A sensitivity-based method for bilevel optimization problems: Theoretical analysis and computational performance

Downloaded from: https://research.chalmers.se, 2026-09-14 08:34 UTC

Citation for the original published paper (version of record):
Nolasco, E., King, R., Vassiliadis, V. (2026). A sensitivity-based method for bilevel optimization problems: Theoretical analysis and
computational performance. Computers and Chemical Engineering, 213.
http://dx.doi.org/10.1016/j.compchemeng.2026.109709

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Computers and Chemical Engineering 213 (2026) 109709

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

A sensitivity-based method for bilevel optimization problems: Theoretical analysis and computational performance

Eduardo Nolasco [a,*], Ross D. King [a,b], Vassilios S. Vassiliadis [c]

[a] Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge, United Kingdom
[b] Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden
[c] Independent Academic Consultant, Retired from the University of Cambridge, Larnaca, Cyprus

ARTICLE INFO

Keywords:
Bilevel optimization
Sensitivity analysis
Augmented Lagrangian method
Adjoint method
Strong stationarity
Mathematical program with complementarity constraints

ABSTRACT

Bilevel optimization provides a powerful framework for modeling hierarchical decision-making systems. This work presents a sensitivity-based algorithm that addresses the bilevel structure directly by treating the lower-level optimal solution as an implicit, locally differentiable function of the upper-level variables, thereby avoiding classical single-level reformulations. Under standard regularity assumptions on the lower level, an adjoint-based representation of the reduced upper-level gradient is derived, replacing explicit construction of the sensitivity Jacobian with a single linear adjoint solve per iteration and reducing gradient evaluation cost by a factor equal to the upper-level dimension. The reduced problem is solved within an Augmented Lagrangian framework, with inner subproblems managed by an L-BFGS-B quasi-Newton solver. Convergence to KKT points of the reduced problem is established, and these points are shown to be equivalent to S-stationary solutions of the associated mathematical program with complementarity constraints under MPEC-LICQ. Computational experiments on benchmark bilevel problems validate the method's correctness and robustness, and demonstrate the effectiveness of a pragmatic dual-criterion stopping condition in handling the asymmetric primal–dual convergence rates characteristic of augmented Lagrangian methods.

1. Introduction

Bilevel optimization problems (BLPs) are mathematical programs in which one optimization problem is constrained by the solution of a subordinate optimization problem. This hierarchical structure defines two levels: an upper-level (or leader) problem, whose decisions influence the feasible set or objective of a lower-level (or follower) problem. The solution to the lower-level problem, in turn, feeds back into the upper-level decision-making process, creating a coupled dependency between the two levels.

The origin of BLPs can be traced back to leader-follower games introduced by von Stackelberg (1934) within the economic context. They were later introduced to the operation research community by Bracken and McGill (1973) as optimization problems with an optimization problem in their constraints, and have since found widespread application in multiple disciplines. In chemical engineering, notable examples include the optimal design of processes involving thermodynamic equilibrium (Clark and Westerberg, 1990), parameter estimation in phase equilibrium problems (Mitsos et al., 2009), capacity planning (Garcia-Herreros et al., 2016), supply chain management (Yue and You, 2014), among others.

Despite their practical relevance, BLPs pose significant computational challenges. Their feasible region is often discontinuous and non-differentiable, rendering the overall problem nonconvex, even when each level is convex. In addition, there exist multiple, non-equivalent formulations of BLPs, which complicate the derivation of general optimality conditions. A common approach is to reformulate the problem as a single-level optimization problem. However, such reformulations are not always faithful: specific regularity and structural assumptions must be satisfied to preserve equivalence with the original bilevel structure (Dempe et al., 2015). Classical approaches include KKT-based reformulations leading to mathematical programs with complementarity constraints, exact global methods based on bounding schemes, multiparametric programming for problems with linear or quadratic lower levels, and data-driven surrogate approaches; each carries significant limitations in the general smooth nonlinear setting considered here, as reviewed in Section 2.

This work proposes a sensitivity-based descent algorithm for solving deterministic, continuous bilevel optimization problems in which the lower-level problem is convex. The method leverages parametric sensitivity analysis to treat the lower-level optimal solution as an implicit,

* Corresponding author.
E-mail addresses: en307@cam.ac.uk, eduardo.nolasco@cantab.net (E. Nolasco).

https://doi.org/10.1016/j.compchemeng.2026.109709
Received 1 October 2025; Received in revised form 9 April 2026; Accepted 13 May 2026
Available online 19 May 2026
0098-1354/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

locally differentiable function of the upper-level variables, enabling the construction of descent directions for the upper-level objective without recourse to classical single-level reformulations. Gradient evaluation is performed via an adjoint system that avoids explicit formation of the sensitivity Jacobian, reducing the per-iteration gradient cost to a single linear solve independent of the upper-level dimension. The reduced problem is solved within a robust Augmented Lagrangian framework, where inner subproblems are managed by an L-BFGS-B quasi-Newton solver with a strong Wolfe line search. Convergence to KKT points of the reduced problem is established under standard regularity assumptions, and these points are shown to be equivalent to S-stationary solutions of the associated MPCC reformulation under MPEC-LICQ. A pragmatic dual-criterion stopping condition is introduced to address the asymmetric primal–dual convergence rates characteristic of augmented Lagrangian methods.

The remainder of the paper is structured as follows. Section 2 discusses alternative formulations of BLPs and reviews classical solution approaches, including KKT-based reformulations, value-function methods, exact global bounding schemes, multiparametric programming, and data-driven surrogate approaches. Section 3 introduces the proposed sensitivity-based method, including the adjoint-based gradient formulation, the Augmented Lagrangian algorithm, and the convergence and S-stationarity equivalence results. In Section 4 presents the computational experiments, covering implementation details, illustrative examples, and benchmark results on the BOLIB library. Section 5 concludes the paper and outlines directions for future research.

2. BLP formulation and classical solution approaches

2.1. Bilevel optimization problems formulation

The general, yet inherently ambiguous, formulation of a BLP can be expressed as:

$$\begin{aligned} &\min_{x\in X} \; F(x,y) \\ &\text{s.t. } G(x,y)\le 0, \\ &\quad y\in \operatorname*{argmin}_{y\in Y}\{\, f(x,y) \mid g(x,y)\le 0\,\}, \end{aligned} \tag{2.1}$$

where $x \in X \subset \mathbb{R}^n$ denotes the upper-level variables, $y \in Y \subset \mathbb{R}^m$ denotes the lower-level variables. The functions $F, f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ define the upper- and lower-level objectives, while $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^r$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^s$ denote the vector-valued inequality constraints functions at the upper and lower-level, respectively. To simplify the notation and exposition, equality constraints are omitted in this work noting that they can be represented by pairs of inequalities. Throughout this work, uppercase letters denote upper-level functions, while lowercase letters denote lower-level functions.

Despite its apparent simplicity, the BLP (2.1) is not well-posed whenever the lower-level problem admits multiple optimal solutions. Consider the illustrative example due to Lucchetti et al. (1987) whose upper-level objective is $F(x,y) = x^2 + y^2$ and lower-level problem is $\operatorname{argmin}_y\{-xy \mid 0 \le y \le 1\}$; if the upper-level decision is $x = 0$, then any $y \in [0,1]$ is optimal for the lower-level, thus rendering the BLP ill-defined.

To appropriately formulate the BLP, we first introduce the *parametric* lower-level problem:

$$\begin{aligned} &\min_{y\in Y} \; f(x,y) \\ &\text{s.t. } g(x,y)\le 0, \end{aligned} \tag{2.2}$$

for a given value of the upper-level variable x. The solution set of this problem defines a set-valued mapping $\Psi : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ given by

$$\Psi(x) := \operatorname*{argmin}_{y\in Y}\{\, f(x,y) \mid g(x,y)\le 0\,\}. \tag{2.3}$$

If the upper-level can influence the lower-level's decision when multiple solutions exist, then one obtains the *optimistic* bilevel optimization problem:

$$\begin{aligned} &\min_{x,y} \; F(x,y) \\ &\text{s.t. } G(x,y)\le 0, \\ &\quad x\in X, \\ &\quad y\in \Psi(x). \end{aligned} \tag{2.4}$$

It is important to note that the operator $\min_{x,y}$ in this context does not imply a simultaneous optimization over the variables x and y. Instead, the constraint $y \in \Psi(x)$ dictates a sequential process where, for a given x, the upper-level selects a specific y from the set $\Psi(x)$ that is most favorable to its own objective. The sets X and Y are typically compact sets defined by box constraints on the variables which are treated as inequality constraints and included in G and g.

If the leader cannot influence the lower-level decision making process, further assumptions or hierarchical selection rules must be imposed leading to the *pessimistic* formulation of BLPs (Dempe, 2002; Wiesemann et al., 2013). Throughout this work, we consider the optimistic case and assume that $\Psi(x)$ is single-valued, ensuring that the BLP (2.4) is well defined.

Note that, by definition, the mapping Ψ in (2.3) denotes the set of *global minimizers* of the parametric lower-level problem (2.2). Thus, any algorithm for solving bilevel problems must ensure global optimality at the lower-level. If local minimizers or stationary points of the lower-level problem are admitted, then the solution of the resulting relaxed problem will generally differ from that of the original problem (Mirrlees, 1999).

2.2. Classical solution methods

Bilevel optimization problems are intrinsically difficult to analyze and solve. In particular, optimality conditions based on classical nonlinear programming concepts (stationarity, constraints qualifications or duality) are not readily available for the bilevel case. Therefore, the usual approach to solve the BLP (2.4) is to reformulate as a single-level optimization problem.

One such reformulations replaces the lower-level problem (2.2) with its Karush-Kuhn–Tucker (KKT) optimality conditions, which are then included as constraints in the upper-level problem. If the functions f, g are differentiable and convex, and if a suitable constraint qualification holds for all x at $y \in \Psi(x)$, then the bilevel problem (2.4) can be reformulated as its *KKT transformation*:

$$\begin{aligned} &\min_{x,y,\lambda} \; F(x,y) \\ &\text{s.t. } G(x,y)\le 0 \\ &\quad \nabla_y f(x,y) + \lambda^\top \nabla_y g(x,y) = 0, \\ &\quad g(x,y)\le 0 \\ &\quad 0\le \lambda,\ \lambda^\top g(x,y) = 0 \\ &\quad y\in Y,\ x\in X. \end{aligned} \tag{2.5}$$

Problem (2.5) is a *mathematical program with complementarity constraints (MPCC)*. Both bilevel optimization problems and MPCCs are special cases of the broader class of *mathematical programs with equilibrium constraints (MPECs)* (Kočvara and Outrata, 2004). Due to the complementarity constraint, MPCCs violate standard constraint qualifications at any feasible point, which makes the derivation of optimality conditions a challenging task. To address these challenges, several generalized *stationarity concepts* have been developed within the MPEC framework (Outrata, 1990; Scheel and Scholtes, 2000). The strongest among these is *Strong stationarity (S-stationarity)*. A feasible point of the MPCC reformulation (2.5) is called S-stationary if its standard KKT conditions are satisfied. This means there exist Lagrange multipliers such that the gradient of the MPEC Lagrangian is zero, and the multipliers

associated with the inequality constraints are all non-negative. An S-stationary point is a highly desirable solution, as it is the most rigorous of the MPCC stationarity conditions. This MPCC reformulation has been studied in Gümüş and Floudas (2001).

