Computing Counterfactual Explanations for Linear Optimization: A New Class of Bilevel Models and a Tailored Penalty Alternating Direction Method

Henri Lefebvre, Martin Schmidt

Abstract. Explainable artificial intelligence is one of the most important trends in modern machine-learning research. The idea is to explain the outcome of a model by presenting a certain change in the input of the model so that the outcome changes significantly. In this paper, we study this question for linear optimization problems as an automated decision-making tool. This leads to a new class of linear bilevel optimization problems that have more nonlinearities in their single-level reformulations compared to traditionally studied linear bilevel problems. For this class of problems, we present a tailored penalty alternating direction method and present its convergence theory that mainly ensures that we compute stationary points of the single-level reformulation. Finally, we illustrate the applicability of this method using the example of a real-world energy system model as well as by computing counterfactual explanations for a large set of linear optimization problems from the NETLIB as it has been proposed in the recent literature.

1. Introduction

In today's societies, artificial intelligence (AI) and automated systems are playing an increasingly important role in decision making. To name just a few examples, AI is used for medical diagnosis (Zeng et al. 2016), homeless individual house allocation (Azizi et al. 2018), or credit risk evaluation (Baesens et al. 2003). As these systems continue to have a profound impact on critical decisions, the demand for transparency and explainability is growing. This need becomes even more pronounced when AI is used in social areas. Specifically, it is known that such techniques may lead to unfair outcomes with respect to a subset of individuals in a given dataset (Besse et al. 2021; Miron et al. 2021).

Developing explainable models or methods is not only essential to ensure fairness and accountability in AI, but also for enhancing public trust and providing a foundation for regulation. This need is further emphasized by recent legislative initiatives such as those of the European Commission (EUR-Lex 2021; Goodman and Flaxman 2017) and in the White House Office of Science and Technology Policy (OSTP 2022).

As a result, significant attention has been devoted to the issue of explainability in AI, leading to a new field of research known as explainable artificial intelligence (XAI). For a comprehensive overview of XAI, we refer to Yang et al. (2023). One promising approach to achieve explainability is through the use of counterfactual explanations (CEs); see, e.g., Wachter et al. (2017). Counterfactual explanations offer an intuitive way to understand AI decisions by showing how changes in input data lead to a different, and perhaps more favorable, outcome. Notably, this

Date: December 19, 2024.

2020 *Mathematics Subject Classification.* 90-XX, 90-08,90C05, 90C59, 90C90.

Key words and phrases. Explainability, Linear optimization, Counterfactual explanations, Bilevel optimization, Penalty alternating direction method.

approach does not require technical knowledge of the underlying AI system. Instead, it offers a more accessible and understandable argument for specific decisions.

Mathematically, computing a CE can be formulated as a mathematical optimization problem aiming for the smallest data modification such that a desired outcome is produced by the underlying AI system. For instance, one may ask for the smallest increase in annual salary (all other things being equal) so that a trained classification model approves a loan for a given individual that has been rejected before. Arguably, the concept of CEs has been mainly developed and applied to machine-learning algorithms. For more details, we refer to the recent surveys by Artelt and Hammer (2019) and Verma et al. (2020).

In this paper, we focus on the computation of CEs for automated systems derived from mathematical optimization problems. Specifically, we consider linear optimization problems and seek the minimal change in the constraint matrix of the problem that makes a desired outcome optimal. In such settings, computing a CE can be formulated as a bilevel optimization problem as shown in Korikov et al. (2021) and Kurtz et al. (2024). Bilevel optimization is a field of mathematical optimization in which two optimization problems are intertwined: the upper-level and the lower-level problem. The upper-level problem minimizes a given objective function taking into account the solution to the lower-level problem, which is parameterized by the upper-level's decision. There is a vast literature on bilevel optimization, and we refer to Dempe (2002) for a thorough introduction, as well as to Dempe and Zemkoho (2020) for more recent contributions in the field. Mathematically, a standard linear bilevel problem is given by

$$\begin{aligned} \min_{x,y} \quad & c^\top x + d^\top y \\ \text{s.t.} \quad & Ax + By \geq a, \\ & y \in \arg\min \left\{ f^\top y : Dy \geq b - Cx \right\}. \end{aligned} \tag{BO-RHS}$$

This class of problems is known to be strongly NP-hard (Hansen et al. 1992) and NP-complete (Buchheim 2023). For this reason, heuristic approaches have been developed for (BO-RHS). In particular, the use of metaheuristics in bilevel optimization has recently been surveyed in Camacho-Vallejo et al. (2024) and a matheuristic based on a penalty alternating direction method has been introduced by Kleinert and Schmidt (2021).

In the context of CEs, the lower-level problem represents the underlying automated system (here, a linear optimization problem), while the upper-level problem seeks a minimal change to the input data that leads to a more desirable outcome. In contrast to (BO-RHS), the upper level should therefore be able to modify the constraint matrix D of the lower-level problem; see Kurtz et al. (2024) for a more thorough discussion. This motivates the following new bilevel formulation, which considers matrix modifications in the lower-level problem:

$$\begin{aligned} \min_{x,y} \quad & c^\top x + d^\top y \\ \text{s.t.} \quad & Ax + By \geq a, \\ & y \in \arg\min \left\{ f^\top y : D(x)y \geq b \right\}. \end{aligned} \tag{BO-MAT}$$

Matrix modifications in the lower-level problem introduce a new class of problems, which has only seldomly been studied in the literature. Yet, note the lower level still is a linear optimization problem for a fixed upper-level decision. Hence, standard techniques from bilevel optimization such as the Karush–Kuhn–Tucker (KKT) or the strong-duality reformulation still apply. By using them, one reformulates the bilevel problem (BO-MAT) as a single-level optimization problem using the lower-level optimality conditions. Unfortunately, these approaches lead to a new type of nonlinearity compared to those obtained for (BO-RHS). As a consequence, the reformulated single-level model is considerably harder to solve from a computational point of view. To be more specific, the KKT-based single-level reformulation of (BO-RHS) is a linear problem with complementarity constraints while that of (BO-MAT) is a nonconvex quadratically constrained problem with complementarity constraints.

To the best of our knowledge, the only computational work considering (BO-MAT) is the unpublished short paper by Hajikazemi and Steinke (2024), which is based on approximating products between the upper- and lower-level decisions. However, no convergence analysis is conducted and the approach is only tested on a single academic example.

Contributions and Structure of the Paper. In this paper, we introduce a heuristic approach to compute CEs of linear optimization problems. This approach is inspired by the work of Kleinert and Schmidt (2021), which applies to (BO-RHS). Note that a direct extension to (BO-MAT) is not possible due to the presence of new nonlinearities in the single-level reformulation. To circumvent this fact, we use a different variable split resulting in a different penalty alternating direction method (PADM).

In Section 2, we formally introduce the problem class under consideration and derive a single-level reformulation of the original bilevel problem. Section 3 reviews the framework of PADM for general optimization problems as introduced in Geißler et al. (2017). In Section 4, we apply the PADM to the single-level reformulation derived in Section 2 and discuss its convergence properties. Finally, in Section 5, we report on computational results on two test sets. First, we consider a real-world application arising from the energy sector. Then, we compute CEs of general linear problems taken from the NETLIB (Netlib 2024), as defined in Kurtz et al. (2024).

