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 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 fashion of inverse optimization. The MWU method requires less assumptions but the convergence guarantee is also only on the objective function values, whereas the inverse KKT method requires stronger assumptions but actually allows to learn the objective function itself. 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.

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; see, e.g., Hansen et al. (1992) and Jeroslow (1985) for hardness results. 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 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.

Our main contribution are two methods to learn the objective function (values) of the follower. 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

Date: August 18, 2023.

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.

et al. (2017) and Bärmann et al. (2018), 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 and by considering quadratic objective functions, for which we 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 the details 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.

As the last reference already suggests, 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. (2023) 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.

Moreover, 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 an uncertain parameter of the overall model. For a general overview of the field of bilevel optimization under uncertainty, we refer to the recent survey by Beck et al. (2023b) and for its connection to robust optimization to Goerigk et al. (2023). 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. 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. A very similar setting compared to ours is studied by Borrero et al. (2022). However, the methods used and the results obtained are completely different; 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). 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 consider a kind of opposite situation in which the uncertainty is in the upper- and not in the lower-level problem. In a very 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, similar 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, where we summarize our findings and where we also sketch some ideas for potential future research directions.

2. Problem Statement

We consider the hierarchical interaction between a leader and a follower modeled by a bilevel problem of the general form

$$\begin{aligned} \max_{x,y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \leq 0, \\ & y \in \Psi(x), \end{aligned}$$

where $\Psi(x)$ is the set of optimal solutions of the x-parameterized problem

$$\begin{aligned} \max_{y} \quad & f(x,y) \\ \text{s.t.} \quad & y \in Y(x). \end{aligned}$$

The quadratic objective function $f(x,y) \coloneqq y^\top Qy + c^\top y$ of the follower is defined by the matrix $Q \in \mathbb{R}^{n\times n}$ and the vector $c \in \mathbb{R}^n$ and is assumed to be only partially known to the leader. We denote by $\mathcal{U}^c \subseteq [n] \coloneqq \{1,\dots,n\}$ and $\mathcal{U}^Q \subseteq [n]^2 \coloneqq [n]\times[n]$ the index sets of the unknown entries in c and Q, respectively. As a consequence, the leader needs to make a decision x while being unaware of the lower-level objective function and needs to wait to observe the corresponding reaction y of the follower. We assume that the corresponding bilevel game between the leader and the follower is played repeatedly for $T < \infty$ times, leading to a sequence of T upper-level decisions x^t and corresponding lower-level reactions y^t, one for every $t \in [T] \coloneqq \{1,\dots,T\}$.

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

$$\max_{y} \quad y^\top Qy + c^\top y \tag{1a}$$

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

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), $t\in[T]$, and present two inverse optimization methods to learn the objective function (value) of the follower.

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, we consider quadratic lower-level objective functions (1a) that are not or only partially known and we use past observations (x^t, y^t), $t \in [T]$, from leader-follower interactions to iteratively update the leader's guess of the unknown follower's objective function.

Our analysis follows the one applied by Bärmann et al. (2017) and Bärmann et al. (2018), who use an MWU algorithm to learn objective 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. For doing 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 re-scaled so that the unknown coefficients satisfy the assumption.

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

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

holds for all upper-level feasible decisions x.

Note that Assumption 2 is equivalent to requiring 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*

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

and

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

are known.

If the latter assumption is strong in practice depends on the specific problem at hand and we will discuss it 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 $t \in [T]$, the algorithm first updates the coefficients c^t and Q^t for the unknown objective coefficients while the known objective ones remain the same; see Line 4–6. Afterward, the follower's problem is solved using these computed

Algorithm 1 A Multiplicative Weight Update Method

Input: Observations (x^t, y^t) for all $t \in [T]$, index sets $\mathcal{U}^c, \mathcal{U}^Q$, known coefficients $c_i, i \in [n] \setminus \mathcal{U}^c$, and $Q_{ij}, (i,j) \in [n]^2 \setminus \mathcal{U}^Q$, as well as index sets $\mathcal{P}^c$, $\mathcal{N}^c$, $\mathcal{P}^Q$, $\mathcal{N}^Q$
Output: A sequence of objectives $(\bar{c}^1, \bar{Q}^1), \dots, (\bar{c}^T, \bar{Q}^T)$