Another classical reformulation of the bilevel problem (2.4) is the *optimal value function* reformulation, originally introduced by Outrata (1990). In this formulation, the optimal value function associated with the lower-level problem (2.2) is defined as

$$\varphi(x) = \min_{y \in Y} \{ f(x,y) \mid g(x,y) \leq 0 \}, \tag{2.6}$$

and is subsequently introduced as a constraint in the upper-level problem. By combining this with the feasible set of the lower-level problem, the bilevel problem is reformulated as the following single-level problem:

$$\begin{aligned} \min_{x,y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x) \leq 0 \\ & f(x,y) \leq \varphi(x) \\ & g(x,y) \leq 0 \\ & y \in T \\ & x \in X. \end{aligned} \tag{2.7}$$

Problem (2.7) is fully equivalent to the original bilevel problem (2.4) both in terms of local and global solutions (Dempe et al., 2015). However, direct solution of (2.7) is computationally intractable in general, since $\varphi(x)$ is implicit, nonsmooth, and nonconvex, and no closed-form expression is available. This has motivated two distinct strategies in the global optimization literature, both of which use $\varphi(x)$ as a conceptual object without computing it explicitly.

The first strategy preserves the bilevel structure and constructs convergent bounds on $\varphi(x)$ through nested bounding problems. Kleniati and Adjiman (2014) proposed a branch-and-sandwich algorithm that maintains separate upper- and lower-level bounding problems, sandwiching $\varphi(x)$ from above and below without collapsing the problem to the single-level form (2.7). This was later extended to mixed-integer bilevel problems (Kleniati and Adjiman, 2015).

The second strategy avoids explicit reference to $\varphi(x)$ altogether, instead constructing convergent approximations of the lower-level optimality constraint through discretization-based bounding schemes. Mitsos et al. (2008) introduced a bounding algorithm for the global solution of continuous bilevel programs with nonconvex lower levels. The approach constructs convergent lower and upper bounds by solving a relaxed single-level program augmented with parametric upper bounds on the lower-level optimal value, guaranteeing ε-optimality without branching. This framework was subsequently extended to handle mixed-integer variables (Mitsos, 2010), and later adapted to accommodate lower-level equality constraints (Djelassi et al., 2019), which arise naturally in process systems engineering applications where the lower-level problem encodes equilibrium conditions or steady-state process models.

Another approach to solve the optimistic bilvel problem (2.4), developed within the process systems engineering community, exploits the structure of bilevel problems with linear or quadratic lower levels through multiparametric programming. Faísca et al. (2007) first proposed solving the lower-level problem as a multiparametric program parametrized by the upper-level variables, reducing the bilevel problem to a sequence of independent LP or QP problems solvable to global optimality. This was subsequently extended to mixed-integer linear and quadratic lower levels by Avraamidou and Pistikopoulos (2019), providing exact global solutions for B-MILP and B-MIQP problems. A general reference for multiparametric programming in the multilevel case is found in Avraamidou and Pistikopoulos (2022).

In contrast to exact methods, data-driven approaches sacrifice optimality guarantees in exchange for the ability to handle general nonlinear and black-box problem structures. Beykal et al. (2020) proposed DOMINO, a framework that reformulates the bilevel problem as a single-level gray-box optimization problem by sampling the upper-level objective at points where the lower-level has been solved to global optimality, and applying a derivative-free solver to the resulting surrogate. DOMINO provides bilevel feasibility guarantees and near-optimal solutions across a range of problem classes including B-MINLP, but does not provide a certificate of global optimality, as the surrogate approximation of the upper-level objective does not in general reproduce the true bilevel structure. Surrogate-assisted evolutionary approaches (Islam et al., 2017; Sinha and Shaikh, 2022) follow a similar paradigm, constructing Kriging or other regression models of the lower-level response and embedding these within population-based search; these methods share the same limitation that solution quality depends on surrogate accuracy, which degrades in high dimensions.

The single-level reformulations and solution strategies reviewed above each carry significant limitations for the general smooth nonlinear bilevel setting considered here. The KKT reformulation (2.5) produces an MPCC that violates standard constraint qualifications at every feasible point, complicating both optimality theory and algorithmic design. The value-function reformulation (2.7), while theoretically equivalent to the original problem, is computationally intractable: $\phi(x)$ is implicit, nonsmooth, and nonconvex, with no closed-form expression available. Exact global methods that work around these difficulties — whether through bounding schemes or parametric solution maps — either require certified global optimality of the lower-level problem at every iteration, incurring substantial overhead, or are restricted to lower levels with linear or quadratic structure. Data-driven and surrogate-based approaches handle general nonlinear structures but approximate the bilevel objective rather than operating on it directly, providing no certificate on the gap to the true optimum.

The method proposed in this work takes a different route. Under certain regularity conditions (see next section), the lower-level solution $\bar{y}(x)$ is a continuously differentiable implicit function of x, and its sensitivity with respect to x can be computed efficiently via an adjoint system. This allows the bilevel problem to be solved directly as a smooth optimization problem in x alone, without constructing any single-level reformulation, while retaining the theoretical guarantees of KKT-based optimality conditions as developed in the following section.

3. The sensitivity-based solution method

This section details the proposed sensitivity-based algorithm for solving the optimistic bilevel problem (2.4). Our method belongs to the class of gradient-based algorithms and follows a nested approach where an outer loop updates the upper-level variables and an inner loop solves the lower-level problem for a given upper-level decision. The main idea is to circumvent the single-level reformulations discussed previously by treating the lower-level problem as a parametric optimization problem and the upper-level as an implicit problem.

3.1. The implicit upper-level problem

The algorithm is based on the insight that the lower-level's optimal decision can be viewed as an *implicit function* of x, denoted $\bar{y}(x)$. This allows to transform the bilevel problem (2.4) into an equivalent single-level problem:

$$\begin{aligned} \min_{x \in X} \quad & F(x, \bar{y}(x)) \\ \text{s.t.} \quad & G(x, \bar{y}(x)) \leq 0. \end{aligned} \tag{3.1}$$

While this problem cannot be solved directly, as $\bar{y}(x)$ is not known in closed form, this formulation enables a gradient-based solution strategy. The total derivatives of the functions in the above problem can be computed via sensitivity analysis, as detailed in the subsequent sections.

For this transformation to be valid and for the implicit function $\bar{y}(x)$ to be locally unique and continuously differentiable, we impose the following standard assumptions.

Assumption 3.1 (*Regularity Conditions*)**.** Let $(\bar{x}, \bar{y})$ be a feasible point of the bilevel problem (2.4).

(a) **Smoothness:** The functions F, G, f, and g are twice continuously differentiable in a neighborhood of $(\bar{x}, \bar{y})$.
(b) **Lower-Level Convexity:** For any feasible x in the neighborhood of $\bar{x}$, the lower-level problem (2.2) is strictly convex.
(c) **Lower-Level Regularity:** For any feasible x in the neighborhood of $\bar{x}$, the lower-level solution $\bar{y}(x)$ satisfies the Linear Independence Constraint Qualification (LICQ), the Second-Order Sufficient Condition (SOSC), and the Strict Complementarity Condition (SCC).

The above assumptions ensure, via the Implicit Function Theorem, that the solution map $\bar{y}(x)$ and its associated Lagrange multipliers $\bar{\lambda}(x)$ are continuously differentiable functions in the neighborhood of $\bar{x}$. This differentiability is fundamental to compute derivatives of the upper-level problem (3.1). For notational clarity throughout the remainder of this work, we omit the explicit dependency of the optimal lower-level solution $\bar{y}$ and its associated multipliers $\bar{\lambda}$ on x.

The strict convexity assumption in Assumption 3.1(b) excludes the important class of bilevel optimization problems whose lower-level is a linear program (LP). To accommodate this class of problems within our framework, we employ a standard regularization technique. The linear objective $f(x, y) = c(x)^T y$ is replaced by the regularized objective $f_\varepsilon(x, y) = c(x)^T y + \varepsilon\|y\|^2$, where ε is a small, positive constant (e.g., 10^{-6}). This ensures the LP has a unique solution and satisfies the necessary regularity conditions for sensitivity analysis, allowing the rest of the algorithm to be applied without modification.

3.2. Sensitivity analysis of the lower level

The optimality of the lower-level problem (2.2) for a fixed x is characterized by its Karush-Kuhn–Tucker (KKT) conditions. These are derived from the problem's Lagrangian function:

$$\mathcal{L}_f(x, y, \lambda) = f(x, y) + \lambda^\top g(x, y), \tag{3.2}$$

where $\lambda \in \mathbb{R}^s$ are the Lagrange multipliers. The KKT conditions must hold at the optimal solution $(\bar{y}, \bar{\lambda})$. The lower-level KKT conditions are:

$$\nabla_y \mathcal{L}_f(x, \bar{y}, \bar{\lambda}) = 0, \tag{3.3a}$$
$$g(x, \bar{y}) \le 0, \tag{3.3b}$$
$$\bar{\lambda} \ge 0, \tag{3.3c}$$
$$\bar{\lambda}_i g_i(x, \bar{y}) = 0, \quad \forall i = 1, \dots, s. \tag{3.3d}$$

Because of the complementarity constraints (3.3d), inactive constraints do not play a role in the optimization process. For sensitivity analysis, we only consider the constraints that are binding at the solution. We define the *active set* at the solution $\bar{y}$ as $A(x, \bar{y}) = \{ i \mid g_i(x, \bar{y}) = 0 \}$.