2. Problem Statement and Reformulations

2.1. Problem Statement.

We start by considering a linear optimization problem with data $\tilde{D} \in \mathbb{R}^{m_y \times n_y}$, $b \in \mathbb{R}^{m_y}$, and $f \in \mathbb{R}^{n_y}$, given by

$$\min_{y \in \mathbb{R}^{n_y}} \quad f^\top y \quad \text{s.t.} \quad \tilde{D} y \geq b. \tag{LP}$$

The objective in what follows is to minimally modify the matrix $\tilde{D}$ so that at least one optimal point to the perturbed problem is within a predefined “favored solution space” $Y \subseteq \mathbb{R}^{n_y}$. Following the terminology of Kurtz et al. (2024), we are therefore interested in finding a weak counterfactual explanation to the question:

> *“Why are the optimal points of* (LP) *not in the favored solution space Y?”*

To answer this question, we look for the smallest modification to the matrix $\tilde{D}$, parameterized by the vector $x \in \mathbb{R}^{n_x}$, so that an optimal point of the perturbed problem

$$\min_{y} \quad f^\top x \quad \text{s.t.} \quad D(x) y \geq b, \tag{LP$_x$}$$

is in Y. Here, $D : \mathbb{R}^{n_x} \to \mathbb{R}^{m_y \times n_y}$ is a given function that maps x to a matrix $D(x)$. For instance, we may assume that $D(\cdot)$ is such that each row is parameterized by x in an affine way. We let $S(x)$ denote the set of optimal points of the x-parameterized problem (LP$_x$). Given a scoring function $J : \mathbb{R}^{n_x} \to \mathbb{R}$, finding a weak counterfactual explanation for (LP) can be stated as the bilevel problem

$$\begin{aligned}
\inf_{x,y} \quad & J(x) && \text{(1a)}\\
\text{s.t.} \quad & x \in X, && \text{(1b)}\\
& y \in Y \cap S(x). && \text{(1c)}
\end{aligned}$$

Here, Constraint (1b) enforces that the modification x to the matrix $\tilde{D}$ belongs to the set of admissible modifications $X \subseteq \mathbb{R}^{n_x}$. Constraint (1c) ensures that there exists an optimal solution to the perturbed problem, which belongs to the favored solution space Y. Finally, the objective (1a) is to minimize the scoring function $J(x)$ of the selected matrix modification x. Note that the feasible region (1b)–(1c) does not need to be closed as shown by Example 3.1 in Kurtz et al. (2024). This holds even if X and Y are compact sets and if S is compact-valued, i.e., if $S(x)$ is compact for all $x \in X$. Hence, we write Problem (1) as an infimum instead of a minimization problem.

We now briefly discuss the generality of Model (1). First, consider the lower-level problem (LP$_x$). Although it is presented here with "$\geq$"-constraints, this is without loss of generality. The lower-level model can readily accomodate "$\leq$"- and "$=$"-constraints by well-known transformations. Additionally, bounds on the decision variables y can be incorporated in the more general constraints. Second, Problem (1) considers only modifications of the left-hand side of Problem (LP). Yet this is also without loss of generality since objective and/or right-hand side changes can easily be incorporated in (1). Clearly, the objective function can be moved to the constraints and right-hand side modifications can be treated by augmenting the matrix D to include an additional column $D_{\cdot, n_y+1} = b$, fixing $y_{n_y+1} = 1$, and setting the right-hand side to zero. Consequently, Problem (1) can represent traditional bilevel problems in which either the constraints' right-hand sides or the objective function is influenced by the upper-level's decision.

2.2. **Single-Level Reformulation.** We now derive a single-level formulation for the bilevel problem (1). To this end, let $\varphi(x)$ denote the value of the x-parameterized Problem (LP$_x$), i.e., let

$$\varphi(x) := \min_{y} \left\{ f^\top y \colon D(x) y \geq b \right\}.$$

It is clear that a point $y \in \mathbb{R}^{n_y}$ belongs to $S(x)$ if and only if $D(x)y \geq b$ and $f^\top y \leq \varphi(x)$ are satisfied. Hence, Problem (1) can be reformulated as

$$\begin{aligned} \inf_{x,y} \quad & J(x) \\ \text{s.t.} \quad & x \in X, \quad y \in Y, \quad D(x)y \geq b, \\ & f^\top y \leq \varphi(x). \end{aligned}$$

This reformulation is well-known and is called the value-function reformulation. Unfortunately, even for simpler bilevel problems with right-hand side modifications, the absence of a closed-form expression for φ often prevents practical implementations of this model. To circumvent this, we resort to duality theory. As mentioned before, the x-parameterized lower-level problem (LP$_x$) is a linear optimization problem. Hence, for any $x \in \mathbb{R}^{n_x}$ such that the lower level is feasible, strong duality implies

$$\varphi(x) = \max_{\lambda \in \mathbb{R}^{m_y}} \left\{ b^\top \lambda \colon D(x)^\top y = f, \lambda \geq 0 \right\}.$$

This leads to the equivalent reformulation

$$\begin{aligned} \inf_{x,y} \quad & J(x) \\ \text{s.t.} \quad & x \in X, \quad y \in Y, \quad D(x)y \geq b, \\ & f^\top y \leq \max_{\lambda} \left\{ b^\top \lambda \colon D(x)^\top \lambda = f, \lambda \geq 0 \right\}, \end{aligned}$$

which, in turn, can be easily shown to be equivalent—on the level of (projected) solutions—to the so-called strong-duality reformulation

$$\begin{aligned} \inf_{x,y,\lambda} \quad & J(x) \\ \text{s.t.} \quad & x \in X, \quad y \in Y, \quad D(x)y \geq b, \\ & f^\top y \leq b^\top \lambda, \\ & D(x)^\top \lambda = c, \quad \lambda \geq 0. \end{aligned} \tag{2}$$

Although Problem (2) is now a single-level optimization problem, it remains challenging to solve due to the bilinear terms involving the variable x and the lower-level's primal and dual variables y and λ. In the following section, we recall and apply the PADM framework to Problem (2) to compute feasible points of good quality.

An alternative approach to derive a single-level reformulation of (1) would be to exploit the KKT optimality conditions of the x-parameterized lower-level problem. However, in doing so, one obtains even more nonlinearities than in the strong-duality reformulation (2). Indeed, the KKT conditions read

$$D(x)y \geq b, \quad D(x)^\top \lambda = f, \quad \lambda \geq 0, \quad \lambda^\top (D(x)y - b) = 0,$$

where we see that the complementarity constraints involve products of three variables; namely, x, y, and λ. Hence, the strong duality reformulation (2) is preferred.

3. The Penalty Alternating Direction Method

In this section, we review the alternating direction method (ADM) and its penalized version (PADM). For more details, we refer to the work of Geißler et al. (2017) and Gorski et al. (2007). To this end, let us consider the general optimization problem

$$\begin{aligned} \min_{x,y} \quad & F(x,y) \\ \text{s.t.} \quad & x \in \mathcal{X}, \quad y \in \mathcal{Y}, \\ & g(x,y) \leq 0. \end{aligned} \tag{3}$$

The ADM is an optimization method in which variables are split in two (or more) disjoint blocks, namely x and y, and which iteratively solves the respective sub-problems over each block. Given a current iterate (x^ℓ, y^ℓ), Problem (3) is solved by fixing variables y to y^ℓ. This leads to the next x-iterate $x^{\ell+1}$. Then, Problem (3) is solved by fixing variables x to $x^{\ell+1}$ to obtain $y^{\ell+1}$. We then repeat these two steps by increasing ℓ. The complete procedure is given in Algorithm 1.

Algorithm 1 Alternating Direction Method (ADM) for Problem (3)

```
1: Given: Initial values $(x^0, y^0) \in \mathcal{X} \times \mathcal{Y}$.
2: for $\ell = 0, 1, \dots$ do
3:     Compute
           $x^{\ell+1} \in \arg\min \left\{F(x, y^\ell) : g(x, y^\ell) \leq 0,\ x \in \mathcal{X}\right\}.$
4:     Compute
           $y^{\ell+1} \in \arg\min \left\{F(x^{\ell+1}, y) : g(x^{\ell+1}, y) \leq 0,\ y \in \mathcal{Y}\right\}.$
5: end for
```

Under mild conditions, it can be shown that Algorithm 1 may converge to a partial minimizer of Problem (3), i.e., to a point (x^*, y^*) such that

$$\begin{aligned} F(x^*, y^*) &\le F(x, y^*) \quad \text{for all } x \in \mathcal{X} \text{ with } g(x, y^*) \le 0,\\ F(x^*, y^*) &\le F(x^*, y) \quad \text{for all } y \in \mathcal{Y} \text{ with } g(x^*, y) \le 0. \end{aligned}$$

The following convergence results are due to Gorski et al. (2007).

Theorem 1 (Theorem 4.9, Gorski et al. (2007)). *Let $\mathcal{X}$ and $\mathcal{Y}$ be closed sets and let $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be continuous and biconvex, i.e., $F(x, \cdot)$ and $F(\cdot, y)$ are convex functions for fixed x and y, respectively. Let each subproblem in Algorithm 1 be solvable.*

(1) If the sequence of iterates $\{(x^\ell, y^\ell)\}_{\ell\in\mathbb{N}}$ of Algorithm 1 is contained in a compact set, then it has at least one accumulation point (x^*, y^*).