1: Set $\eta \leftarrow \sqrt{\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|)/T}$ and $t \leftarrow 1$.
2: Set $w^t_{c_i} \leftarrow 1$ for all $i \in \mathcal{U}^c$ and set $w^t_{Q_{ij}} \leftarrow 1$ for all $(i,j) \in \mathcal{U}^Q$.
3: **while** $t \leq T$ **do**
4: Set $c^t_i \leftarrow c_i$ for $i \in [n] \setminus \mathcal{U}^c$ and $Q^t_{ij} \leftarrow Q_{ij}$ for $(i,j) \in [n]^2 \setminus \mathcal{U}^Q$.
5: Set $f \leftarrow \sum_{i \in \mathcal{U}^c} |w^t_{c_i}| + \sum_{(i,j) \in \mathcal{U}^Q} |w^t_{Q_{ij}}|$.
6: Set $c^t_i \leftarrow w^t_{c_i}/f$ for all $i \in \mathcal{U}^c$ and set $Q^t_{ij} \leftarrow w^t_{Q_{ij}}/f$ for all $(i,j) \in \mathcal{U}^Q$.
7: Compute $\bar{y}^t$ by solving $(P(x^t, Q^t, c^t))$.
8: Set $d \leftarrow \max\left\{\max_{i \in \mathcal{U}^c} \{|\bar{y}^t_i - y^t_i|\}, \max_{(i,j) \in \mathcal{U}^Q} \left\{\left|\bar{y}^t_i \bar{y}^t_j - y^t_i y^t_j\right|\right\}\right\}$.
9: **if** $d = 0$ **then**
10: Set $e^t_{c_i} \leftarrow 0$ for all $i \in \mathcal{U}^c$ and set $e^t_{Q_{ij}} \leftarrow 0$ for all $(i,j) \in \mathcal{U}^Q$.
11: **else**
12: Set $e^t_{c_i} \leftarrow (\bar{y}^t_i - y^t_i)/d$ for all $i \in \mathcal{P}^c \cap \mathcal{U}^c$ and $e^t_{c_i} \leftarrow (y^t_i - \bar{y}^t_i)/d$ for all $i \in \mathcal{N}^c \cap \mathcal{U}^c$.
13: Set $e^t_{Q_{ij}} \leftarrow (\bar{y}_i \bar{y}_j - y_i y_j)/d$ for all $(i,j) \in \mathcal{P}^Q \cap \mathcal{U}^Q$ and $e^t_{Q_{ij}} \leftarrow (y_i y_j - \bar{y}_i \bar{y}_j)/d$ for all $(i,j) \in \mathcal{N}^Q \cap \mathcal{U}^Q$
14: **end if**
15: Set $w^{t+1}_{c_i} \leftarrow w^t_{c_i}(1 - \eta e^t_{c_i})$ for all $i \in \mathcal{U}^c$ and set $w^{t+1}_{Q_{ij}} \leftarrow w^t_{Q_{ij}}(1 - \eta e^t_{Q_{ij}})$ for all $(i,j) \in \mathcal{U}^Q$.
16: 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 $i \in [n] \setminus (\mathcal{U}^c \cap \mathcal{N}^c)$.
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 $(i,j) \in [n]^2 \setminus (\mathcal{U}^Q \cap \mathcal{N}^Q)$.
17: Set $t \leftarrow t + 1$.
18: **end while**
19: **return** $(\bar{c}^1, \bar{Q}^1), \dots, (\bar{c}^T, \bar{Q}^T)$

coefficients for the objective function, which leads to solving the problem

$$\max_{y \in Y(x^t)} \left\{ \sum_{(i,j) \in I_1} Q^t_{ij} y_i y_j - \sum_{(i,j) \in I_2} Q^t_{ij} y_i y_j + \sum_{i \in I_3} c^t_i y_i - \sum_{i \in I_4} c^t_i y_i \right\} \quad (P(x^t, Q^t, c^t))$$

with

$$I_1 := [n]^2 \setminus (\mathcal{U}^Q \cap \mathcal{N}^Q), \quad I_2 := \mathcal{U}^Q \cap \mathcal{N}^Q, \quad I_3 := [n] \setminus (\mathcal{U}^c \cap \mathcal{N}^c), \quad I_4 := \mathcal{U}^c \cap \mathcal{N}^c.$$

Note that the update in Line 6 assigns every unknown coefficient c^t_i, $i \in \mathcal{U}^c$, and Q^t_{ij}, $(i,j) \in \mathcal{U}^Q$, a positive value. Thus, we use the index sets $\mathcal{N}^c$ and $\mathcal{N}^Q$ to equivalently re-introduce the missing negative signs into the objective function. Note that we also re-introduce the negative signs of the objective coefficients for the output of the algorithm in Line 16.

As a final step of each iteration, we compute weights that 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 actual solution $\bar{y}^t$ of $P(x^t, Q^t, c^t)$ and the solution y^t of the follower's problem that can be observed by the leader over time; see Lines 8–15. 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 actually decision of the follower.

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

Theorem 1. *Suppose that Assumptions 1–3 hold. Further, let* $T \geq 4\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|)$ *with* $|\mathcal{U}^c| + |\mathcal{U}^Q| > 1$ *be satisfied. Then, applying Algorithm 1 to Problem (1) leads to*

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

Proof. In order to prove the theorem, we apply Corollary 2.2 from Arora et al. (2012) to Algorithm 1. To this end, we have to show that all assumptions are fulfilled. First, the errors computed in Lines 12 and 13 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 $t \in T$, 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 Line 16. The corresponding unknown coefficients are nonnegative and they sum up to one due to Lines 6 and 15. 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, our computed coefficients c_i^t, Q_{ij}^t for $i \in \mathcal{U}^c, (i,j) \in \mathcal{U}^Q$, and $t \in [T]$ 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 2.2 in Arora et al. (2012) to the nonnegative output (c^t, Q^t), $t \in [T]$, of our algorithm to obtain that

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

holds for all $\eta \in (0, 1/2]$. Furthermore, $|e_{c_i}^t| \leq 1$ holds for all $i \in \mathcal{U}^c$ and $|e_{Q_{ij}}^t| \leq 1$ holds for all $(i,j) \in \mathcal{U}^Q$. We divide the inequality above by T, use Assumption 1, and obtain

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

The right-hand side expression $\eta + \ln(|\mathcal{U}^c| + |\mathcal{U}^Q|)/(\eta T)$ attains its minimum for

$$\eta = \sqrt{\frac{\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|)}{T}} \leq 1/2,$$

which holds due to the assumption $T \geq 4\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|)$. Hence,

$$\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i\in\mathcal{U}^c} c_i^t e_{c_i}^t + \sum_{(i,j)\in\mathcal{U}^Q} Q_{ij}^t e_{Q_{ij}}^t\right) - \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i\in\mathcal{U}^c} |c_i| e_{c_i}^t + \sum_{(i,j)\in\mathcal{U}^Q} |Q_{ij}| e_{Q_{ij}}^t\right)$$
$$\leq 2\sqrt{\frac{\ln(|\mathcal{U}^c| + |\mathcal{U}^Q|)}{T}}$$

holds. Using the definitions of $e_{c_i}^t$ and $e_{Q_{ij}}^t$ (Lines 12 and 13 of Algorithm 1) and additionally multiplying the previous inequality by the constant K, see Assumption 2, leads to