If the conditions in Assumption 3.1 are satisfied, we can differentiate the stationarity condition (3.3a) and the complementarity constraints (3.3d) for the active constraints. This yields the following linear system for the sensitivities:

$$\underbrace{\begin{bmatrix} H_{\mathcal{L}_f} & \nabla_y g_A^\top \\ \Lambda_A \nabla_y g_A & 0 \end{bmatrix}}_{=: M} \begin{bmatrix} \frac{d\bar{y}}{dx} \\ \frac{d\bar{\lambda}_A}{dx} \end{bmatrix} = \begin{bmatrix} -\nabla^2_{yx} \mathcal{L}_f \\ -\Lambda_A \nabla_x g_A \end{bmatrix}, \tag{3.4}$$

where $H_{\mathcal{L}_f}$ is the Hessian of the Lagrangian (3.2) with respect to y evaluated at $(x, \bar{y}, \bar{\lambda})$; g_A is the vector of active constraints, g_i for $i \in A(x, \bar{y})$; and $\Lambda_A = \mathrm{diag}(\bar{\lambda}_i)_{i \in A}$ is the diagonal matrix of active multipliers. Under Assumption 3.1, the matrix M is nonsingular, so the Jacobian $J := d\bar{y}/dx$ is uniquely defined by (3.4).

3.3. Total gradient computation for the upper level

To solve the implicit upper-level problem (3.1) via a gradient-based method, we need the total derivatives of its objective and constraint functions with respect to x. Using the sensitivity term $d\bar{y}/dx$ from the linear system (3.4), these gradients are computed via the chain rule:

$$\nabla_x F(x, \bar{y}) = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial y} \frac{d\bar{y}}{dx} \tag{3.5a}$$
$$\nabla_x G(x, \bar{y}) = \frac{\partial G}{\partial x} + \frac{\partial G}{\partial y} \frac{d\bar{y}}{dx} \tag{3.5b}$$

These gradients are essential for the iterative optimization procedure, which is designed to find a point satisfying the problem's KKT conditions. The Lagrangian for the upper-level problem is:

$$\mathcal{L}_F(x, \mu) = F(x, \bar{y}) + \mu^\top G(x, \bar{y}), \tag{3.6}$$

where $\mu \in \mathbb{R}^r$ are the Lagrange multipliers associated with the upper-level constraints. The goal of the upper-level solver is to find a point $(\bar{x}, \bar{\mu})$ that satisfies the following KKT conditions:

$$\nabla_x \mathcal{L}_F(\bar{x}, \bar{\mu}) = 0, \tag{3.7a}$$
$$G(\bar{x}, \bar{y}) \le 0, \tag{3.7b}$$
$$\bar{\mu} \ge 0, \tag{3.7c}$$
$$\bar{\mu}_i G_i(\bar{x}, \bar{y}) = 0, \quad \forall i = 1, \dots, r. \tag{3.7d}$$

Computing the full Jacobian $J = d\bar{y}/dx$ explicitly by solving (3.4) for each of the n columns requires n linear solves with M. For gradient-based upper-level methods, however, what is required is not J itself but the product $J^\top q$, where q encodes the upper-level multiplier information. This product can be obtained at the cost of a *single* linear solve by introducing an adjoint formulation.

Define the adjoint vector

$$q(x, \mu) := \nabla_y F(x, \bar{y}) + \sum_{i=1}^{r} \mu_i \nabla_y G_i(x, \bar{y}), \tag{3.8}$$

which collects the y-partial derivatives of the upper-level Lagrangian (3.6). The adjoint variables (v, w) are then defined as the solution of the linear system

$$M^\top \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} q(x, \mu) \\ 0 \end{bmatrix}. \tag{3.9}$$

Since M is nonsingular under Assumption 3.1, the adjoint system (3.9) has a unique solution (v, w). Transposing (3.4) and multiplying on the left by $[v^\top, w^\top]$, then substituting (3.9), yields the identity

$$J^\top q(x, \mu) = \left(\nabla^2_{yx} \mathcal{L}_f \right)^\top v + \left(\nabla_x g_A \right)^\top \Lambda_A w.$$

Substituting into the stationarity condition (3.7a), the gradient of the upper-level Lagrangian can be written entirely in terms of the adjoint variables:

$$\nabla_x \mathcal{L}_F(x, \mu) = \nabla_x F + \sum_{i=1}^{r} \mu_i \nabla_x G_i + \left(\nabla^2_{yx} \mathcal{L}_f \right)^\top v + \left(\nabla_x g_A \right)^\top \Lambda_A w. \tag{3.10}$$

Eq. (3.10) requires only a single solve of (3.9) per gradient evaluation and avoids forming J explicitly. This is the formulation used in the implementation.

The computational cost per inner iteration comprises of one lower-level NLP solve of dimension m and one solve of the adjoint linear system (3.9) of dimension $(m + |A|)$, where $|A| \le s$ is the number of active lower-level constraints. Forming the sensitivity Jacobian J would require n solves with the same matrix M. The adjoint formulation reduces this to a single solve regardless of the upper-level dimension n, so the gradient evaluation cost scales with the lower-level problem size only.

3.4. The proposed algorithm

Throughout this section we distinguish two iteration indices. The outer index k tracks the Augmented Lagrangian updates of the upper-level problem, producing the sequence of outer iterates x_k with associated multipliers (μ_k, ρ_k). The inner index j tracks the descent iterations used to approximately minimize each fixed Augmented Lagrangian subproblem; $x_{k,j}$ denotes the jth inner iterate for fixed (μ_k, ρ_k), and $\bar{y}_{k,j}$ is the corresponding lower-level solution obtained by solving (2.2) at $x_{k,j}$.

The upper-level update step is performed using an *Augmented Lagrangian Method (ALM)* based on the *Powell–Hestenes–Rockafellar (PHR)* augmented Lagrangian function (Powell, 1969; Hestenes, 1969; Rockafellar, 1973):

$$\mathcal{L}_\rho(x, \mu; \rho) = F(x, \bar{y}) + \frac{1}{2\rho} \sum_{i=1}^{r} \Big[\big(\max\{0,\ \mu_i + \rho G_i(x, \bar{y})\} \big)^2 - \mu_i^2 \Big], \tag{3.11}$$

where $\rho > 0$ is the penalty parameter. Defining $\hat{\mu}_i := \max\{0,\ \mu_i + \rho G_i(x, \bar{y})\}$, its first derivative with respect to x is

$$\nabla_x \mathcal{L}_\rho(x, \mu; \rho) = \nabla_x F(x, \bar{y}) + \sum_{i=1}^{r} \hat{\mu}_i \nabla_x G_i(x, \bar{y}), \tag{3.12}$$

which is a continuous function. Using the adjoint formulation (3.9)–(3.10) with μ replaced by $\hat{\mu}$, this gradient is evaluated in practice without forming the Jacobian J, by solving a single adjoint system per gradient evaluation.

At each outer iteration k, the ALM forms the augmented Lagrangian subproblem

$$\min_x \ \mathcal{L}_\rho(x, \mu_k; \rho_k). \tag{3.13}$$

This subproblem is solved inexactly using a gradient-based method:

$$x_{new} \leftarrow x_{current} + \alpha p, \tag{3.14}$$

where p is the L-BFGS-B descent direction and α is the step size determined via a line search satisfying the strong Wolfe conditions

$$\mathcal{L}_\rho(x + \alpha p, \mu_k; \rho_k) \le \mathcal{L}_\rho(x, \mu_k; \rho_k) + c_1 \alpha \nabla_x \mathcal{L}_\rho(x, \mu_k; \rho_k)^\top p, \tag{3.15a}$$

$$\left| \nabla_x \mathcal{L}_\rho(x + \alpha p, \mu_k; \rho_k)^\top p \right| \le c_2 \left| \nabla_x \mathcal{L}_\rho(x, \mu_k; \rho_k)^\top p \right|, \tag{3.15b}$$

with $0 < c_1 < c_2 < 1$.

Following the primal update, the dual variables are updated using the new iterate x_{k+1}:

$$\mu_{k+1,i} \leftarrow \max\{0,\ \mu_{k,i} + \rho_k G_i(x_{k+1}, \bar{y}_{k+1})\}. \tag{3.16}$$

This step uses the constraint violation at the new point $(x_{k+1}, \bar{y}_{k+1})$ to update the multipliers effectively. Finally, the penalty parameter ρ is managed by an adaptive scheme that balances the minimization of the objective and the satisfaction of the constraints: the penalty is increased by a factor $\gamma > 1$ only when the improvement in primal feasibility between iterations is deemed insufficient, avoiding unnecessarily large penalty values that could lead to ill-conditioning.

It is worth noting that the sensitivity and adjoint systems (3.4) and (3.9) are reconstructed at each inner iterate $x_{k,j}$ using the active set $A(x_{k,j}, \bar{y}_{k,j})$ determined by the lower-level solve at that point (Algorithm 2, Step 9, introduced below). Consequently, the gradient $\nabla_x \mathcal{L}_\rho$ used at iteration $j+1$ is always consistent with the active set at $x_{k,j+1}$, and no stale sensitivity information is carried forward. If a change in the active set occurs between iterates, the updated active set is automatically incorporated at the next gradient evaluation. The theoretical guarantees presented in the next subsection apply locally on neighborhoods of active-set stability; across transitions, the algorithm continues to produce descent steps based on the current active set, though convergence results do not extend globally across such changes.

We use the residuals of the upper-level KKT conditions (3.7) to define a stopping criterion for our algorithm. At the end of each iteration, using the new iterate (x_{k+1}, μ_{k+1}), we define the residuals as:

$$r_{stat} = \|\nabla_x \mathcal{L}_F(x_{k+1}, \mu_{k+1})\|_\infty, \tag{3.17a}$$

$$r_{feas} = \|\max\{0, G(x_{k+1}, \bar{y}_{k+1})\}\|_\infty, \tag{3.17b}$$

$$r_{comp} = \|\mathrm{diag}(\mu_{k+1}) G(x_{k+1}, \bar{y}_{k+1})\|_\infty, \tag{3.17c}$$

and the overall KKT residual is

$$r_{KKT} = \max\{r_{stat}, r_{feas}, r_{comp}\}. \tag{3.18}$$

The algorithm terminates when the KKT residual r_{KKT} falls below the prescribed tolerance $\epsilon > 0$. In practice, a secondary stall criterion is also imposed: the algorithm terminates if both $\|x_{k+1} - x_k\|_\infty < \epsilon_{stall}$ and $|F(x_{k+1}, \bar{y}_{k+1}) - F(x_k, \bar{y}_k)| < \epsilon_{stall}$ for a prescribed tolerance $\epsilon_{stall} > 0$, detecting cases where the iterates and the upper-level objective have ceased to make meaningful progress.

The complete procedure is presented in Algorithm 1; a high-level schematic of the overall framework is provided in Fig. 1.