(2) If, in addition, all accumulation points (x^*, y^*) are such that one of the two subproblems for $x = x^*$ or $y = y^*$ has a unique solution, then all accumulation points are partial minimizers of (3) and have the same value.

(3) If, in addition, all accumulation points (x^*, y^*) are such that both of the subproblems for $x = x^*$ or $y = y^*$ have a unique solution, then $(x^\ell, y^\ell) \to (x^*, y^*)$ for $\ell \to \infty$.

Corollary 1 (Corollary 4.10, Gorski et al. (2007)). *Let the assumptions of Theorem 1, Case (3) hold, and suppose that F is differentiable. Then, all accumulation points (x^*, y^*) which lie in the interior of $\mathcal{X} \times \mathcal{Y}$ are stationary points of (3).*

The main idea of the ADM is that each subproblem will be (much) easier to solve than the original problem (3). Indeed, under the biconvex assumption of Theorem 1, both subproblems are convex while Problem (3) is not. In practice, it can be observed that further decoupling of the blocks x and y leads to better performance; see, e.g., Boyd (2010). Hence, we now consider a variant of Problem (3) in which the coupling constraints "$g(x, y) \le 0$" are penalized by a weighted ℓ_1 penalty function, i.e., we consider

$$\min_{x\in\mathcal{X}, y\in\mathcal{Y}} \quad F(x, y) + \sum_{i=1}^{m} \mu_i [g_i(x, y)]^+. \tag{4}$$

Here, $[u]^+ = \max\{0, u\}$ and $\mu \in \mathbb{R}^m_{\ge 0}$ is a vector of penalty parameters. We let $\phi(\cdot, \cdot; \mu)$ denote the objective function of (4).

At every iteration, the PADM solves Problem (4) using the ADM as described in Algorithm (1). If a partial minimizer of Problem (4) is found by the ADM, we check if the coupling constraints are satisfied. If so, the PADM terminates. Otherwise, the penalty parameters μ are increased and the process is repeated. The full procedure is given in Algorithm 2.

The following convergence result is due to Geißler et al. (2017).

Theorem 2. *Let the assumptions of Theorem 1 hold and assume $\mu_i^k \nearrow \infty$ for all $i = 1, \dots, m$. Moreover, let $\{(x^k, y^k)\}_{k\in\mathbb{N}}$ be a sequence of partial minimizers of (4) for $\mu = \mu^k$ generated by the inner loop of Algorithm 2 with $(x^k, y^k) \to (x^*, y^*)$. Then, there exists $\bar\mu \in \mathbb{R}^m_{\ge 0}$ such that (x^*, y^*) is a partial minimizer of the weighted feasibility measure*

$$\chi_{\bar\mu}(x, y) = \sum_{i=1}^{m} \bar\mu_i [g(x, y)]^+.$$

If, in addition, (x^*, y^*) is feasible for the original problem (3), the following holds:

(1) If F is continuous, then (x^*, y^*) is a partial minimizer of (3).

(2) If F is continuously differentiable, then (x^*, y^*) is a stationary point of (3).

Algorithm 2 Penalty Alternating Directioon Method (PADM) for Problem (3)

```
Given: Initial values $(x^{0,0}, y^{0,0}) \in \mathcal{X} \times \mathcal{Y}$ and $\mu^0 \in \mathbb{R}^m_{\geq 0}$.
for $k = 0, 1, \dots$ do
    Set $\ell \leftarrow 0$.
    while $(x^{k,\ell}, y^{k,\ell})$ is not a partial minimizer of (4) with $\mu = \mu^k$ do
        Compute
            $x^{k,\ell+1} \in \arg\min \left\{ \phi(x, y^{k,\ell}; \mu^k) : x \in \mathcal{X} \right\}.$
        Compute
            $y^{k,\ell+1} \in \arg\min \left\{ \phi(x^{k,\ell+1}, y; \mu^k) : y \in \mathcal{Y} \right\}.$
        Increase $\ell \leftarrow \ell + 1$.
    end while
    Choose new penalty parameters $\mu^{k+1} \geq \mu^k$.
end for
```

(3) If F is continuously differentiable and convex and the feasible region of (3) is convex, then (x^*, y^*) is a global minimizer of (3).

4. Computing Counterfactual Explanations Using the PADM

In this section, we apply the PADM to the strong-duality reformulation (2) of Problem (1) to avoid the bilinearities between x, y, and λ. To this end, we introduce two vectors of penalty parameters, $\rho \in \mathbb{R}^{m_y}_{\geq 0}$ and $\mu \in \mathbb{R}^{n_y}_{\geq 0}$, to penalize the primal and dual feasibility constraints, respectively. The reason for penalizing these constraints is twofold: (i) They are the only nonconvex constraints, and (ii) they are the only constraints coupling x, on the one hand, and y and λ, on the other hand. The penalized problem reads

$$\min_{x,y,\lambda} \quad J(x) + \sum_{i=1}^{m_y} \rho_i \left[b_i - d_{i\cdot}(x) y\right]^+ + \sum_{j=1}^{n_y} \mu_j \left| f_j - d_{\cdot j}(x)^\top \lambda \right| \tag{5a}$$

$$\text{s.t.} \quad x \in X, \tag{5b}$$

$$y \in Y, \quad \lambda \geq 0, \quad f^\top y \leq b^\top \lambda. \tag{5c}$$

Now, observe that the penalized problem (5) has a feasible region which is polyhedral if X and Y are, themselves, polyhedral. Hence, the feasible region of (5) is "as nice as it can be" from a computational perspective. However, the objective function remains nonconvex and nonsmooth. Nevertheless, Problem (5) becomes a convex albeit nonsmooth optimization problem for x being fixed. The same holds true for fixed y and λ if J is a convex function. Thus, a natural splitting of variables is to consider the blocks (y, λ) and x. More specifically, we identify the sets $\mathcal{X}$ and $\mathcal{Y}$ from Section 3 as

$$x \in \mathcal{X} \quad \longleftrightarrow \quad x \text{ satisfies (5b)},$$
$$y \in \mathcal{Y} \quad \longleftrightarrow \quad (y, \lambda) \text{ satisfies (5c)}.$$

We highlight here that this splitting of variables differs from the one suggested by Kleinert and Schmidt (2021) in the context of right-hand side modifications. In the latter, the authors consider the upper- and lower-level primal variables x and y on the one hand, and the lower-level dual variables λ on the other hand. However, such a split would not eliminate the bilinearities between x and y in the current setting. Hence, the (x, y)-subproblem would then be intractable, which we avoid by using a different block decomposition.

Assuming that x is fixed to some value $\hat{x}$, the (y, λ)-subproblem reads

$$\begin{aligned} \min_{y,\lambda} \quad & \sum_{i=1}^{m_y} \rho_i \left[b_i - d_{i\cdot}(\hat{x})y\right]^+ + \sum_{j=1}^{n_y} \mu_j \left|f_j - d_{\cdot j}(\hat{x})^\top \lambda\right| \\ \text{s.t.} \quad & (y, \lambda) \in Y \times \mathbb{R}^{m_y}_{\geq 0}, \quad f^\top y \leq b^\top \lambda. \end{aligned}$$

Essentially, this problem minimizes the (weighted) primal and dual infeasibility by choosing appropriate values for y and λ. Let $(\hat{y}, \hat{\lambda})$ denote one of its optimal points. It is clear that an optimal value of 0 with strictly positive weights implies that $\hat{y} \in S(\hat{x})$ holds and $(\hat{x}, \hat{y})$ is a feasible point of Problem (1). Otherwise, the x-subproblem

$$\min_{x \in X} \quad J(x) + \sum_{i=1}^{m_y} \rho_i \left[b_i - d_{i\cdot}(x)\hat{y}\right]^+ + \sum_{j=1}^{n_y} \mu_j \left|f_j - d_{\cdot j}(x)^\top \hat{\lambda}\right|,$$

is solved, which produces a new candidate for x.