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

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

$$\frac{1}{T}\sum_{t=1}^{T}\Bigg(\sum_{i\in\mathcal{P}^c}(|c_i^t| - |c_i|)(\bar{y}_i^t - y_i^t) + \sum_{i\in\mathcal{N}^c}(|c_i^t| - |c_i|)(y_i^t - \bar{y}_i^t)$$
$$+ \sum_{(i,j)\in\mathcal{P}^Q}(|Q_{ij}^t| - |Q_{ij}|)(\bar{y}_i^t\bar{y}_j^t - y_i^t y_j^t) + \sum_{(i,j)\in\mathcal{N}^Q}(|Q_{ij}^t| - |Q_{ij}|)(y_i^t y_j^t - \bar{y}_i^t\bar{y}_j^t)\Bigg),$$

which is equal to

$$\frac{1}{T}\sum_{t=1}^{T}\Bigg(\sum_{i\in\mathcal{U}^c\cap\mathcal{P}^c}(c_i^t - |c_i|)(\bar{y}_i^t - y_i^t) + \sum_{i\in\mathcal{U}^c\cap\mathcal{N}^c}(c_i^t - |c_i|)(y_i^t - \bar{y}_i^t) \tag{4a}$$
$$+ \sum_{i\in\mathcal{P}^c\cap([n]\setminus\mathcal{U}^c)}(c_i - c_i)(\bar{y}_i^t - y_i^t) + \sum_{i\in\mathcal{N}^c\cap([n]\setminus\mathcal{U}^c)}(|c_i| - |c_i|)(y_i^t - \bar{y}_i^t) \tag{4b}$$
$$+ \sum_{(i,j)\in\mathcal{U}^Q\cap\mathcal{P}^Q}(Q_{ij}^t - |Q_{ij}|)(\bar{y}_i^t\bar{y}_j^t - y_i^t y_j^t) \tag{4c}$$
$$+ \sum_{(i,j)\in\mathcal{U}^Q\cap\mathcal{N}^Q}(Q_{ij}^t - |Q_{ij}|)(y_i^t y_j^t - \bar{y}_i^t\bar{y}_j^t) \tag{4d}$$
$$+ \sum_{(i,j)\in\mathcal{P}^Q\cap([n]^2\setminus\mathcal{U}^Q)}(Q_{ij} - Q_{ij})(\bar{y}_i^t\bar{y}_j^t - y_i^t y_j^t) \tag{4e}$$
$$+ \sum_{(i,j)\in\mathcal{N}^Q\cap([n]^2\setminus\mathcal{U}^Q)}(|Q_{ij}| - |Q_{ij}|)(y_i^t y_j^t - \bar{y}_i^t\bar{y}_j^t)\Bigg), \tag{4f}$$

since the newly added terms in Lines (4b), (4e), and (4f) equal zero. In addition, the nonnegativity of each term follows from the optimality of $\bar{y}^t$ w.r.t. $\bar{c}^t, \bar{Q}^t$, respectively of y^t w.r.t. c, Q. Using the variable mapping of Line 16, we obtain the desired result. □

From the latter theorem, we immediately obtain the following corollary.

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

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

and

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

hold.

We now discuss the shown guarantees in more detail for the considered sequential bilevel setting. The theorem implies that both the objective value error in terms of the actual objective coefficients as well as the error in terms of the guessed objective coefficients is getting small for large observations T. This means that if the leader uses the guess $(\bar{c}^t, \bar{Q}^t)$, on average and for T large enough, to predict a follower's solution $\bar{y}^t$, then the latter is nearly optimal w.r.t. the follower's problem under complete information. Finally, although we do not have a convergence theory for the coefficients, we will later show in our numerical results that they very well match the true coefficients in practice.

4. An Inverse KKT Method

In this section, we focus on learning the objective function of the follower's problem instead of the corresponding objective 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 symmetric and negative semidefinite. Moreover, let (x^t, y^t), $t \in [T]$, be pairs of observations consisting of leader decisions x^t and optimal follower replies y^t.

We now 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 $t \in [T]$.

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

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

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

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

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

If $y^t \in \mathbb{R}^n$ is optimal for Problem (5) given a leader's decision x^t, there exists 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

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

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

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

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

are close to zero. Here, the operator $[\alpha]^- := \max\{0, -\alpha\}$ in (7a) 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 for all $t \in [T]$ 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 respective bound constraints, which is why $r^t_{\text{pos}}(\lambda) = 0$ can be always be achieved by feasibility. Further, we can exploit a priori known information about the objective function by imposing additional constraints. The problem to compute a consistent objective function then reads

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

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

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

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

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

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

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

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

$$Q \preceq 0. \tag{8i}$$

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, $i \in [m], t \in [T]$. 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. In doing so, we fix the known objective coefficients, which we denote by $\hat{Q}_{ij}$, $(i,j) \in [n]^2 \setminus \mathcal{U}^Q$, and $\hat{c}_i$, $i \in [n] \setminus \mathcal{U}^c$. In addition, we impose the a priori known sign of the unknown coefficients. Finally, by the semidefinite-programming constraint (8i), we ensure that the computed objective function is negative semidefinite as the original objective function of the follower's problem (5). Overall, Problem (8) is a semidefinite problem.