Algorithm 1 Sensitivity-Based Algorithm for Bilevel Optimization

1: **Initialize:** Set (x_0, μ_0, ρ_0) and tolerances $\epsilon > 0$, $\epsilon_{stall} > 0$.
2: **Initial Solve:** Solve (2.2) at x_0 to obtain the KKT pair $(\bar{y}_0, \bar{\lambda}_0)$.
3: Compute $r_{feas,0} = \|\max\{0, G(x_0, \bar{y}_0)\}\|_\infty$.
4: Set $k \leftarrow 0$.
5: **repeat**
6: Find an improved iterate $(x_{k+1}, \bar{y}_{k+1}, \bar{\lambda}_{k+1}, \mu_{k+1}, \rho_{k+1})$ by solving the augmented Lagrangian subproblem (3.13) using Algorithm 2.
7: Calculate the overall KKT residual r_{KKT} using (3.17) at (x_{k+1}, μ_{k+1}).
8: $k \leftarrow k + 1$.
9: **until** $r_{KKT} < \epsilon$ **or** $\|x_k - x_{k-1}\|_\infty < \epsilon_{stall}$ **and** $|F(x_k, \bar{y}_k) - F(x_{k-1}, \bar{y}_{k-1})| < \epsilon_{stall}$
10: **return** Final solution $(\bar{x}, \bar{y})$.

The upper-level update step (Step 6 in Algorithm 1) is implemented as Algorithm 2. The most computationally intensive part of each inner iteration is the evaluation of the gradient $\nabla_x \mathcal{L}_\rho$, which is an implicit function of x; the multi-step workflow, involving solving the lower-level NLP and the adjoint sensitivity system, is illustrated in Figs. 2–3.

3.5. Convergence analysis

All results in this section are of a local nature. Convergence holds in neighborhoods of $(\bar{x}, \bar{y})$ where the lower-level KKT system is strongly regular and the active set $A(x, \bar{y})$ remains constant. On such neighborhoods the solution mapping $\bar{y}(x)$ and the reduced functions $F(x, \bar{y}(x))$ and $G(x, \bar{y}(x))$ are continuously differentiable, and the adjoint-based gradient (3.10) is well defined. Globally, the reduced objective is only piecewise smooth and may fail to be differentiable across active-set changes; accordingly, the convergence and stationarity statements below apply locally on regions of active-set stability.

For completeness, we provide a sketch of the convergence proof of Algorithm 1 using the upper-level update methodology presented in Algorithm 2. The proof adapts the standard convergence analysis for the Augmented Lagrangian method (Nocedal and Wright, 2006) to our sensitivity-based framework for bilevel optimization.

Theorem 3.1 (*Convergence to a KKT point*). *Let $\{x_k, \mu_k\}$ be a sequence of iterates generated by Algorithm 1, with the upper-level update step given by Algorithm 2. Assume that:*

(a) The regularity conditions in Assumption 3.1 hold for every point in the sequence, and the inner solves in Algorithm 2 produce steps satisfying the strong Wolfe conditions (3.15).
(b) The sequence of iterates $\{x_k\}$ is contained within a compact set X, and the corresponding sequence of generated multipliers $\{\mu_k\}$ is bounded.

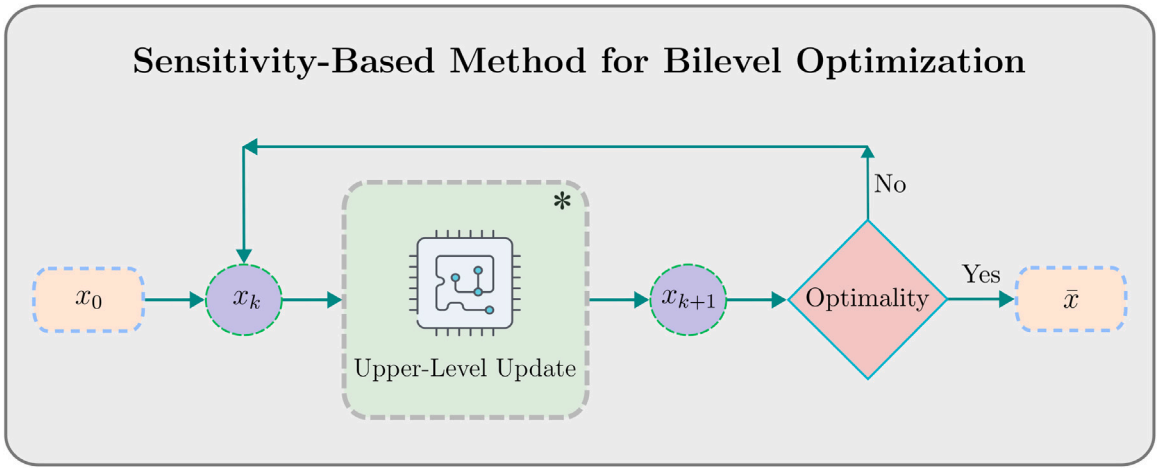

Fig. 1. High-level schematic of the overall sensitivity-based Augmented Lagrangian framework, corresponding to Algorithm 1.

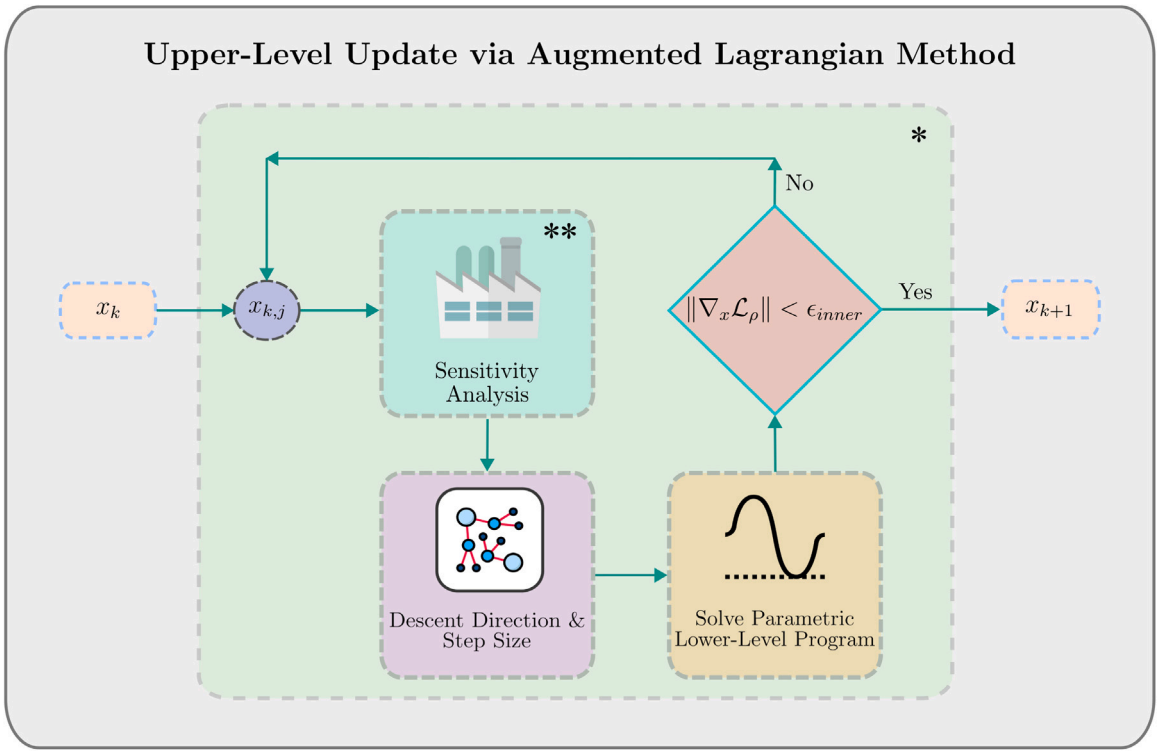

Fig. 2. Workflow for the ALM subproblem solution (Algorithm 2). At each outer iteration, a NLP solver is used to find an approximate minimizer of the implicit Augmented Lagrangian function.

Then any limit point $(\bar{x}, \bar{\mu})$ *of the sequence* $\{x_k, \mu_k\}$ *is a KKT point of* (3.1)*, with stationarity in the sense of* (3.10)*.*

Proof. First, we establish the existence of a limit point for the sequence of iterates. By Assumption (b), the sequence $\{x_k\}$ is contained within a compact set X, and the sequence of multipliers $\{\mu_k\}$ is bounded. This implies that the joint sequence $\{x_k, \mu_k\}$ is also contained within a compact set. Therefore, by the Bolzano–Weierstrass theorem, there exists at least one convergent subsequence. Let $(\bar{x}, \bar{\mu})$ be the limit point of such a subsequence, indexed by $\mathcal{K} \subseteq \mathbb{N}$, such that:

$$\lim_{k\to\infty,\, k\in\mathcal{K}} (x_k, \mu_k) = (\bar{x}, \bar{\mu}). \tag{3.19}$$

We now show that $(\bar{x}, \bar{\mu})$ satisfies the KKT conditions of the upper-level problem (3.7).

1. Dual feasibility: The multiplier update rule in Algorithm 2, given by (3.16), ensures that every component of μ_k is non-negative for all $k > 1$. Since the terms of the convergent subsequence are non-negative, their limit must also be non-negative, i.e., $\bar{\mu} \geq 0$.

2. Primal feasibility: Consider the two possible behaviors of the penalty parameter sequence $\{\rho_k\}$.

(a) The sequence remains bounded. This implies that there exists sufficiently large $\bar{k} \in \mathcal{K}$, after which the penalty parameter is no longer increased, i.e., $\rho_k = \rho_{\bar{k}}$ for $k > \bar{k}$. By Algorithm 2, this happens if the condition

$$\|\max\{0, G(x_{k+1}, \bar{y}_{k+1})\}\|_\infty \leq c \cdot \|\max\{0, G(x_k, \bar{y}_k)\}\|_\infty, \tag{3.20}$$

with $c \in (0, 1)$, is satisfied for all $k > \bar{k}$. This implies that the sequence of feasibility residuals $\{r_{feas,k}\}$ converges to zero. Therefore, $r_{feas,k} \to 0$, implying $G(\bar{x}, \bar{y}) \leq 0$ by continuity.