We note that, although the two subproblems in x and (y, λ) are nonsmooth convex problems, they can be smoothed by representing the penalty functions $|\cdot|$ and $[\cdot]^+$ using linear constraints (in a higher dimensional space).

We now discuss some convergence results, which are directly obtained from Theorem 1 and 2. We start with the inner loop of Algorithm 2 applied to the strong-duality reformulation (2).

Theorem 3. *Let X and Y be closed sets and let J be a continuous and convex function. Consider the kth inner loop of Algorithm 2 applied to Problem (2) for fixed penalty parameters $\mu \in \mathbb{R}^{m_y}_{\geq 0}$ and $\rho \in \mathbb{R}^{n_y}_{\geq 0}$ and let $\{(x^{k,\ell}, y^{k,\ell}, \lambda^{k,\ell})\}_{\ell \in \mathbb{N}}$ be the generated sequence of iterates. Let each subproblem be solvable.*

(1) If the sequence of iterates $\{(x^{k,\ell}, y^{k,\ell}, \lambda^{k,\ell})\}_{\ell \in \mathbb{N}}$ is contained in a compact set, then it has at least one accumulation point $(x^{k*}, y^{k*}, \lambda^{k*})$.

(2) If, in addition, all accumulation points $(x^{k*}, y^{k*}, \lambda^{k*})$ are such that one of the two subproblems has a unique solution, then all accumulation points are partial minimizers of the penalized problem (2).

(3) If, in addition, all accumulation points $(x^{k*}, y^{k*}, \lambda^{k*})$ are such that both subproblems have a unique solution, then $(x^{k,\ell}, y^{k,\ell}, \lambda^{k,\ell}) \to (x^{k*}, y^{k*}, \lambda^{k*})$.

We now discuss the assumptions of Theorem 3. First, note that closedness of X and Y are not strong assumptions in practice. In fact, further assuming that X and Y are compact is mild and ensures that each subproblem is solvable so that all conditions in Theorem 3 are satisfied. Ensuring that the sequence of iterates $\{(x^{k,\ell}, y^{k,\ell}, \lambda^{k,\ell})\}_{i \in \mathbb{N}}$ is contained in a compact set is more challenging, even if X and Y are compact. Still, note that all iterates satisfy

$$(x^{k,\ell}, y^{k,\ell}, \lambda^{k,\ell}) \in \operatorname{lev}_{\alpha^k}(\phi) \text{ with } \alpha^k = \phi(x^{k,0}, y^{k,0}, \lambda^{k,0}; \rho^k, \mu^k),$$

where ϕ denotes the objective function of (5) and $\operatorname{lev}_\alpha(\phi)$ denotes the sub-level set of the function ϕ over the feasible region of (2). Hence, it is enough to ensure that $\operatorname{lev}_{\alpha^k}(\phi)$ is compact to obtain the first statement of Theorem 3. One way to achieve this is to add a strictly convex regularizer to the objective function in (5). By doing so, one would also enforce that the (y, λ)-subproblem always has a unique solution, and, therefore, that all accumulation points are partial minimizers of the regularized penalized problem (5).

We are now ready to state the convergence result regarding the PADM applied to Problem (2).

Theorem 4. *Let X and Y be closed sets and let J be a continuous and convex function and assume $\rho_i^k \nearrow \infty$ and $\mu_j^k \nearrow \infty$ for all $i = 1, \dots, m_y$ and all $j =$*

$1, \dots, n_y$. *Moreover, let* $\{(x^k, y^k, \lambda^k)\}_{k \in \mathbb{N}}$ *be a sequence of partial minimizers of* (2) *for* $\mu = \mu^k$ *and* $\rho = \rho^k$ *generated by the inner loop of Algorithm* 2 *with* $(x^k, y^k, \lambda^k) \to (x^*, y^*, \lambda^*)$. *Then, there exist* $\bar{\rho}$ *and* $\bar{\mu}$ *such that* (x^*, y^*, λ^*) *is a partial minimizer of the weighted primal-dual feasibility measure*

$$\chi_{\bar{\rho},\bar{\mu}} := \sum_{i=1}^{m_y} \bar{\rho}_i [b_i - d_{i\cdot}(x)y]^+ + \sum_{j=1}^{n_y} \bar{\mu}_j |f_j - d_{\cdot j}(x)^\top \lambda|.$$

If, in addition, (x^*, y^*, λ^*) *is feasible for the strong-duality reformulation* (2)*, then* (x^*, y^*, λ^*) *is a partial minimizer of* (2)*, and* (x^*, y^*) *is feasible for the original bilevel problem* (1)*. If, in addition,* J *is continuously differentiable, then* (x^*, y^*, λ^*) *is a stationary point of* (2)*.*

We close this section with a brief discussion of this theorem. In particular, note that the theorem only regards stationary points of the strong-duality reformulation (2) and not about stationary points of the original bilevel problem (1). Nevertheless, if (x^*, y^*, λ^*) is feasible for Problem 2, then (x^*, y^*) is feasible for the bilevel problem (1). Hence, the algorithm computes a CE. Moreover, we highlight that J being continuously differentiable can be weakened so that J is only required to be continuous and closed, i.e., that its epigraph is closed. If this is the case, one can always reformulate Problem (2) by moving the objective function to the constraints. By Theorem 2.2 in Stein (2024), the two formulations are completely equivalent.

5. Numerical Results

In this section, we apply the PADM presented in Section 4 to two sets of instances. After discussing some implementation details, we first study a real-world application stemming from the energy sector. Then, we consider the general instances derived from the NETLIB as considered in Kurtz et al. (2024).

5.1. Implementation.

Our implementation was done in C++ using the idol library (Lefebvre 2023) with all underlying optimization problems solved by Gurobi version 10.0.1. In addition to the standard scheme presented in Algorithm 2, several practical modifications were implemented to improve numerical stability and computational efficiency. Note that these modifications do not impact the convergence theory presented in Section 4. The following subsections detail these modifications.

5.1.1. *Update Rules and Rescaling of Penalty Parameters.*

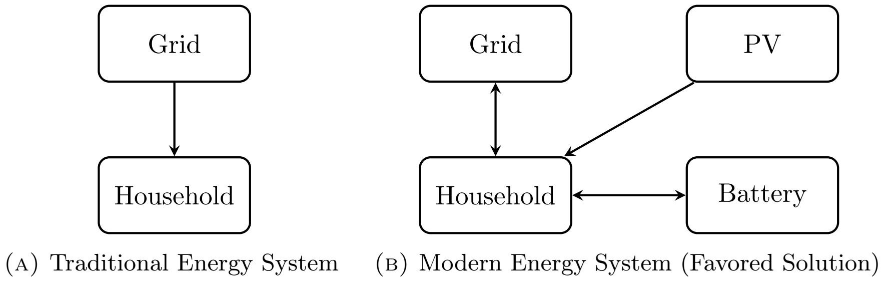
In Line 9 of Algorithm 2, the penalty parameters are updated. However, when the penalty parameters get too large, numerical instabilities may arise. To avoid this, we perform a systematic rescaling of the penalty parameters (ρ, μ) as soon as $\|(\rho, \mu)\|_\infty$ exceeds the threshold of 10^9. This rescaling uses a sigmoid function, denoted by σ, as suggested by Schewe et al. (2020). Formally, we perform the rescaling as follows:

$$\begin{aligned} \rho_i &\leftarrow \sigma(\rho_i, (\rho, \mu)), \quad i = 1, \dots, m_y, \\ \mu_j &\leftarrow \sigma(\mu_j, (\rho, \mu)), \quad j = 1, \dots, n_y, \end{aligned}$$

where σ is defined by

$$\sigma(u_i, u) := 5 \times \left(\frac{u_i - \|u\|_\infty}{\|u\|_\infty + |u_i - \|u\|_\infty|} + 1 \right).$$

Initially, a straightforward penalty update rule is used, which doubles every penalty parameter corresponding to a violated constraint. Yet, while rescaling prevents numerical instabilities, it can sometimes introduce undesired cycles in which the penalty parameters repeat after rescaling. To prevent this, we check



(A) Traditional Energy System (B) Modern Energy System (Favored Solution)

Figure 1. Traditional and Modern Energy Systems

for cycling at every rescaling operation. Any time a cycle is detected, the penalty parameter update rule is changed.