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), $t \in [T]$, and the value of the slack of this constraint is known a priori. Using this knowledge we define the index sets of active and inactive constraints

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

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 $t \in [T]$. Introducing these equalities and inequalities as constraints, we obtain $r_{\text{comp}}^t = 0$ for all $t \in [T]$. Consequently, we can equivalently reformulate Problem (8) as

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

Remark 2. *For the special case that the matrix Q of the follower's problem (5) is a diagonal matrix, 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.*

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, after every round $t \in T$, the leader can apply the inverse KKT method to the previously observed optimal decisions $\{y_1, \dots, y_t\}$ of the follower to obtain a consistent objective function. This computed function perfectly explains the observed optimal decisions of the follower from a perspective of inverse optimization. However, consistent objective functions are generally not unique and, thus, the presented inverse KKT approach does not allow for conclusions about the corresponding optimal objective value. In addition, the derivation of the inverse KKT method does not provide any guarantees regarding the error w.r.t. the "true" objective function of the follower. Thus, from a bilevel perspective, it is of special interest how a consistent objective function, computed in round t of the sequential bilevel game, performs in practice if it is used by the leader instead of the unknown objective function of the follower in the following round $t+1$, which we analyze in the following case studies.

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 reason 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 inverse KKT approach, are implemented in Python 3.9.7 and the corresponding optimization problems are solved with Gurobi 9.5.1. 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_100_1, which has 100 items and which has been considered before in Beck et al. (2023a). 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 this instance by relaxing the integrality constraints and considering a quadratic objective function. This leads to a continuous knapsack interdiction problem of the form

$$\min_{x \in [0,1]^n} \quad y^\top Q y + c^\top y \tag{10a}$$

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

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

where $v, w \in \mathbb{R}^n$ are the vectors of leader's and follower's weights, 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

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

For this application, our task is to learn all diagonal entries of Q and all entries of c, i.e., $2n$ coefficients in total. For an algorithmic study of the linear version of this problem we refer to Carvalho et al. (2018).

We note that in the considered application all variables are bounded. Consequently, Assumption 2 is satisfied. Moreover, the follower's objective is a quadratic utility function, in which the quadratic term $\bar{y}^\top Q \bar{y}$ 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 makes Assumption 3 straightforward to satisfy.

5.1.1. *Sampling.* For our numerical experiments, we need a data set of past interactions (x^t, y^t) between the leader and the follower, where y^t is an optimal response of the follower to the leader's decision x^t. The follower computes y^t as a solution to Problem (10c) for all $t \in [T]$, since unlike the leader, he has access to the problem data. For our study, we generate such a data set (x^t, y^t), $t \in [T]$, by first sampling $T = 1000$ upper level decisions x^t on the hyperplane $v^\top x = B$, i.e., we assume that the leader always uses her entire budget. We do so by using the pool search functionality of Gurobi. Second, for every sampled point x^t, we solve the x^t-parameterized lower-level problem to obtain y^t. Due to the fact that the follower's problem is feasible for all sampled x^t, we obtain a data set of 1000 pairs (x^t, y^t), $t \in [T]$. The originally given instance BKIP_100_1 is linear and we use the given coefficient vector c. Based on this vector, we randomly sample diagonal entries for Q between the minimum and maximum entry of c. We then switch the sign of these samples and re-scale c and Q to satisfy the required assumptions. We finally note that the leader usually can directly obtain (x^t, y^t), $t \in [T]$, over time since the leader knows her own decision x^t and can observe the follower's decision y^t.

5.1.2. *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 every sample (x^t, y^t), $t \in [T]$, y^t is optimal w.r.t. the corresponding x^t. Thus, we can apply the inverse KKT method to compute the data c and Q that minimize the violation of the stationarity condition as discussed in Section 4. Using the generated samples, we build and solve Problem (9) for the lower level. To

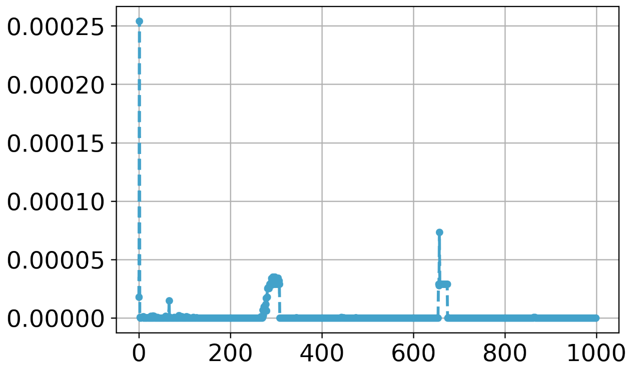

FIGURE 1. Relative consistency measure (11) for the solution of the inverse KKT method when applied to the continuous knapsack interdiction instance BKIP_100_1.

this end, we make use of Remark 2. After solving Model (9), we check whether the objective function determined by the obtained solution c^*, Q^* is consistent with the observed data. To do so, we compute the relative consistency measure

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

where y^t is the observation, and $\bar{y}^t$ is an optimum computed using the solution c^*, Q^*, for the corresponding x^t. For a consistent objective function, the relative consistency measure is zero for all $t \in [T]$. Figure 1 shows that the computed objective function (c^*, Q^*) is able to identify every observation y^t, $t \in [T]$, as a lower-level optimum, i.e., the computed objective function is consistent as the corresponding theoretical results guarantee.

The inverse KKT approach is designed to provide a consistent explanation for the follower's objective function w.r.t. already observed and known solutions of the follower. However, in the considered sequential bilevel game, the leader is interested in anticipating the future follower's objective values and solutions to improve her own decisions. Consequently, the out-of-sample performance of the inverse KKT method is of special interest. To this end, we apply the inverse KKT approach to the first 100 observations (x^t, y^t), $t \in [100]$, to compute a consistent objective function regarding these observed solutions of the follower. Afterward, we use this objective function to predict the follower's objective values and solutions for the remaining 900 leader's decisions x^t. The difference between the objective values and solutions using the imputed objective function and the true objective values and solutions of the follower's problems are illustrated in Figure 2. For doing so, we measure the error regarding the objective values in terms of the MWU error (2). Note that we do not take into account any result within the first 100 iterations since in the inverse KKT approach we have to first observe these data and do not dynamically update the objective coefficients from iteration to iteration as in the MWU approach.