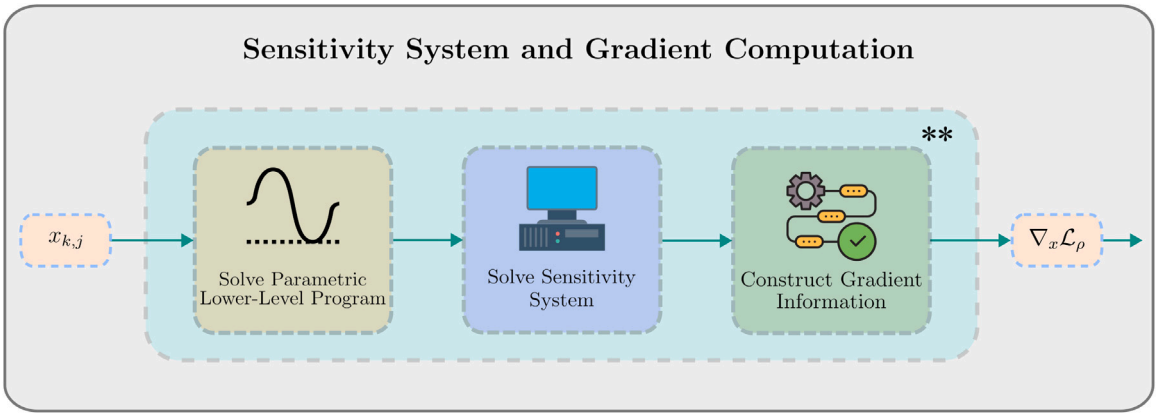

Fig. 3. Detailed workflow for the implicit objective and gradient evaluation. This multi-step process, which includes solving the lower-level NLP and the adjoint linear system, is called at each iteration of the inner loop.

Algorithm 2 Upper-Level Update via Augmented Lagrangian Method

Input: Current iterates $(x_k, \bar{y}_k, \bar{\lambda}_k, \mu_k, \rho_k, r_{feas,k})$ and parameters $\gamma > 1$, $c \in (0,1)$, $\epsilon_{inner} > 0$.
Initialize inner loop: Set $x_{k,0} \leftarrow x_k$, $\bar{y}_{k,0} \leftarrow \bar{y}_k$, $\bar{\lambda}_{k,0} \leftarrow \bar{\lambda}_k$.
Set $j \leftarrow 0$.
repeat
  Solve the adjoint system (3.9) with $q(x_{k,j}, \hat{\mu}_k)$ defined in (3.8) and compute $\nabla_x \mathcal{L}_\rho(x_{k,j}, \mu_k; \rho_k)$ using (3.10), without explicitly forming $J = d\bar{y}/dx$.
  Compute descent direction $p_{k,j}$ using L-BFGS-B with gradient $\nabla_x \mathcal{L}_\rho(x_{k,j}, \mu_k; \rho_k)$.
  Find step size $\alpha_{k,j}$ via a line search satisfying the strong Wolfe conditions (3.15).
  Update primal variables: $x_{k,j+1} \leftarrow x_{k,j} + \alpha_{k,j}\, p_{k,j}$.
  Solve the lower-level problem (2.2) at $x_{k,j+1}$ to get the KKT pair $(\bar{y}_{k,j+1}, \bar{\lambda}_{k,j+1})$.
  $j \leftarrow j + 1$.
until $\|\nabla_x \mathcal{L}_\rho(x_{k,j}, \mu_k; \rho_k)\|_\infty < \epsilon_{inner}$
Set final iterates: $x_{k+1} \leftarrow x_{k,j}$, $\bar{y}_{k+1} \leftarrow \bar{y}_{k,j}$, and $\bar{\lambda}_{k+1} \leftarrow \bar{\lambda}_{k,j}$.
Update dual variables: $\mu_{k+1,i} \leftarrow \max\{0, \mu_{k,i} + \rho_k G_i(x_{k+1}, \bar{y}_{k+1})\}$.
Calculate constraint violation $r_{feas,k+1} \leftarrow \|\max\{0, G(x_{k+1}, \bar{y}_{k+1})\}\|_\infty$.
if $r_{feas,k+1} > c \cdot r_{feas,k}$ **then**
  $\rho_{k+1} \leftarrow \gamma \cdot \rho_k$
else
  $\rho_{k+1} \leftarrow \rho_k$
end if
return $(x_{k+1}, \bar{y}_{k+1}, \bar{\lambda}_{k+1}, \mu_{k+1}, \rho_{k+1}, r_{feas,k+1})$.

(b) The sequence diverges to infinity, i.e., $\rho_k \to \infty$. Suppose for contradiction that the limit point $\bar{x}$ is infeasible, meaning there is at least one constraint j such that $G_j(\bar{x}, \bar{y}) > 0$. By continuity of G_j and $\bar{y}(x)$ (Assumption (a)), $G_j(x_k, \bar{y}_k) > 0$ for sufficiently large k. Since $\{x_k\}$ and $\{\mu_k\}$ are bounded, all other terms in $\mathcal{L}_\rho$ remain bounded. However, the penalty term for constraint j grows asymptotically like $\frac{\rho_k}{2} G_j(x_k, \bar{y}_k) \to \infty$ as $\rho_k \to \infty$.

The strong Wolfe conditions (3.15) ensure that the sequence of augmented Lagrangian values is non-increasing,

$$\mathcal{L}_\rho(x_{k+1}, \mu_k; \rho_k) \le \mathcal{L}_\rho(x_k, \mu_k; \rho_k), \tag{3.21}$$

and hence bounded above. This contradicts the divergence just established. Therefore, the initial assumption of infeasibility is false, and the limit point must be feasible.

3. Stationarity: By construction of Algorithm 2, the primal update step satisfies the strong Wolfe conditions (3.15). Under Assumption 3.1, the reduced augmented Lagrangian is continuously differentiable on the neighborhood of active-set stability, and standard results for strong Wolfe line searches ensure the gradient norm converges to zero (Nocedal and Wright, 2006). Therefore:

$$\lim_{k\to\infty} \|\nabla_x \mathcal{L}_\rho(x_k, \mu_k; \rho_k)\| = 0. \tag{3.22}$$

We substitute the dual variable update rule (3.16) into the adjoint gradient (3.10) and use continuity of the adjoint mapping (Assumption (a)) to pass to the limit along $\mathcal{K}$. As $k \to \infty$ for $k \in \mathcal{K}$, we have $x_k \to \bar{x}$, $\mu_k \to \bar{\mu}$, and $\mu_{k+1} \to \bar{\mu}$. Taking the limit gives:

$$0 = \lim_{k\to\infty,\, k\in\mathcal{K}} \left\| \nabla_x F(x_k, \bar{y}_k) + \mu_{k+1}^\top \nabla_x G(x_k, \bar{y}_k) \right\| \tag{3.23a}$$

$$= \left\| \nabla_x F(\bar{x}, \bar{y}) + \bar{\mu}^\top \nabla_x G(\bar{x}, \bar{y}) \right\| \tag{3.23b}$$

$$= \|\nabla_x \mathcal{L}_F(\bar{x}, \bar{\mu})\|. \tag{3.23c}$$

This directly implies that the stationarity condition $\nabla_x \mathcal{L}_F(\bar{x}, \bar{\mu}) = 0$ is satisfied.

4. Complementarity: Having established primal feasibility, we consider the two cases for any constraint j at the limit point.

(a) The constraint is inactive, i.e., $G_j(\bar{x}, \bar{y}) < 0$. By continuity, there exists a $\bar{k}$ such that for all $k \in \mathcal{K}$ with $k > \bar{k}$ we have $G_j(x_k, y_k) < 0$. By Assumption (b), the sequence $\{\mu_k\}$ is bounded. Regardless of whether $\{\rho_k\}$ is bounded or diverges, the negative term $\rho_k G_j(x_k, y_k)$ is guaranteed to eventually dominate the bounded, non-negative $\mu_{k,j}$. This ensures that the term $\mu_{k,j} + \rho_k G_j(x_k, y_k)$ will be negative for all $k \in \mathcal{K}$ with $k > \bar{k}$, forcing $\mu_{k+1,j}$ to be zero via the max operator in (3.16). Therefore, the limit $\bar{\mu}_j$ must be zero.

(b) The constraint is active, i.e., $G_j(\bar{x}, \bar{y}) = 0$. In this case, the condition $\bar{\mu}_j G_j(\bar{x}, \bar{y}) = 0$ is trivially satisfied.

Since all KKT conditions are satisfied, any limit point $(\bar{x}, \bar{\mu})$ is a KKT point of the implicit problem (3.1). □

3.6. Equivalence to S-stationarity

We now establish the equivalence between a KKT point of the implicit problem (3.1) and a stationary point of the MPCC reformulation (2.5). A constraint qualification tailored for these problems is *MPEC-LICQ*, which requires that the gradients of the active upper-level constraints, the active lower-level constraints, and the lower-level stationarity equations are linearly independent. This condition ensures that the multipliers of the MPCC are well-defined (Luo et al., 1996).

The Lagrangian for the MPCC (2.5) is given by:

$$\mathcal{L}_{MPCC}(x, y, \lambda, \mu, \nu, \pi, \xi) = F(x, y) + \mu^\top G(x, y) + \nu^\top \nabla_y \mathcal{L}_f(x, y, \lambda) + \pi^\top g(x, y) - \xi^\top \lambda, \tag{3.24}$$

where μ, ν, π, ξ are the Lagrange multipliers following standard sign convention: the equality constraint multiplier ν is free, and the inequality constraint multipliers μ, π, ξ are non-negative. The complementarity condition $\bar{\lambda}_i g_i(\bar{x}, \bar{y}) = 0$ for all i is not incorporated into (3.24) directly; it is handled through index-set dependent sign rules on the multipliers.