More formally, let $Q := \{i : [b_i - (a_{i\cdot} + \delta_{i\cdot})x]^+ > 0\}$ and let $R := \{j : |f_j - x^\top(a_{\cdot j} - \delta_{\cdot j})| > 0\}$ be the set of indices of primal and dual constraints that are violated. Then, the following update rules are used:

Rule 1: $\rho_i \leftarrow 2 \times \rho_i$ for $i \in Q$, $\mu_j \leftarrow 2 \times \mu_j$ for $j \in R$;
Rule 2: $\rho_i \leftarrow 1.5 \times \rho_i$ for $i \in Q$, $\mu_j \leftarrow 1.5 \times \mu_j$ for $j \in R$;
Rule 3: $\rho_i \leftarrow \rho_i(1 + \rho_i/\|(\rho,\mu)\|_\infty)$ for $i \in Q$, $\mu_j \leftarrow \mu_j(1 + \mu_j/\|(\rho,\mu)\|_\infty)$ for $j \in R$;
Rule 4: $\rho_i \leftarrow \rho_i + 500$ for $i \in Q$, $\mu_j \leftarrow \mu_j + 500$ for $j \in R$.

Note again that the update rule only changes if a cycle is detected.

5.1.2. *Initialization, Warmstart, and Restart.* All penalty parameters have an initial value of 500. Unfortunately, Algorithm 2 may occasionally converge to a stationary point of Problem (5) that is infeasible for Problem (1). To address this, we therefore check for feasibility improvements throughout the execution. If the feasibility error does not improve over 1000 iterations, the algorithm is restarted with initial penalty parameters set to 1/500. If the situation repeats, the algorithm stops and reports a failure status.

Algorithm 2 requires an initial value for x. We initially set it to 0. Alternatively, we implemented a "warmstart" initialization as follows. First, we apply Algorithm 2 to Problem (5) with $J = 0$. If a feasible point $(\bar{x}, \bar{y}, \bar{\lambda})$ is found, we use $\bar{x}$ as a starting point for another run of Algorithm 2 using the true objective function J.

5.2. **Modern Household Energy System: A Case Study.** In this section we apply the concept of CEs to a model for the optimal operation of a household that manages its own energy production via photovoltaic (PV) panels and a battery. To this end, we consider a dwelling having a certain predicted electricity demand. In traditional settings, this demand is met by purchasing power from the grid. However, in today's more decentralized energy systems, states or their regulatory authorities try to incentivize consumers to consider the integration of PV panels and a battery in their household's energy system. In our setup, we then also consider the possibility to sell energy back to the grid, as it occurs in modern energy systems. Figure 1 depicts the traditional and the favored design.

The goal of this case study is to show how CEs can be used to identify parameters that need to be adjusted to make the integration of PV panels advantageous for the household. This is a setup of high practical importance as it shows that CEs can give guidance to regulatory authorities w.r.t. what needs to be changed so that a desired change in the energy system is achieved.

The rest of this section is organized as follows. Section 5.2.1 presents an LP to optimize the energy system of a given household and Section 5.2.2 then discusses the optimal solution obtained using real-world data. In Section 5.2.3, we finally use

CEs to compute small changes of the parameters of the model so that a desired solution is obtained, which is not the case in the unmodified setup.

5.2.1. *The Underlying LP.* We use an LP model based on Hülsmann et al. (2023) to optimize the energy system of a given household. This model uses a finite and discretized time horizon $\{1, \dots, T\}$. At every time step $t \in \{1, \dots, T\}$, the predicted energy demand d_t as well as prices for buying and selling energy, c_t^+ and c_t^-, are known. The level of exploitable sunshine for the PV panel is called ρ_t and is assumed to be given. This parameter also includes the efficiency of the PV module. Furthermore, we let c_{PV} and c_{B} denote the investment costs for the PV panel and the battery, respectively.

The main decision variables are the following. We use z_t^+ and z_t^- to denote the amount of energy bought and sold at time t while q_{PV} and q_{B} refer to the capacity of the PV panel and the battery. To keep track of the energy flow in the model, we also introduce the following variables. The energy produced by the PV panel at time t is denoted by y_t and s_t denotes the amount of energy stored in the battery at time t. Finally, we use u_t^+ and u_t^- to model the flow of energy leaving and entering the battery.

With this notation at hand, we can state the overall model, which reads

$$\min_{y,z,u,s,q} \quad c_{\text{PV}} q_{\text{PV}} + c_{\text{B}} q_{\text{B}} + \sum_{t=1}^{T} \left(c_t^+ z_t^+ - c_t^- z_t^- \right) \tag{6a}$$

$$\text{s.t.} \quad z_t^+ + y_t + u_t^+ = z_t^- + d_t + u_t^-, \quad t = 1, \dots, T, \tag{6b}$$

$$s_t = s_{t-1} + u_t^- - u_t^+, \quad t = 1, \dots, T, \tag{6c}$$

$$y_t \le \rho_t q_{\text{PV}}, \quad t = 1, \dots, T, \tag{6d}$$

$$s_t \le q_{\text{B}}, \quad t = 1, \dots, T, \tag{6e}$$

$$z_t^+, z_t^-, u_t^+, u_t^-, y_t, s_t, q_{\text{PV}}, q_b \ge 0, \quad t = 1, \dots, T, \tag{6f}$$

$$q_{\text{PV}}, q_{\text{B}} \ge 0. \tag{6g}$$

Constraints (6b) enforce the energy balance in the overall system: At any point in time, the energy z_t^+ bought from the grid, the energy y_t produced by the PV, and the energy u_t^+ taken from the battery is equal to the energy z_t^- sold to the grid, the energy demand d_t, and the energy u_t^- used to charge the battery. Constraints (6c) model the storage level of the battery. The storage level s_t at time t is given by the storage level of the previous time step and the (dis)charge $u_t^{\pm}$ of the battery. By convention, we use $s_0 = s_T$ to obtain a periodic battery profile. Constraints (6d) links the PV capacity q_{PV} to the PV panels power output y_t. Constraints (6e) enforces that the battery capacity q_{B} is not exceeded. Finally, the objective function (6a) minimizes the investment and operational costs over the entire time horizon.

5.2.2. *Traditional Solution.* For the numerical study, we consider an instance of Model (6) with a time horizon of one week discretized into hours. To this end, we use the following data:

Energy Demand: Demand profiles are taken from the Open Power System Data website (https://data.open-power-system-data.org/household_data/2020-04-15) and correspond to a usual residential building.

Exploitable Sunshine Ratios: The levels of exploitable sunshine for the PV panels are taken from the simulation tool accessible at https://www.renewables.ninja/ and are associated to the city of Darmstadt in Germany.

Energy Prices: The prices for buying and selling energy are assumed to be constant and set to 0.25 €/kWh and 0.05 €/kWh.

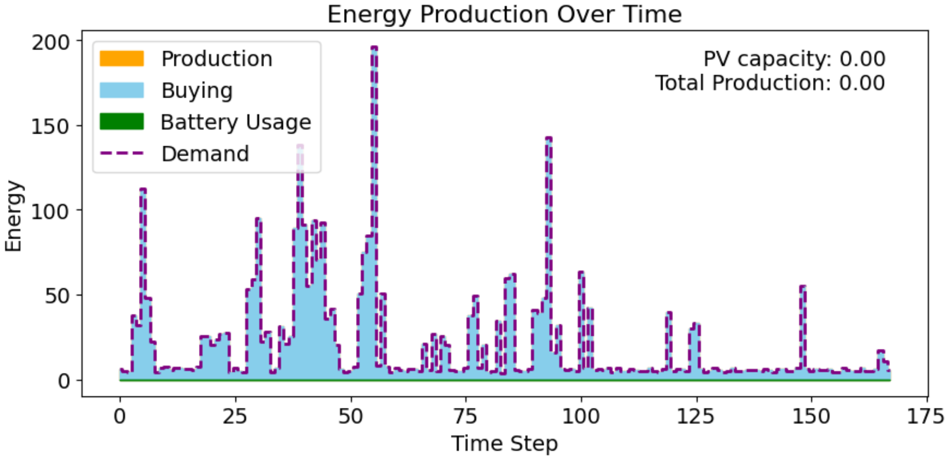


FIGURE 2. Traditional solution with an amortized cost for PV panel cells of 34 €/kWh.