Although we have no theoretical guarantees for the out-of-sample performance, we empirically observe that both the relative errors regarding the follower's objective values, i.e., the MWU error, and the error regarding the solutions of the follower are small; see Figure 2. However, we also observe some clustered occurrences of outliers w.r.t. the objective value and solution error, e.g., around time period 300. This indicates that the training data, i.e., the first 100 observations, do not contain

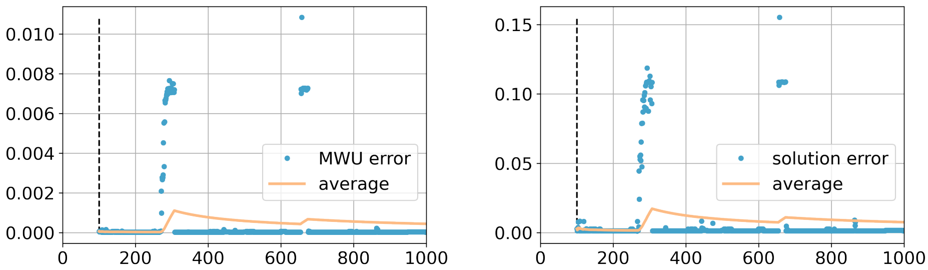

FIGURE 2. Results for the inverse KKT approach imputing the objective function w.r.t. the first 100 data points and then continue with these learned coefficients for the remaining time periods. Left: MWU error (2). Right: Relative error w.r.t. the follower's solution $\|\bar{y}^t - y^t\|_2/\|y^t\|_2$, $t \in [T]$.

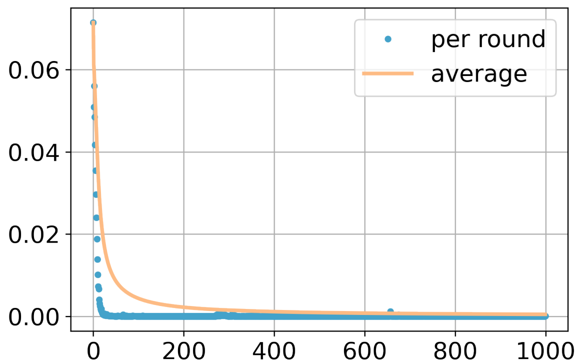

FIGURE 3. MWU error (2), average and per round, for every MWU iteration when applied to the continuous knapsack interdiction instance BKIP_100_1.

all information necessary to perfectly learn the true objective function. To address this in practice, the leader can detect such a situation and re-compute the imputed objective function at a certain point in time when she observes that the predicted and observed solutions differ in too many cases.

5.1.3. *Learning with the MWU Method.* We now apply the MWU method to the generated data set and iteratively update 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 (c^t, Q^t), $t \in [T]$.

Figure 3 shows the MWU error (2) made in every iteration of the MWU method when applied to Model (10) for the instance BKIP_100_1. In line with the theoretical result, the figure illustrates that this error (in average) is decreasing for an increasing number of samples. Consequently, also the error bounds of Corollary 1 converge to zero.

In theory, these results do not extend to the solutions $\bar{y}^t$ and the observations y^t, $t \in [T]$. However, as illustrated in Figure 4 (left), the computed solutions match the observations well over time. In addition, Figure 5 shows a similar development for the computed coefficients c^t, Q^t. Comparing Figure 4 (left) and Figure 5 reveals

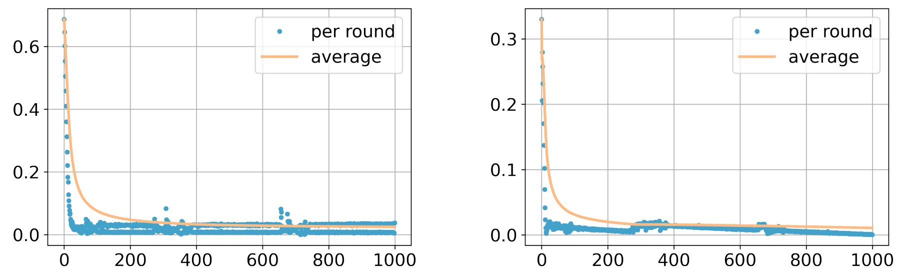

FIGURE 4. Left: Relative error w.r.t. the follower's solution $\|\bar{y}^t - y^t\|_2/\|y^t\|_2$, $t \in [T]$. Right: Relative consistency measure (11). Both per MWU iteration applied to the continuous bilevel knapsack instance BKIP_100_1.

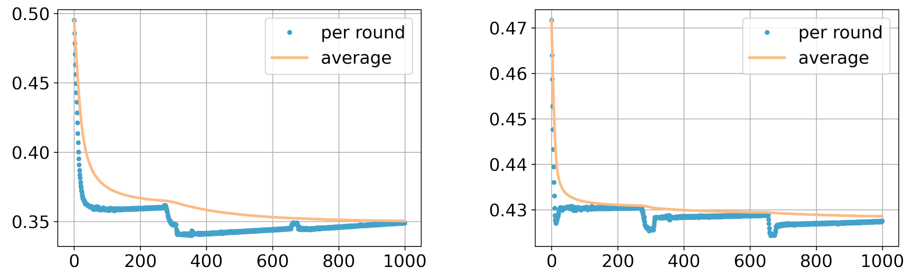

FIGURE 5. Left: $\|c^t - c\|_2/\|c\|_2$, $t \in [T]$. Right: $\|Q^t - Q\|_F/\|Q\|_F$, $t \in [T]$. Both per MWU iteration when applied to the continuous knapsack interdiction instance BKIP_100_1.

that, in practice, we can learn the actual solutions better than the "true" objective coefficients. A possible reason for this might be objective coefficients that belong to items that are rarely used in optimal solutions and, thus, there is little information to precisely learn these coefficients. Further, we empirically check if the computed objective coefficients are consistent w.r.t. observations of the previous rounds in Figure 4 (right). Although it is not theoretically guaranteed, the consistency measure is quite small, but larger than in the inverse KKT approach; see Figure 1.

