025). "Connections between Robust and Bilevel Optimization." In: *Open Journal of Mathematical Optimization* 6.2, pp. 1–17. DOI: 10.5802/ojmo.38.

Hansen, P., B. Jaumard, and G. Savard (1992). "New branch-and-bound rules for linear bilevel programming." In: *SIAM Journal on Scientific and Statistical Computing* 13.5, pp. 1194–1217. DOI: 10.1137/0913069.

Iraj, E. H. and D. Terekhov (2021). *Comparing Inverse Optimization and Machine Learning Methods for Imputing a Convex Objective Function*. Tech. rep. DOI: 10.48550/arXiv.2102.10742.

Jeroslow, R. G. (1985). "The polynomial hierarchy and a simple model for competitive analysis." In: *Mathematical Programming* 32.2, pp. 146–164. DOI: 10.1007/BF01586088.

Keshavarz, A., Y. Wang, and S. Boyd (2011). "Imputing a convex objective function." In: *2011 IEEE international symposium on intelligent control*. IEEE, pp. 613–619. DOI: 10.1109/ISIC.2011.6045410.

Khanduri, P., S. Zeng, M. Hong, H.-T. Wai, Z. Wang, and Z. Yang (2021). "A Near-Optimal Algorithm for Stochastic Bilevel Optimization via Double-Momentum." In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., pp. 30271–30283. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/fe2b421b8b5f0e7c355ace66a9fe0206-Paper.pdf.

Kleinert, T., M. Labbé, I. Ljubić, and M. Schmidt (2021). "A Survey on Mixed-Integer Programming Techniques in Bilevel Optimization." In: *EURO Journal on Computational Optimization* 9, p. 100007. DOI: 10.1016/j.ejco.2021.100007.

Kwon, S., H. Choi, and S. Park (2024). "Deep learning based high accuracy heuristic approach for knapsack interdiction problem." In: *Computers & Operations Research*, p. 106965. DOI: https://doi.org/10.1016/j.cor.2024.106965.

Kwon, S. and S. Park (2022). *Solving Bilevel Knapsack Problem using Graph Neural Networks*. Tech. rep. URL: https://arxiv.org/abs/2211.13436.

Labbé, M., P. Marcotte, and G. Savard (1998). "A bilevel model of taxation and its application to optimal highway pricing." In: *Management Science* 44.12-part-1, pp. 1608–1622. DOI: 10.1287/mnsc.44.12.1608.

Labbé, M. and A. Violin (2013). "Bilevel programming and price setting problems." In: *4OR* 11.1, pp. 1–30. DOI: 10.1007/s10288-012-0213-0.

Li, Y. and S. Han (2023). *Solving Strongly Convex and Smooth Stackelberg Games Without Modeling the Follower*. Tech. rep. URL: https://arxiv.org/abs/2303.06192.

Martello, S., D. Pisinger, and P. Toth (1999). "Dynamic Programming and Strong Bounds for the 0-1 Knapsack Problem." In: *Management Science* 45.3. DOI: 10.1287/mnsc.45.3.414.

Molan, I. and M. Schmidt (2023). "Using neural networks to solve linear bilevel problems with unknown lower level." In: *Optimization Letters* 17.5, pp. 1083–1103. DOI: 10.1007/s11590-022-01958-7.

Sessa, P. G., I. Bogunovic, M. Kamgarpour, and A. Krause (2020). "Learning to Play Sequential Games versus Unknown Opponents." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 8971–8981. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/65cf25ef90de99d93fa96dc49d0d8b3c-Paper.pdf.

Stigler, G. J. (1945). "The Cost of Subsistence." In: *Journal of Farm Economics* 27.2, pp. 303–314. DOI: 10.2307/1231810.

Tan, Y., D. Terekhov, and A. Delong (2020). "Learning Linear Programs from Optimal Decisions." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 19738–19749. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/e44e875c12109e4fa3716c05008048b2-Paper.pdf.

Vlah, D., K. Šepetanc, and H. Pandˇ zić (2022). *Solving Bilevel Optimal Bidding Problems Using Deep Convolutional Neural Networks*. Tech. rep. DOI: 10.48550/ARXIV.2207.05825.

Yang, J., J. S. Borrero, O. A. Prokopyev, and D. Sauré (2021). "Sequential Shortest Path Interdiction with Incomplete Information and Limited Feedback." In: *Decision Analysis* 18.3, pp. 218–244. DOI: 10.1287/deca.2021.0426.

(I. Molan, M. Schmidt) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany

Email address: molan@uni-trier.de

Email address: martin.schmidt@uni-trier.de

(J. Thürauf) University of Technology Nuremberg (UTN), Department Liberal Arts and Social Sciences, Discrete Optimization Lab, Dr.-Luise-Herzberg-Str. 4, 90461 Nuremberg, Germany

Email address: johannes.thuerauf@utn.de