PV Panel Prices: The cost of photovoltaic cells per kWh is assumed to be 36140 €/kWh. Note that this price is intentionally high and is in line with the average price in 1980; see, e.g., https://ourworldindata.org/grapher/solar-pv-prices. We assume a life time of 20 years. Hence, c_{PV} is set to

$$\frac{36140€/\mathrm{kWh}}{20\text{ years} \times 365\text{ days}} \times 7\text{ days} \approx 34\ €/\mathrm{kWh}.$$

Battery Costs: The cost for the battery is assumed to be 500 €/kWh with a life time of 5 years. Hence, c_{B} is set to

$$\frac{500\ €/\mathrm{kWh}}{5\text{ years} \times 365\text{ days}} \times 7\text{ days} \approx 2\ €/\mathrm{kWh}.$$

Figure 2 illustrates an optimal solution to Model (6). The dotted curve represents the household's electricity demand over the week, while the shaded areas indicate the sources of energy supply. The blue area represents energy bought from the grid, which satisfies the entire demand in this solution. The absence of orange (PV panel production) and green (battery contribution) areas indicates that no investment is made in PV panels or battery installation at all.

This shows that, given the high amortized costs of PV panels (34 €/kWh) and batteries (2 €/kWh), it is more advantageous to satisfy the demand by solely buying energy from the grid.

5.2.3. *Counterfactual Explanation.* In this section, we use the PADM presented in Section 4 to answer the following question:

> *Given that users should be incentivized to invest in PV panels, how should the situation change from an economical or technological viewpoint so that it becomes advantageous to do so?*

In particular, we will focus on the economical parameters. More precisely, we ask for a minimal change in the PV and battery investment costs q_{PV} and q_{B} so that a user invests in a battery of at least 20 kWh and installs a PV panel large enough to produce at least 1000 kWh of energy during one week. To this end, we introduce two variables x_{PV} and x_{B} used to modify the investment costs in the underlying

Model (6). The bilevel formulation reads

$$\min_{x,y,z,u,s,q} \quad |x_{\mathrm{PV}}| + |x_{\mathrm{B}}| \tag{7a}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} y_t \geq 1000, \quad q_{\mathrm{B}} \geq 20, \tag{7b}$$

$$(y, z, u, s, q) \in S(x_{\mathrm{PV}}, x_{\mathrm{B}}), \tag{7c}$$

where $S(x_{\mathrm{PV}}, x_{\mathrm{B}})$ denotes the set of solutions to the problem

$$\min_{y,z,u,s,q} \quad (c_{\mathrm{PV}} + x_{\mathrm{PV}})q_{\mathrm{PV}} + (c_{\mathrm{B}} + x_{\mathrm{B}})q_{\mathrm{B}} + \sum_{t=1}^{T} \left(c_t^+ x_t^+ - c_t^- x_t^- \right) \quad \text{s.t.} \quad \text{(6b)–(6g)}.$$

Here, Constraints (7b) models the favored solution space. Note further that we do not modify the matrix of the originally given LP but its objective function, which is a special case of the more general setup studied in the last sections.

Applying the PADM from Section 4 to Model (7) leads to an upper-level solution $(x_{\mathrm{PV}}, x_{\mathrm{B}}) \approx (-23, 0)$, corresponding to a new price for PV panels of 11 €/kWh. In particular, it is not required to modify all parameters to achieve the overall goal as $x_{\mathrm{B}} = 0$. Figure 3 illustrates an optimal solution if the prices are changed accordingly. Under this scenario, it can be seen that a PV panel of capacity 18.59 kWh is installed and used to produce a total of 1000 kWh during the week. It can also be seen that it becomes advantageous for the user to invest in a battery of capacity 22.13 kWh, and to store the excess of energy produced during the day for when the level of exploitable sunshine becomes too low.

Hence, we have shown that the PADM presented in Section 4 can be used for computing counterfactual explanations for a simple energy system model. By doing so, we compute a necessary change of investment prices so that users are incentivized to invest in PV panels and battery storage. Hence, this case study shows that counterfactual explanations cannot only be used for explainability in AI, but also help practitioners in designing systems so that a more favorable output is accepted by its users.

5.3. **NETLIB.** In this section, we analyze the performance of the PADM on instances taken from the literature. We consider the set of instances introduced in Kurtz et al. (2024), which are based on the NETLIB instances (Netlib 2024). These instances are generated as follows. For each NETLIB instance, we randomly select 1, 5, or 10 columns associated with non-negative variables to be impacted by matrix modification. In doing so, we make sure that columns which are selected in a smaller group (e.g., 5-mutable-columns) are also part of larger groups (e.g., 10-mutable-columns). Given a selected column, only fractional coefficients are considered mutable. Instances with only integer coefficients are dealt with in a different way: coefficients whose absolute value is larger than 10 and not a multiple of 10 are considered mutable. Every mutable coefficient is allowed for a maximum deviation of ±100 %. The favored solution space X is generated from an optimal point y^* of the unperturbed problem. Three variables are randomly selected. Then, for a selected variable y_j, we add the constraint $y_j \geq 1.05 y_j^*$ to the favored space if this does not violate the variable's upper bound, otherwise we add $y_j \leq 0.95 y_j^*$. If $y_j^* = 0$, we add $y_j \geq 0.05$ or $y_j \leq -0.05$ if the former leads to conflicting bounds. We highlight that all instances are publicly available in the GitHub repository https://github.com/JannisKu/CE4LOPT. In Table 1, we recall the classification of NETLIB instances in terms of size as described in Kurtz et al. (2024).

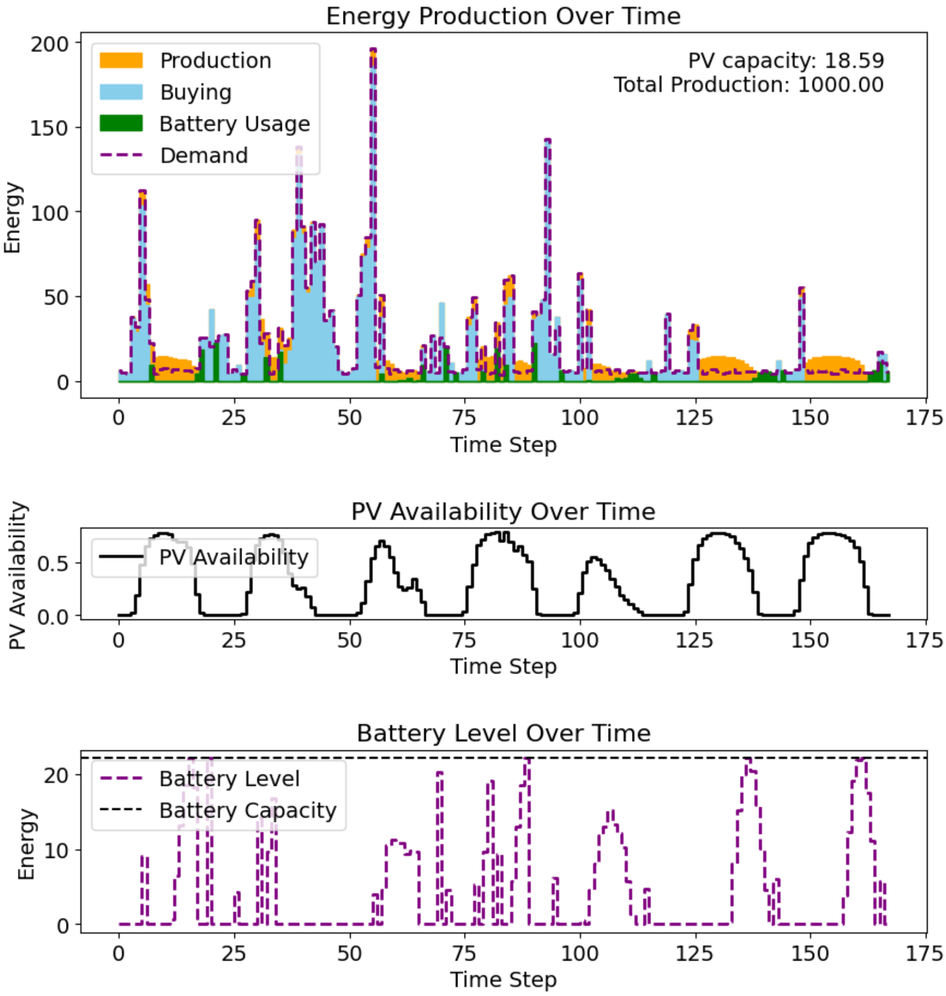


FIGURE 3. Solution for an amortized cost for PV panel cells of 11 €/kWh.