Finally, we exemplarily analyze the out-of-sample performance of the MWU approach. To this end, we apply the MWU approach to the first 100 observations (x^t, y^t), $t \in [100]$, and then proceed without any further updates of the objective coefficients for the remaining iterations. As illustrated in Figure 6 (left), the leader is able to learn a well-fitting objective function of the follower's problem in terms of the objective function values within the first 100 iterations. More precisely, if the leader chooses the 100th update (c^{100}, Q^{100}) of the MWU method as objective function for the remaining 900 iterations, then this leads to very small values of the MWU error. To a certain extent, this also carries over to learning the follower's solution as we can see in Figure 6 (right). We further compare the out-of-sample performance of the MWU approach with its original version, in which we update the objective coefficients in each round. Note that even in the first 100 iterations, the computed objective coefficients and follower's solutions do not have to be the same in both methods since the corresponding learning rates η differ. In terms of the follower's objective function values, i.e., the MWU error, the performance

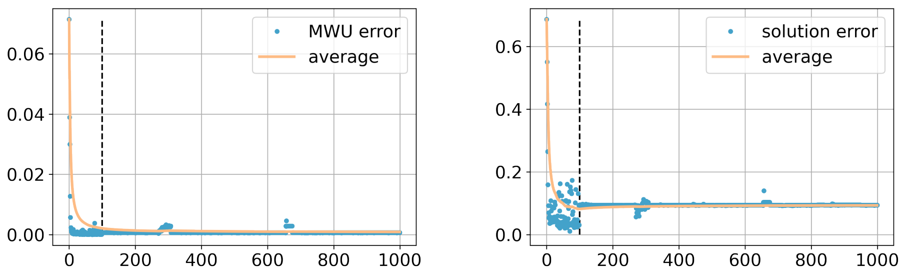

FIGURE 6. Results for the MWU algorithm with the adaption that we stop updating the unknown objective coefficients after iteration $t = 100$ and continue with these learned coefficients, i.e., (c_{100}, Q_{100}) for the remaining iterations. Left: MWU error (2). Right: Relative error w.r.t. the follower's solution $\|\bar{y}^t - y^t\|_2 / \|y^t\|_2$, $t \in [T]$. Both per MWU iteration when applied to the instance BKIP_100_1.

of both variants of the MWU method are comparable, see Figure 3 and Figure 6 (left). In terms of learning the follower's solution, we obtain more accurate results when the follower's objective function is continuously updated over the complete time horizon, which follows from comparing Figure 4 (left) and Figure 6 (right). Concluding, the results regarding the out-of-sample performance indicate that in the considered sequential bilevel setting, the leader does not have to continuously update the learned objective coefficients in each time period in practice. Instead, it suffices to learn the objective coefficients within a specific number of time periods, which in practice can come along with additional costs, e.g., for observing the follower's decision. After this learning period, the leader then can continue with the learned follower's objective function without any further updates if her main interest is on the objective function values. If, however, highly accurate y-estimates are required, further updating of the coefficients may be worth the additional effort.

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 diagonal matrix with negative diagonal entries. Thus, while the leader maximizes her revenue by setting prices x, the follower makes a purchase decision y to maximize his utility (in which we subtract, as usual, costs). Consequently, as in the previous application, Assumptions 1–3 are rather mild and straightforward 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).

For our computations, we randomly sample the entries in the vector c between 0 and 1 and the entries on the main diagonal of Q between -1 and 0. Furthermore, we use 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 food items considered. Thus, the leader sets $n = 78$ prices (x_i) for the 78 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 $y_i^+ \coloneqq \max\{b_j/A_{ji} \colon j \in [m] \text{ with } A_{ji} \neq 0\}$ for $i \in [n]$.

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

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

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

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

Concerning the variable μ,

$$\mu = \sum_{i \in [n]} x_i y_i$$

holds in an optimum. Hence, we also have

$$\mu^- \coloneqq \sum_{i \in [n]} x_i^- y_i^- \leq \mu \leq \sum_{i \in [n]} x_i^+ y_i^+ \eqqcolon \mu^+.$$

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

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

Then, the lower level can be written as

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

with $z^+ \coloneqq (y^+, \mu^+)$ and $z^- \coloneqq (y^-, \mu^-)$.

5.2.1. *Sampling.* As in the previous section, we need the data (x^t, z^t) corresponding to the interactions between the leader and follower, i.e., z^t is an optimal response of the follower, knowing the true objective function, to the leader's decision x^t.