A feasible point $(\bar{x}, \bar{y}, \bar{\lambda})$ of the MPCC (2.5) is *S–stationary* if there exist multipliers $(\bar{\mu}, \bar{\nu}, \bar{\pi}, \bar{\xi})$ such that the following hold:

1. Stationarity:

$$\nabla_x \mathcal{L}_{MPCC} = \nabla_x F + \nabla_x G^\top \bar{\mu} + (\nabla^2_{yx} \mathcal{L}_f)^\top \bar{\nu} + \nabla_x g^\top \bar{\pi} = 0, \quad (3.25a)$$

$$\nabla_y \mathcal{L}_{MPCC} = \nabla_y F + \nabla_y G^\top \bar{\mu} + (\nabla^2_{y} \mathcal{L}_f)^\top \bar{\nu} + \nabla_y g^\top \bar{\pi} = 0, \quad (3.25b)$$

$$\nabla_\lambda \mathcal{L}_{MPCC} = (\nabla_y g)\, \bar{\nu} - \bar{\xi} = 0. \quad (3.25c)$$

2. Primal feasibility:

$$G(\bar{x}, \bar{y}) \le 0, \quad \nabla_y \mathcal{L}_f(\bar{x}, \bar{y}, \bar{\lambda}) = 0, \quad g(\bar{x}, \bar{y}) \le 0, \quad \bar{\lambda} \ge 0, \quad \bar{\lambda}_i g_i(\bar{x}, \bar{y}) = 0 \ \forall i.$$

3. Dual feasibility and complementarity slackness:

$$\bar{\mu} \ge 0, \quad \bar{\pi} \ge 0, \quad \bar{\xi} \ge 0,$$

$$\bar{\mu}^\top G(\bar{x}, \bar{y}) = 0, \quad \bar{\pi}^\top g(\bar{x}, \bar{y}) = 0, \quad \bar{\xi}^\top \bar{\lambda} = 0.$$

4. Sign rules: Define index sets

$$\mathcal{I}^+ = \{i : \ g_i(\bar{x}, \bar{y}) = 0, \ \bar{\lambda}_i > 0\},$$
$$\mathcal{I}^- = \{i : \ g_i(\bar{x}, \bar{y}) \le 0, \ \bar{\lambda}_i = 0\},$$
$$\mathcal{I}^0 = \{i : \ g_i(\bar{x}, \bar{y}) = 0, \ \bar{\lambda}_i = 0\}.$$

Then for every i:

$$\begin{cases} i \in \mathcal{I}^+ : & \bar{\pi}_i \ge 0, \ \bar{\xi}_i = 0, \\ i \in \mathcal{I}^- : & \bar{\pi}_i = 0, \ \bar{\xi}_i \ge 0, \\ i \in \mathcal{I}^0 : & \bar{\pi}_i \ge 0, \ \bar{\xi}_i \ge 0. \end{cases}$$

Theorem 3.2 (*Equivalence to S-stationarity*)**.** *Let $(\bar{x}, \bar{y})$ be a feasible point for the bilevel problem (2.4), and let $\bar{\lambda}$ be a multiplier such that $(\bar{x}, \bar{y}, \bar{\lambda})$ satisfies the KKT conditions of the lower-level problem (2.2). Assume that the lower-level regularity conditions (Assumption 3.1) hold in a neighborhood of $(\bar{x}, \bar{y})$. Assume moreover that MPEC–LICQ holds for the MPCC (2.5) at $(\bar{x}, \bar{y}, \bar{\lambda})$.*

Then, there exists $\bar{\mu}$ such that $(\bar{x}, \bar{\mu})$ is a KKT point of the implicit problem (3.1) if and only if $(\bar{x}, \bar{y}, \bar{\lambda})$ is an S–stationary point of the MPCC reformulation (2.5).

Proof. The sensitivity matrix M and sensitivity system are as defined in (3.4); by Assumption 3.1, M is nonsingular. Denote the active and inactive index sets at $(\bar{x}, \bar{y})$ by $A = \{i : \ g_i(\bar{x}, \bar{y}) = 0, \ \bar{\lambda}_i > 0\}$ and $I = \{i : \ g_i(\bar{x}, \bar{y}) < 0, \ \bar{\lambda}_i = 0\}$, respectively; SCC (Assumption 3.1) rules out the biactive case.

The stationarity condition of the implicit problem (3.1) at $(\bar{x}, \bar{\mu})$, written via the chain rule, is:

$$\nabla_x F(\bar{x}, \bar{y}) + \nabla_x G(\bar{x}, \bar{y})^\top \bar{\mu} + \left(\frac{d\bar{y}}{dx}\right)^\top \left(\nabla_y F(\bar{x}, \bar{y}) + \nabla_y G(\bar{x}, \bar{y})^\top \bar{\mu}\right) = 0. \quad (3.26)$$

($\Rightarrow$) Suppose $(\bar{x}, \bar{\mu})$ is a KKT point of (3.1). The adjoint variables $(\bar{\nu}, \bar{w})$ are the unique solution of the adjoint system (3.9) evaluated at $(\bar{x}, \bar{y}, \bar{\lambda}, \bar{\mu})$, with right-hand side given by $q(\bar{x}, \bar{\mu})$ as defined in (3.8). Applying the adjoint identity (3.10) to (3.26) gives:

$$\left(\frac{d\bar{y}}{dx}\right)^\top \left(\nabla_y F(\bar{x}, \bar{y}) + \nabla_y G(\bar{x}, \bar{y})^\top \bar{\mu}\right) = \left(\nabla^2_{yx} \mathcal{L}_f\right)^\top \bar{\nu} + (\nabla_x g_A)^\top \bar{\Lambda}_A \bar{w}, \quad (3.27)$$

and substituting into (3.26) yields:

$$\nabla_x F(\bar{x}, \bar{y}) + \nabla_x G(\bar{x}, \bar{y})^\top \bar{\mu} + \left(\nabla^2_{yx} \mathcal{L}_f\right)^\top \bar{\nu} + (\nabla_x g_A)^\top \bar{\Lambda}_A \bar{w} = 0. \quad (3.28)$$

Define MPCC multipliers:

$$\bar{\pi}_I = 0, \quad \bar{\pi}_A = \bar{\Lambda}_A \bar{w}, \quad \bar{\xi} = \nabla_y g(\bar{x}, \bar{y})\, \bar{\nu}. \quad (3.29)$$

Since $\bar{\Lambda}_A$ is diagonal with strictly positive entries, $\bar{\pi}_A \ge 0$ if and only if $\bar{w} \ge 0$. Then (3.28) is the MPCC stationarity condition with respect to x (3.25a).

Stationarity with respect to y follows from the first block of (3.9):

$$H_{\mathcal{L}_f} \bar{\nu} + \nabla_y g_A(\bar{x}, \bar{y})^\top \Lambda_A \bar{w} = -\left(\nabla_y F(\bar{x}, \bar{y}) + \nabla_y G(\bar{x}, \bar{y})^\top \bar{\mu}\right), \quad (3.30)$$

and since $H_{\mathcal{L}_f} = \nabla^2_{yy} \mathcal{L}_f$ and $\bar{\pi}_I = 0$, this is (3.25b). From the second block of (3.9), $\nabla_y g_A \bar{\nu} = 0$, which implies $\bar{\xi}_A = 0$; hence (3.25c) is satisfied.

Primal feasibility of $(\bar{x}, \bar{y}, \bar{\lambda})$ holds by construction, and complementarity $(\bar{\mu} \ge 0, \ \bar{\mu}^\top G = 0)$ is inherited from the KKT conditions of the implicit problem.

To verify the S–stationarity sign conditions: $\bar{\xi}_A = 0$ follows from the adjoint system, and $\bar{\pi}_I = 0$ holds by construction. For $i \in A$, $\bar{\pi}_i$ represents the sensitivity of F to the lower-level constraint g_i; since $\bar{x}$ is a local minimum, relaxing g_i cannot improve F, so $\bar{\pi}_A \ge 0$. For $i \in I$, $\bar{\xi}_i$ is the sensitivity to $\lambda_i = 0$; a negative $\bar{\xi}_i$ would imply the objective improves by activating g_i, contradicting optimality of $\bar{x}$, so $\bar{\xi}_I \ge 0$. The S–stationarity sign rules are therefore satisfied:

$$i \in A : \ \bar{\pi}_i \ge 0, \ \bar{\xi}_i = 0; \qquad i \in I : \ \bar{\pi}_i = 0, \ \bar{\xi}_i \ge 0.$$

This proves that $(\bar{x}, \bar{y}, \bar{\lambda})$ is S–stationary for (2.5).

($\Leftarrow$) Conversely, suppose $(\bar{x}, \bar{y}, \bar{\lambda})$ is S–stationary for the MPCC, with multipliers $(\bar{\mu}, \bar{\nu}, \bar{\pi}, \bar{\xi})$ satisfying (3.25). Under SCC (Assumption 3.1), $\bar{\Lambda}_A$ is invertible, and we define

$$\bar{w} := \bar{\Lambda}_A^{-1} \bar{\pi}_A. \quad (3.31)$$

Since $\bar{\pi}_A \ge 0$ and $\bar{\Lambda}_A$ has strictly positive diagonal entries, $\bar{w} \ge 0$, so the sign rules are preserved. The MPCC stationarity conditions with respect to y (3.25b) and with respect to λ (3.25c), together with (3.31), are equivalent to the adjoint system (3.9). The MPCC stationarity condition with respect to x (3.25a), combined with the identity (3.27), which follows from the adjoint system, directly yields the implicit problem's stationarity condition (3.26). Feasibility and complementarity transfer directly. Hence $(\bar{x}, \bar{\mu})$ is a KKT point of the implicit problem (3.1).

In summary, the adjoint variables $(\bar{\nu}, \bar{w})$ provide the MPCC multipliers via $\bar{\pi}_A = \bar{\Lambda}_A \bar{w}$ and $\bar{\xi} = \nabla_y g\, \bar{\nu}$, while the upper-level multipliers $\bar{\mu}$ are preserved, establishing the equivalence between the reduced upper-level KKT points and S-stationary solutions of the MPCC reformulation. □

4. Computational tests

This section presents the computational validation of the proposed sensitivity-based algorithm, covering implementation details, illustrative examples, and systematic benchmark experiments.

4.1. Implementation details

The method was implemented in Python, leveraging CasADi (Andersson et al., 2019) for its automatic differentiation capabilities. All computations were performed on a Windows 10 Enterprise (64-bit) workstation equipped with an Intel i5-6500 CPU (4 cores) and 16 GB RAM. The lower-level parametric NLPs were solved using IPOPT (Wächter and Biegler, 2006). The inner ALM subproblem (3.13) was solved using the L-BFGS-B algorithm (Byrd et al., 1995) with a strong Wolfe line search. Upper-level gradients were computed via the adjoint formula (3.10), avoiding explicit formation of the Jacobian J.