All our experiments were run on an Intel XEON SP 6126 (2.60 GHz) on a single core and with 4 GB of RAM. In total, 5760 instances were tested with a time limit of 30 minutes each.

We first compare our results with those obtained in Kurtz et al. (2024). Note, however, that the authors do not consider the same type of CEs. Indeed, they consider relative CEs, which are defined as a minimal change to the constraint matrix so that at least one point in the desired space becomes feasible for the modified problem while, at the same time, having a similar objective value as the unperturbed problem. This is in contrast to the weak CEs considered in this paper defined as a minimal change to the constraint matrix so that at least one solution of the modified problem belongs to the desired space. The reason for this is that computing optimal weak CEs was shown by Kurtz et al. (2024) to be out of reach of state-of-the-art solvers in reasonable time. This was also the main motivation for considering a heuristic approach for weak CE in the current work. Hence, while Kurtz et al. (2024) consider a global optimization approach for a relaxed notion of CEs, the PADM presented in Section 4 is a heuristic approach for a stronger type of CEs. The main differences between these two works are summarized in Table 2.

Type	Category	Intervals
# Variables	small	$0 \leq n \leq 534$
	medium	$534 \leq n \leq 2167$
	large	$2167 \leq n \leq 22275$
# Constraints	small	$0 \leq m \leq 351$
	medium	$351 \leq m \leq 906$
	large	$906 \leq m \leq 16675$

Table 1. Categorization of NETLIB instances according to (Kurtz et al. 2024). Note that each category is defined by closed intervals so that some instances may belong to two categories. Nevertheless, we keep these definitions to compare our results with those reported in Kurtz et al. (2024).

		Quality of the Solution	
Contribution	CE Type	Upper Level	Lower Level
Kurtz et al. (2024)	Relative	Optimal	Feasible
This paper	Weak	Stationary*	Optimal

*Stationary point of the strong-duality reformulation.

Table 2. Comparison of the solution quality with the literatrue.

In Table 3, we report the percentage of instances for which a feasible relative CE was found by Kurtz et al. (2024), and the percentage of instances for which a feasible weak CE was found by the PADM. Note that there are (at least) two reasons for which no feasible point is found by the PADM. First, it may be the case that the problem is actually infeasible. Second, the PADM may have terminated with a stationary point, which is not feasible for the original problem while feasible points exist. Admittedly, a drawback of the PADM is that infeasibility cannot be proven. Nevertheless, it can be observed that the number of feasible relative and weak CEs, which are computed by Kurtz et al. (2024) and ourselves are comparable. This is an interesting result that shows the strength of the PADM because weak CEs are a much stronger notion than relative CEs.

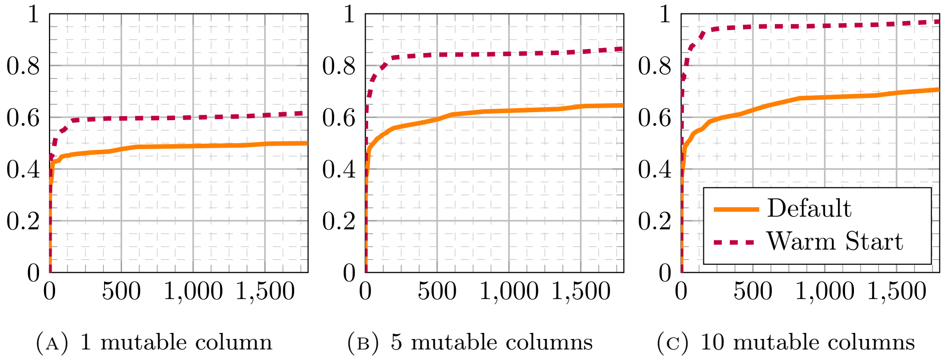
Regarding computation times, we present in Figure 4 the empirical cumulative distribution of runtimes depending on the number of mutable columns. The solid curve is associated to the default version of PADM while the dotted curve corresponds to the PADM with warmstart. For a better readability, we consider only those instances for which at least one of the two methods (with and without warmstart) solve at least one version of the instance (with 1, 5, or 10 mutable columns). Hence, all three plots in Figure 4 refer to the same set of instances with an increasing number of mutable columns. First, we can see that increasing the number of mutable columns increases the number of feasible weak CEs that we can compute. This is to be expected since any point that is feasible with one mutable column remains feasible if the number of mutable columns is increased. However, it can also be seen that the number of feasible CEs with 10 mutable columns does not represent 100 % of the set of instances. This unfortunately shows that the PADM occasionally terminated with an infeasible point even though a feasible point could be computed when solving that instance with 1 or 5 mutable columns.

Figure 4 also shows that warmstart has a significant impact on the PADM runtime. For instance, the default PADM computes a feasible CE for around 70 %

n	m	# mut. columns	# inst.	# mutable objective param.	# mutable constraint param.	% feasible relative CE ($\alpha = 1$)	% feasible weak CE found
small	small	1	28	0.30	4.38	**38.00**	32.32
		5	28	1.55	21.18	**54.00**	50.00
		10	28	4.06	54.96	**59.00**	57.50
	medium	1	7	0.79	5.09	**36.00**	30.00
		5	7	4.11	31.64	**61.00**	57.14
		10	7	10.86	79.63	64.00	**65.00**
medium	small	1	4	0.61	5.92	**84.00**	51.25
		5	4	2.59	22.57	**100.00**	80.00
		10	4	6.75	54.13	**100.00**	90.00
	medium	1	22	0.48	10.42	35.00	**36.59**
		5	22	2.46	19.01	**51.00**	46.59
		10	22	6.39	35.56	**58.00**	51.36
	large	1	8	0.53	2.85	31.00	**32.50**
		5	8	2.05	10.68	**55.00**	50.00
		10	8	5.45	26.21	**63.00**	59.38
large	small	1	2	0.43	12.43	**25.00**	**25.00**
		5	2	0.93	74.13	**48.00**	47.50
		10	2	2.03	199.40	**50.00**	**50.00**
	medium	1	6	0.53	2.25	**43.00**	26.67
		5	6	2.76	8.97	**49.00**	35.83
		10	6	7.33	22.02	**53.00**	44.17
	large	1	21	0.55	5.13	45.00	**45.71**
		5	21	2.37	20.79	**58.00**	53.33
		10	21	6.07	53.09	**65.00**	55.71

TABLE 3. Number of computed CEs in the relative sense with $\alpha = 1$ (Kurtz et al. 2024) and in the weak sense (this paper) for the NETLIB instances. The second last column is directly taken from Kurtz et al. (2024).

of the instances with 10 mutable columns while the PADM with warmstart is able to compute one for more that 95 % of the same instances within the time limit. Moreover, it can be seen that the PADM with warmstart finds a feasible CE for most of the instances in less than 300 seconds, including the time needed to compute the starting point. Finally, Figure 5 presents a boxplot of the ℓ_1-norm of the computed CEs over those instances for which both the PADM with and without warmstart could compute a feasible CE. It can be seen that using a warmstart typically leads to larger CEs (measured in the ℓ_1 norm) than the default version. This, again, is to be expected given that the starting point is computed by disregarding the original objective function and by solely focusing on feasibility.



(A) 1 mutable column (B) 5 mutable columns (C) 10 mutable columns

Figure 4. Empirical cumulative distribution of computation times with $J = \|\cdot\|_1$.