For our study, we generate such a data set by first sampling $T = 1000$ upper level decisions in $[x^-, x^+]$ by using Latin hypercube sampling as implemented in SciPy. Second, for every sampled x^t, we solve the x^t-parameterized lower-level problem to

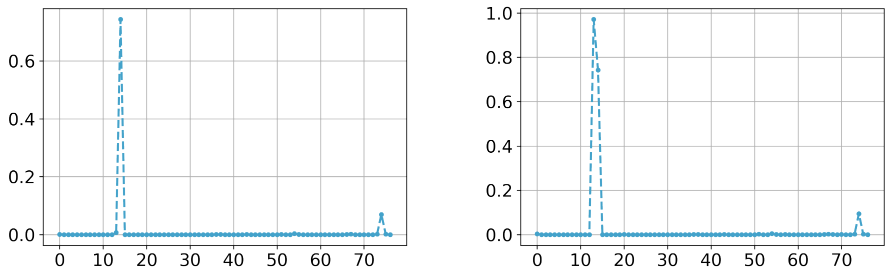

FIGURE 7. Relative deviations between the computed objective coefficients from the true but unknown ones. Left: $|(c_i^* - c_i)/c_i|$, $i \in [n]$. Right: $|(Q_{ii}^* - Q_{ii})/Q_{ii}|$, $i \in [n]$.

obtain a solution z^t. Due to the fact that the follower's problem is feasible for all sampled x^t, we obtain a data set of 1000 pairs (x^t, z^t), $t \in [T]$.

5.2.2. *Learning with the Inverse KKT Method.* Analogously to the previous section, we apply the inverse KKT approach to the bilevel pricing problem and obtain consistent objective coefficients c^* and Q^*. Again, the relative consistency measure (11) is small, i.e., no larger than 10^{-6}. Although the theory only guarantees a consistent objective function as output of the inverse KKT approach, this objective function matches the "true" objective function of the follower's problem quite well with some exceptions, which we illustrate in Figure 7. The outliers in the relative errors correspond to data points with little to no variation. To be more precise, in the generated data set (x^t, y^t), $t \in [1000]$, for the bilevel pricing problem, the items y_i^t linked to outliers are never or only very rarely bought by the follower. As a consequence, the choice of the corresponding objective coefficients has only a minor impact (if at all) on the computed and observed solutions.

We finally analyze the out-of-sample performance of the inverse KKT approach for the pricing application. To this end, we impute an objective function based on the first 100 observations (x^t, y^t), $t \in [100]$. Then, we use this objective function for the remaining iterations. The corresponding errors w.r.t. the objective function values, measured by the MWU error (2), and w.r.t. the solutions of the follower's problem are illustrated in Figure 8. We observe that the out-of-sample performance of the inverse KKT approach is very good, i.e., using only the first 100 observation still allows the leader for learning an objective function that can be used to predict future objective values and decisions of the follower.

5.2.3. *Learning with the MWU Method.* Let (c^t, Q^t), $t \in [T]$, be the output of the MWU method applied to the considered bilevel pricing problem. We again illustrate the MWU error (2) made in every iteration of the MWU method in Figure 9. The general behavior is the same as in the previous section, i.e., as the theory guarantees, the error is decreasing in average. However, we note that the overall error is larger than in the bilevel pricing problem compared to the knapsack interdiction problem.

Analogously to the previous section, we illustrate the relative error of the follower's solution and the relative consistency measure in Figure 10. Both measures converges to zero, but again slower than in the case of the knapsack interdiction problem. In addition, we compare the learned objective coefficients with the "true" coefficients in Figure 11, which shows that we do not perfectly learn the coefficients, but the fit is better compared to the knapsack interdiction case.

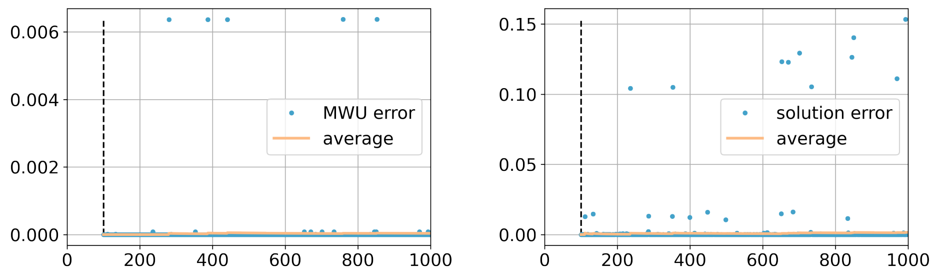

FIGURE 8. Results for the inverse KKT approach imputing the objective function w.r.t. the first 100 data points and then continue with these learned coefficients for the remaining iterations of the pricing instance. Left: MWU error (2). Right: Relative error w.r.t. the follower's solution $\|\bar{y}^t - y^t\|_2/\|y^t\|_2$, $t \in [T]$.

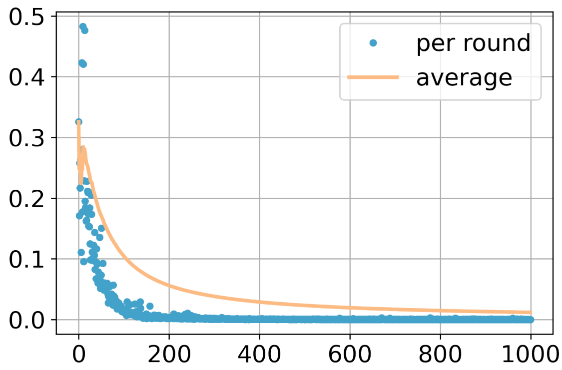

FIGURE 9. MWU error (2), average and per round, per MWU iteration when applied to the bilevel pricing instance.

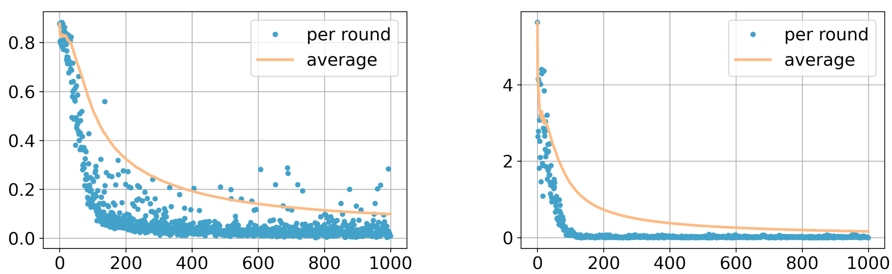

FIGURE 10. Left: Relative error w.r.t. the follower's solution $\|\bar{y}^t - y^t\|_2/\|y^t\|_2$, $t \in [T]$. Right: Relative consistency measure (11). Both per MWU iteration applied to the considered bilevel pricing instance.