The Augmented Lagrangian method was selected over a pure penalty approach because it avoids the severe ill-conditioning associated with large penalty parameters, and over sequential quadratic programming (SQP) because the implicit gradient structure of the reduced problem makes reliable second-order information expensive to obtain. The strong Wolfe line search was preferred over a simple backtracking Armijo rule because the curvature condition (3.15b) is required to

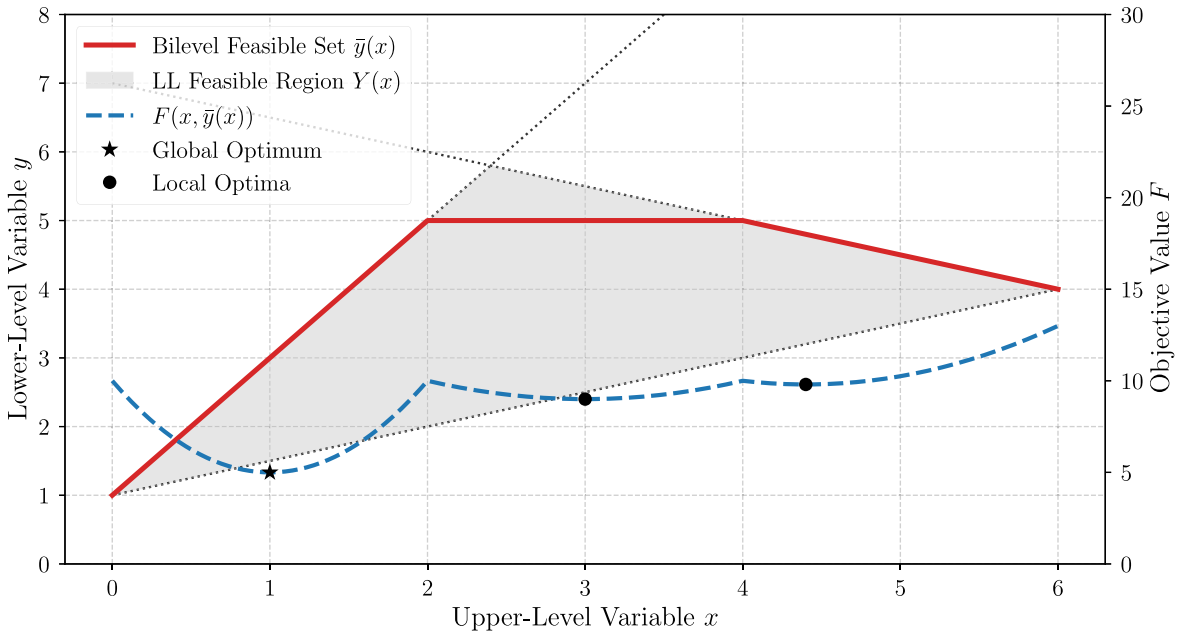

Fig. 4. The implicit upper-level objective $F(x, \bar{y}(x))$ and the lower-level optimal response $\bar{y}(x)$ as a function of the upper-level variable x for the `ClarkWesterberg1990` problem. The local and global optima are highlighted.

guarantee that the L-BFGS-B Hessian approximation remains positive definite throughout the inner iterations.

The tolerances were set to $\epsilon = 10^{-5}$ for the KKT residual (3.18), $\epsilon_{\text{inner}} = 10^{-6}$ for the inner solve, and $\epsilon_{\text{stall}} = 10^{-5}$ for the stall criterion. The initial penalty parameter was set to $\rho_0 = 10$ and the initial upper-level multipliers to $\mu_0 = 0$. For lower-level problems with linear objectives, ϵ-regularization with $\epsilon = 10^{-6}$ was applied to enforce uniqueness of the lower-level solution. A well-known practical feature of the Augmented Lagrangian method is that primal variables often converge much more rapidly than the dual variables, whose first-order update can exhibit slow linear convergence (Nocedal and Wright, 2006). Therefore, while the primary termination criterion is the KKT residual falling below ϵ, a secondary stall criterion terminates the algorithm when successive changes in x and F fall below ϵ_{stall}, preventing excessive iterations spent refining dual variables once no meaningful primal improvement remains.

At each inner iterate $x_{k,j}$, the lower-level problem (2.2) is solved to obtain the KKT pair $(\bar{y}_{k,j}, \bar{\lambda}_{k,j})$ (Algorithm 2, Step 9). Existence and local uniqueness of this solution are guaranteed by Assumption 3.1: the LICQ, SCC, and SSOSC conditions together ensure strong regularity of the lower-level KKT system, which implies that the solution mapping $\bar{y}(x)$ is locally single-valued and continuously differentiable in a neighborhood of each iterate, so the lower-level solve is well-posed throughout the algorithm.

For each benchmark problem, multiple starting points were selected by manual exploration within the reported variable bounds, informed by inspection of the upper-level objective landscape. The best solution found across all initializations is reported in Table 1. We note that systematic space-filling designs commonly used for multi-start strategies in single-level optimization — such as Latin hypercube sampling or uniform grids — are not directly applicable to bilevel problems: the bilevel feasible region is implicitly defined through the lower-level solution mapping $\bar{y}(x)$, and feasibility of a candidate point with respect to the upper-level constraints $G(x, \bar{y}(x)) \leq 0$, can only be assessed after solving the lower-level problem at x_0. Exploration of the implicit objective landscape therefore necessarily proceeds by solving the lower-level problem at each candidate point, making manual exploration guided by problem structure a natural and practical approach for the small-dimensional benchmark problems considered here.

4.2. Test problems

As an illustrative example, we first conduct a detailed analysis of the classic `ClarkWesterberg1990` problem (Clark and Westerberg, 1990), a well-known benchmark in the chemical engineering literature. The problem is defined as:

$$\begin{aligned} &\min_x \ (x-3)^2 + (y-2)^2 \\ &\text{s.t. } 0 \leq x \leq 8 \\ &\quad y \in \Psi(x) = \underset{y}{\operatorname{argmin}} \left\{ (y-5)^2 \;\middle|\; \begin{array}{l} -2x + y - 1 \leq 0, \\ x - 2y + 2 \leq 0, \\ x + 2y - 14 \leq 0 \end{array} \right\}. \end{aligned} \tag{4.1}$$

The geometry of this problem is illustrated in Fig. 4. The implicit upper-level objective, $F(x, \bar{y}(x))$, is continuous but non-smooth, with non-differentiable kinks at $x = 2$ and $x = 4$ that correspond to changes in the lower-level active set. This non-convex landscape gives rise to multiple optima, including a global minimum at $x = 1$ and two distinct local minima. This sensitivity to the initial point underscores the necessity of a multi-start strategy to systematically explore the solution space, a characteristic feature of non-convex bilevel problems.

To demonstrate the algorithm's performance on a problem with active upper-level constraints, we present the convergence results for the `Outrata_Cervinka_2009` problem in Fig. 5. This problem is defined as

$$\begin{aligned} &\min_x \ -2x_1 - 0.5x_2 - y_2 \\ &\text{s.t. } x_1 \leq 0 \\ &\quad y \in \Psi(x) = \underset{y}{\operatorname{argmin}} \left\{ y_1 - y_2 + x^\top y + y^\top y \;\middle|\; \begin{array}{l} y_2 - y_1 \leq 0, \\ y_2 + y_1 \leq 0, \\ y_2 \leq 0 \end{array} \right\}. \end{aligned} \tag{4.2}$$

Unlike cases that terminate due to stalling, this problem demonstrates convergence via the primary KKT criterion, as observed for several problems in Section 4.3. The plot shows the KKT residual decreasing by several orders of magnitude to meet the tolerance, while the upper-level multiplier μ_1 converges rapidly to its optimal value. This provides strong numerical evidence that the algorithm performs as theoretically intended.

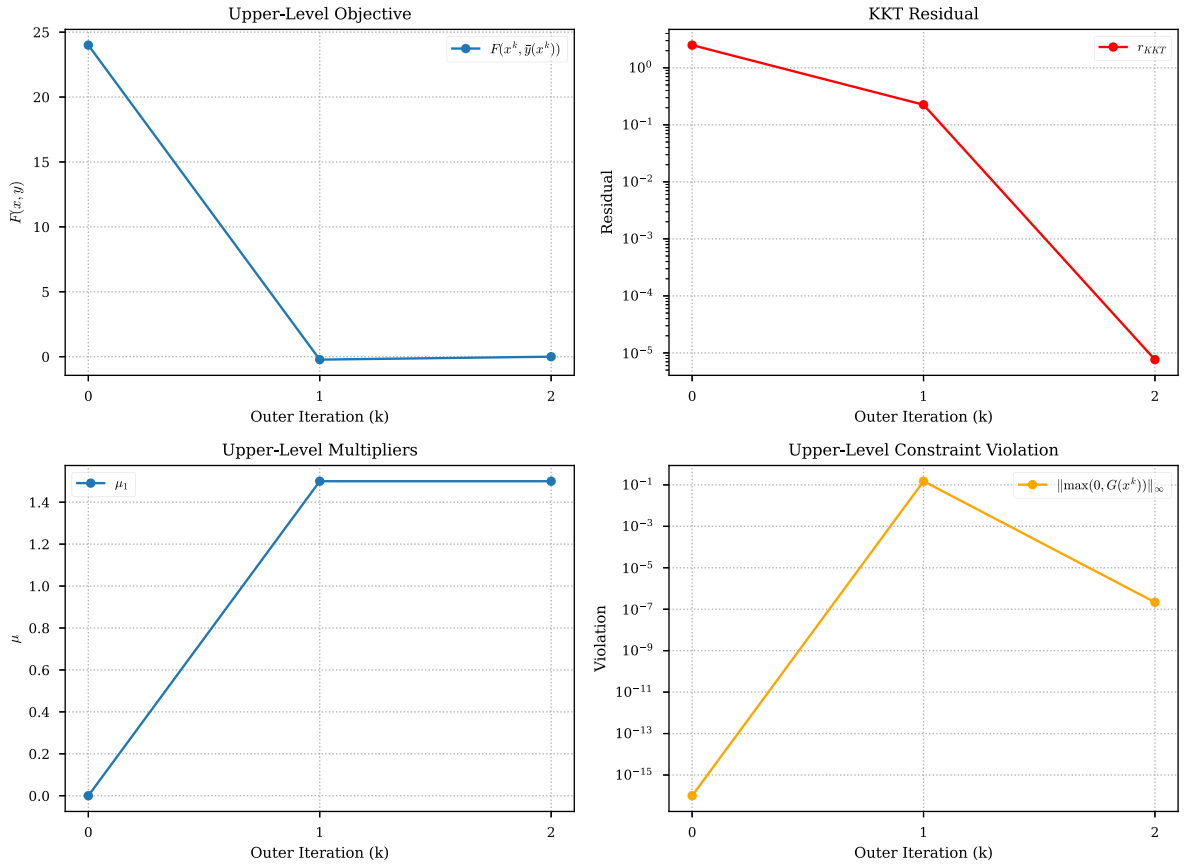

Fig. 5. Convergence behavior for the `Outrata_Cervinka_2009` problem. The algorithm converges in a few iterations as the KKT residual drops below the tolerance $\epsilon = 10^{-5}$.