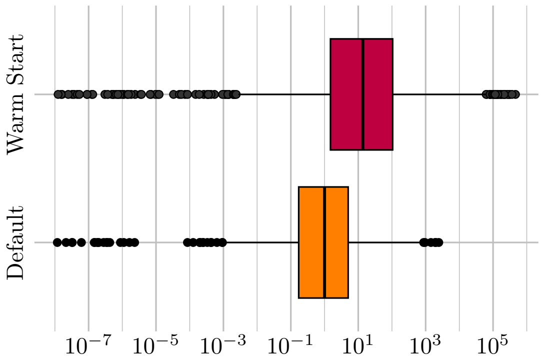


Figure 5. Boxplot of the ℓ_1-norm of the computed CE

6. Conclusion

Counterfactual explanations are important tools to increase the explainability of machine learning models or other automated systems that are based on mathematical optimization. In this paper, we show that computing CEs for linear optimization problems leads to a new and highly challenging class of linear bilevel optimization problems. We presented a tailored penalty alternating direction method to compute feasible points of this problem quickly and applied it to two different fields in order to highlight its applicability. First, we analyzed CEs for energy system models and, second, computed CEs for general linear optimization problems from the NETLIB.

There are many interesting open topics left for future research. First, one could think about designing tailored algorithms that solve the new class of bilevel problems to global optimality. The method presented in this paper can then be used, e.g., as a primal heuristic to speed of the solution process. Second, existence theory for the new class of problems is not yet settled. Third, it might be interesting to also consider the pessimistic variant of the bilevel problem under consideration, which would lead to an approach of computing so-called strong CEs as it is defined in Kurtz et al. (2024).

Acknowledgements

We acknowledge the support by the German Bundesministerium für Bildung und Forschung within the project “RODES” (Förderkennzeichen 05M22UTB). The computations were executed on the high performance cluster “Elwetritsch” at the TU Kaiserslautern, which is part of the “Alliance of High Performance Computing Rheinland-Pfalz” (AHRP). We kindly acknowledge the support of RHRK. We are very grateful for all our project partners in the “RODES” project—in particular, we want to thank Sina Hajikazemi and Florian Steinke for providing us with the data used for the numerical results in Section 5.2. Last but not least, we would like to thank Jannis Kurtz for providing more detailed results on the NETLIB instances used in Section 5.3.

References

Artelt, A. and B. Hammer (2019). *On the computation of counterfactual explanations – A survey.* URL: https://arxiv.org/abs/1911.07749.

Azizi, M. J., P. Vayanos, B. Wilder, E. Rice, and M. Tambe (2018). “Designing Fair, Efficient, and Interpretable Policies for Prioritizing Homeless Youth for Housing Resources.” In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research.* Springer International Publishing, pp. 35–51. DOI: 10.1007/978-3-319-93031-2_3.

Baesens, B., R. Setiono, C. Mues, and J. Vanthienen (2003). “Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation.” In: *Management Science* 49.3, pp. 312–329. DOI: 10.1287/mnsc.49.3.312.12739.

Besse, P., E. del Barrio, P. Gordaliza, J.-M. Loubes, and L. Risser (2021). “A Survey of Bias in Machine Learning Through the Prism of Statistical Parity.” In: *The American Statistician* 76.2, pp. 188–198. DOI: 10.1080/00031305.2021.1952897.

Boyd, S. (2010). “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.” In: *Foundations and Trends® in Machine Learning* 3.1, pp. 1–122. DOI: 10.1561/2200000016.

Buchheim, C. (2023). “Bilevel linear optimization belongs to NP and admits polynomial-size KKT-based reformulations.” In: *Operations Research Letters* 51.6, pp. 618–622. DOI: 10.1016/j.orl.2023.10.006.

Camacho-Vallejo, J.-F., C. Corpus, and J. G. Villegas (2024). “Metaheuristics for bilevel optimization: A comprehensive review.” In: *Computers & Operations Research* 161, p. 106410. DOI: 10.1016/j.cor.2023.106410.

Dempe, S. (2002). *Foundations of Bilevel Programming.* Nonconvex Optimization and Its Applications. Springer US. DOI: 10.1007/b101970.

Dempe, S. and A. Zemkoho (2020). *Bilevel Optimization: Advances and Next Challenges.* Springer International Publishing. DOI: 10.1007/978-3-030-52119-6.

EUR-Lex (2021). *Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts.* URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206.

Geißler, B., A. Morsi, L. Schewe, and M. Schmidt (2017). “Penalty Alternating Direction Methods for Mixed-Integer Optimization: A New View on Feasibility Pumps.” In: *SIAM Journal on Optimization* 27.3, pp. 1611–1636. DOI: 10.1137/16m1069687.

Goodman, B. and S. Flaxman (2017). “European Union Regulations on Algorithmic Decision Making and a “Right to Explanation”.” In: *AI Magazine* 38.3, pp. 50–57. DOI: 10.1609/aimag.v38i3.2741.

Gorski, J., F. Pfeuffer, and K. Klamroth (2007). “Biconvex sets and optimization with biconvex functions: a survey and extensions.” In: *Mathematical Methods of Operations Research* 66.3, pp. 373–407. DOI: 10.1007/s00186-007-0161-1.

Hajikazemi, S. and F. Steinke (2024). *Solving bilevel problems with products of upper- and lower-level variables.* arXiv: 2409.03619 [math.OC].

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.

Hülsmann, J., J. Barbosa, and F. Steinke (2023). “Local Interpretable Explanations of Energy System Designs.” In: *Energies* 16.5. DOI: 10.3390/en16052161.

Kleinert, T. and M. Schmidt (2021). “Computing Feasible Points of Bilevel Problems with a Penalty Alternating Direction Method.” In: *INFORMS Journal on Computing* 33.1, pp. 198–215. DOI: 10.1287/ijoc.2019.0945.

Korikov, A., A. Shleyfman, and C. Beck (2021). “Counterfactual Explanations for Optimization-Based Decisions in the Context of the GDPR.” In: *ICAPS 2021 Workshop on Explainable AI Planning.* URL: https://openreview.net/forum?id=YiR1NIojU2q.

Kurtz, J., S. I. Birbil, and D. d. Hertog (2024). *Counterfactual Explanations for Linear Optimization.* URL: 10.48550/ARXIV.2405.15431.

Lefebvre, H. (2023). *idol: Generic decomposition methods for mathematical programming.* publicly available online. URL: https://hlefebvr.github.io/idol/ (visited on 09/18/2023).

Miron, M., S. Tolan, E. Gómez, and C. Castillo (2021). “Evaluating causes of algorithmic bias in juvenile criminal recidivism.” In: *Artificial Intelligence and Law* 29.2, pp. 111–147. DOI: 10.1007/s10506-020-09268-y.

Netlib (2024). *Netlib Repository.* https://www.netlib.org/. Last accessed: December 19, 2024.

OSTP (2022). *Blueprint for an AI Bill of Rights: Making Automated Systems Work for the American People.* The White House Office of Science and Technology Policy (OSTP). URL: https://www.whitehouse.gov/ostp/ai-bill-of-rights/.

Schewe, L., M. Schmidt, and D. Weninger (2020). “A decomposition heuristic for mixed-integer supply chain problems.” In: *Operations Research Letters* 48.3, pp. 225–232. DOI: 10.1016/j.orl.2020.02.006.

Stein, O. (2024). *A tutorial on properties of the epigraph reformulation.* Published on Optimization Online. URL: https://optimization-online.org/?p=28203.

Verma, S., J. P. Dickerson, and K. Hines (2020). “Counterfactual Explanations for Machine Learning: A Review.” In: *CoRR.* URL: https://ml-retrospectives.github.io/neurips2020/camera_ready/5.pdf.

Wachter, S., B. Mittelstadt, and C. Russell (2017). “Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR.” In: *Harvard Journal of Law & Technology* 31, p. 841. URL: https://arxiv.org/abs/1711.00399.

Yang, W., Y. Wei, H. Wei, Y. Chen, G. Huang, X. Li, R. Li, N. Yao, X. Wang, X. Gu, M. B. Amin, and B. Kang (2023). “Survey on Explainable AI: From Approaches, Limitations and Applications Aspects.” In: *Human-Centric Intelligent Systems* 3.3, pp. 161–188. DOI: 10.1007/s44230-023-00038-y.

Zeng, J., B. Ustun, and C. Rudin (2016). “Interpretable Classification Models for Recidivism Prediction.” In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 180.3, pp. 689–722. DOI: 10.1111/rssa.12227.

(Henri Lefebvre, Martin Schmidt) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany

Email address: henri.lefebvre@uni-trier.de

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