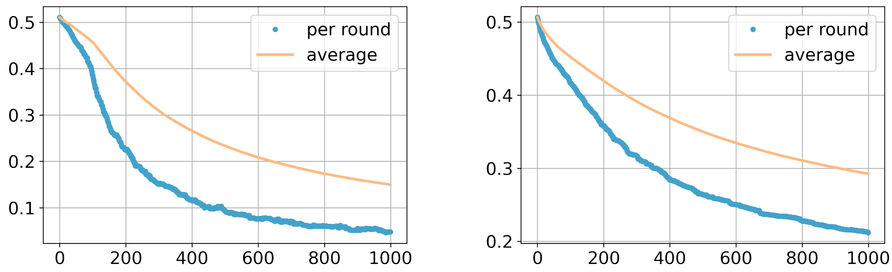

FIGURE 11. Left: $\|c^t - c\|_2/\|c\|_2$, $t \in [T]$. Right: $\|Q^t - Q\|_F/\|Q\|_F$, $t \in [T]$. Both per MWU iteration when applied to the considered bilevel pricing instance.

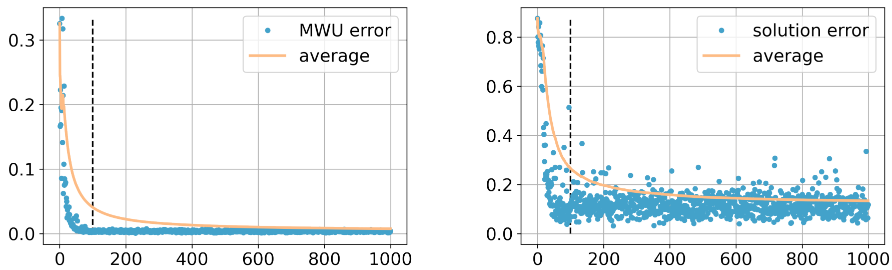

FIGURE 12. Results for the MWU algorithm with the adaption that we stop updating the unknown objective coefficients after iteration $t = 100$ and continue with these learned coefficients, i.e., c_{100} and Q_{100} for the remaining iterations. Left: MWU error (2). Right: Relative error w.r.t. the follower's solution $\|\bar{y}^t - y^t\|_2/\|y^t\|_2$, $t \in [T]$. Both per MWU iteration when applied to the considered bilevel pricing instance.

Finally, we analyze the out-of-sample performance of the MWU approach and its original version, i.e., we only update the learned objective coefficients in the first 100 iterations and then proceed without any further update of these coefficients. In contrast to the bilevel knapsack interdiction problem, we observe in the bilevel pricing problem that using less observations to learn the objective function values of the follower's problem leads to smaller MWU errors and smaller errors w.r.t. the follower's solution at the beginning of the learning process; see Figures 9, 10, and 12. One explanation for this behavior can be that the larger learning rate η in the MWU approach considering only the first 100 iterations leads to less outliers at the beginning of the learning phase. Consequently, the results indicate that for the considered pricing application, the leader can reduce the learning of the objective coefficients to a limited number of time periods without having disadvantages in terms of learning the objective function values and solutions of the follower's problem in the remaining time periods.

Summarizing, we can conclude that both approaches perform well applied to the considered applications. Especially, the good out-of-sample performance of both approaches demonstrates the value of these methods for the leader to learn the objective function (values) and solutions of the follower's problem over time in the considered sequential bilevel games.

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 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.

Several directions for potential future research are possible. First, it would be interesting to see how these approaches work in real-world applications. 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, a hybrid approach based on the MWU and the inverse KKT approach is possible and might lead to better results compared to both approaches used standalone. Fifth and finally, we focused entirely on learning objective functions in this paper while assuming that the follower's feasible set is known. A natural next step would thus be to also learn the follower's constraints.

Acknowledgements

The authors thank the DFG for their support within the research training group 2126 "Algorithmic Optimization".

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.

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.

Beck, Y., I. Ljubić, and M. Schmidt (2023a). "Exact methods for discrete Γ-robust interdiction problems with an application to the bilevel knapsack problem." In: *Mathematical Programming Computation*, pp. 1–50. DOI: 10.1007/s12532-023-00244-6.

Beck, Y., I. Ljubić, and M. Schmidt (2023b). "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.

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 (2023). "Contextual Inverse Optimization: Offline and Online Learning." In: *Operations Research*. Published Online. 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.

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.

Carvalho, M., A. Lodi, and P. Marcotte (2018). "A polynomial algorithm for a continuous bilevel knapsack problem." In: *Operations Research Letters* 46.2, pp. 185–188. DOI: 10.1016/j.orl.2017.12.009.

Clarke, S., G. Dragotto, J. F. Fisac, and B. Stellato (2023). *Learning Rationality in Potential Games*. Tech. rep. URL: https://arxiv.org/abs/2303.11188.

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.

Goerigk, M., J. Kurtz, M. Schmidt, and J. Thürauf (2023). *Connections and Reformulations between Robust and Bilevel Optimization*. Tech. rep. URL: https://opus4.kobv.de/opus4-trr154/frontdoor/index/index/docId/522.

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*. Tech. rep. URL: https://arxiv.org/abs/2102.07367.

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. 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. Pandžić (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, J. Thürauf) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany

Email address: molan@uni-trier.de

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

Email address: johannes.thuerauf@uni-trier.de