4.3. Benchmark results and discussion

To demonstrate the broader applicability and robustness of the proposed method, the algorithm was tested on a suite of benchmark problems from the BOLIB library (Ward et al., 2025). A summary of these computational experiments is presented in Table 1. For each problem, a multi-start strategy was employed, and the best solution found is reported. The table details the problem dimensions as the tuple (n, m, r, s), where n is the dimension of the upper-level decision vector $x \in \mathbb{R}^n$, m is the dimension of the lower-level decision vector $y \in \mathbb{R}^m$, r and s are the number of constraints in the upper and lower levels, respectively, including bound constraints, as reported in the literature. It also reports the initial point x_0 that led to the best solution, the known optimal value from the literature $\bar{F}_r$, the value computed by the proposed method $\bar{F}_c$, the number of outer ALM iterations, and the wall-clock solution time in seconds.

The results across the benchmark suite reveal several consistent behavioral patterns. For problems with relatively simple landscapes, the algorithm converges in very few outer iterations, reflecting the efficiency of the quasi-Newton inner solver. Harder instances typically exhibit rapid primal progress accompanied by slower dual convergence, a well-known feature of the PHR multiplier update (3.16), whose first-order scheme can exhibit slow linear convergence (Nocedal and Wright, 2006). When constraints are strictly feasible at the solution, the projection drives the corresponding multipliers to zero, consistent with KKT conditions for inactive constraints. Near-binding constraints, however, tend to produce small oscillations in the multiplier trajectory and a slower decay of the complementarity residual, making the stall criterion essential. This is illustrated by `AiyoshiShimizu1984Ex2`, where the primal solution is identified rapidly but convergence is declared via the stall criterion due to slow dual refinement. The ϵ-regularization stabilizes lower-level linear program cases (marked by † in Table 1) without altering solutions on regions where the active set remains constant, as confirmed by the agreement between computed and literature values for these instances. Overall, the results are consistent with the best-reported solutions in the literature, demonstrating both the correctness and the robustness of the proposed method.

Direct numerical comparison is not straightforward, as the sensitivity-based paradigm adopted here does not have a close algorithmic counterpart in the existing literature. Three alternative paradigms exist, each addressing different trade-offs between generality, computational cost, and solution quality. Deterministic global methods based on bounding schemes, such as that of Mitsos et al. (2008) for continuous bilevel programs with nonconvex lower levels, provide rigorous ϵ-optimality certificates. However, these methods require certified global optimality of the lower-level problem at every iteration, incurring substantial computational overhead that grows rapidly with problem dimension. For the smooth, locally well-posed problems considered here, this requirement is excessive given the analytic structure available, and such methods do not exploit the gradient information that this structure affords. Other deterministic approaches based on multiparametric programming (Faísca et al., 2007; Avraamidou and Pistikopoulos, 2019, 2022) provide exact global solutions but are restricted to lower-level problems with linear or quadratic structure; they do not extend to the general smooth nonlinear lower levels addressed here. Furthermore, the explicit parametric solution map constructed by these methods scales combinatorially with the number of lower-level constraints and the dimension of the upper-level variable space, whereas the proposed method requires only a single lower-level NLP solve and one adjoint linear system per iteration, with per-iteration cost independent of the upper-level dimension n.

Data-driven methods such as DOMINO (Beykal et al., 2020) and surrogate-assisted evolutionary algorithms (Islam et al., 2017; Sinha

Table 1
Performance of the sensitivity-based ALM on selected benchmark problems. Reported objective values are rounded to one decimal place; agreement with literature values is understood within the convergence tolerance $\epsilon = 10^{-5}$.

Problem	(n, m, r, s)	x_0	$\bar{F}_r$	$\bar{F}_c$	Iters	Time (s)
AiyoshiShimizu1984Ex2	(2, 2, 5, 6)	(20.0, 20.0)	5.0	5.0	12	2.20
AllendeStill2013	(2, 2, 5, 2)	(2.0, 2.0)	−1.0	−1.0	11	1.11
Bard_1988_ex1	(1, 1, 1, 4)	2.0	17.0	17.0	11	0.91
Bard_1991_ex1	(1, 2, 2, 3)	4.0	2.0	2.0	11	0.48
Bard_Book_1998[a]	(2, 2, 4, 7)	(15, 15)	0.0	0.0	6	0.28
ClarkWesterberg1990	(1, 1, 2, 3)	1.7	5.0	5.0	11	0.33
DempeEtal2012[a]	(1, 1, 2, 2)	0.9	−1.0	−1.0	11	0.22
Dempe_Franke_2011_ex42[a]	(2, 2, 4, 3)	(−0.9, 0.9)	3.0	3.0	11	2.75
Dempe_Lohse_2011_ex31a[a]	(2, 2, 0, 4)	(−0.4, −0.4)	−5.5	−5.5	11	5.35
Dempe_Lohse_2011_ex31b[a]	(3, 3, 0, 5)	(4.0, 4.0, 4.0)	−12.0	−12.0	12	1.13
FloudasEtal2013	(2, 2, 4, 7)	(10.0, 10.0)	0.0	0.0	11	0.48
Outrata_Cervinka_2009[a]	(2, 2, 1, 3)	(−10.0, −1.0)	0.0	0.0	12	0.67
Shimizu_Aiyoshi_1981_ex2	(2, 2, 3, 4)	(10.0, 1.0)	225.0	225.0	39	4.20

[a] Lower-level objective is linear; ϵ-regularization with $\epsilon = 10^{-6}$ applied to enforce uniqueness of the lower-level solution.

and Shaikh, 2022) handle general nonlinear and black-box structures without requiring differentiability. However, these approaches construct a surrogate approximation of the upper-level objective from sampled lower-level solutions and optimize the surrogate in place of the original bilevel problem. The surrogate is not in general equivalent to the original bilevel structure: its optimum provides a feasible but only near-optimal solution, with no certificate on the gap to the true bilevel optimum (Beykal et al., 2020). Furthermore, surrogate accuracy degrades with problem dimension due to the curse of dimensionality, limiting scalability. In contrast, the proposed sensitivity-based method operates directly on the true bilevel problem without surrogate approximation, and exploits the available gradient structure therefore the computational effort is concentrated in solving the lower-level NLP and one adjoint linear system per gradient evaluation, both of which exploit the analytic structure of the problem rather than approximating it.

5. Conclusions

In this work, a novel sensitivity-based algorithm for solving continuous, optimistic bilevel optimization problems was developed. By treating the lower-level problem as an implicit function of the upper-level variables, this approach addresses the hierarchical structure of BLPs, avoiding the need for classical KKT or value-function reformulations. The method was embedded within a robust Augmented Lagrangian framework, providing a theoretically sound and practical tool for solving this challenging class of optimization problems. Gradient evaluation via an adjoint system avoids explicit formation of the sensitivity Jacobian, reducing per-iteration cost independently of the upper-level dimension, and the computed KKT points are shown to be equivalent to S-stationary solutions of the associated MPCC reformulation under MPEC-LICQ.

Computational experiments on a suite of benchmark problems demonstrated the effectiveness and efficiency of the proposed method. The results highlighted the critical role of the architectural choice for the inner-loop solver (Algorithm 2); the use of a robust quasi-Newton method (L-BFGS-B) proved essential for handling the ill-conditioned and non-smooth subproblems that arise. The analysis of the problem landscapes confirmed the non-convexity inherent in BLPs, underscoring the necessity of a multi-start strategy. Furthermore, the implemented dual-criterion stopping condition proved to be a robust and efficient solution to the practical challenge of asymmetric convergence rates between primal and dual variables in the Augmented Lagrangian method.

Several directions remain open for future research. The most immediate is the extension to non-convex lower-level problems. Unlike global bounding approaches, which require certified global optimality of the lower level at every iteration, the sensitivity-based structure is amenable to extensions that admit locally optimal lower-level solutions — for instance through multi-start strategies at the lower level or branching on the lower-level feasible region — at the cost of weaker guarantees on the bilevel solution. A second direction concerns scalability: the benchmark suite used in this work consists of low-dimensional problems standard in the BLP literature, and the extension to larger-scale chemical engineering problems remains a primary avenue for future work, since the adjoint-based gradient structure offers the most significant computational advantage over reformulation-based approaches precisely in high upper-level dimensions. Additionally, exploring second-order update schemes for the dual variables could accelerate convergence and warrants further investigation. Finally, the multi-start procedure used here relies on manual exploration of the upper-level variable bounds; the development of a systematic initialization strategy adapted to the implicit structure of the bilevel feasible region remains an open direction.

CRediT authorship contribution statement

Eduardo Nolasco: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ross D. King:** Supervision. **Vassilios S. Vassiliadis:** Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first author acknowledges the CONACYT-SENER (Mexico) for financial support and Dr. Ehecatl Antonio del Río Chanona for helpful discussions on improving the presentation of the manuscript.

Data availability

No data was used for the research described in the article.

References

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M., 2019. CasADi: a software framework for nonlinear optimization and optimal control. Math. Program. Comput. 11 (1), 1–36. http://dx.doi.org/10.1007/s12532-018-0139-4, URL http://link.springer.com/10.1007/s12532-018-0139-4.

Avraamidou, S., Pistikopoulos, E.N., 2019. A multi-parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems. Comput. Chem. Eng. 125, 98–113. http://dx.doi.org/10.1016/j.compchemeng.2019.01.021, URL https://linkinghub.elsevier.com/retrieve/pii/S0098135418311530.

Avraamidou, S., Pistikopoulos, E., 2022. Multi-level Mixed-Integer Optimization. De Gruyter, http://dx.doi.org/10.1515/9783110760316, URL https://www.degruyter.com/document/doi/10.1515/9783110760316/html.

Beykal, B., Avraamidou, S., Pistikopoulos, I.P.E., Onel, M., Pistikopoulos, E.N., 2020. DOMINO: Data-driven optimization of bi-level mixed-integer nonlinear problems. J. Global Optim. 78 (1), 1–36. http://dx.doi.org/10.1007/s10898-020-00890-3.

Bracken, J., McGill, J.T., 1973. Mathematical programs with optimization problems in the constraints. Oper. Res. 21 (1), 37–44. http://dx.doi.org/10.1287/opre.21.1.37, URL https://pubsonline.informs.org/doi/10.1287/opre.21.1.37.

Byrd, R.H., Lu, P., Nocedal, J., Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. 16 (5), 1190–1208. http://dx.doi.org/10.1137/0916069, URL http://epubs.siam.org/doi/10.1137/0916069.

Clark, P.A., Westerberg, A.W., 1990. Bilevel programming for steady-state chemical process design-I. fundamentals and algorithms. Comput. Chem. Eng. 14 (1), 87–97. http://dx.doi.org/10.1016/0098-1354(90)87007-C, URL https://linkinghub.elsevier.com/retrieve/pii/009813549087007C.

Dempe, S., 2002. Foundations of bilevel programming. In: Nonconvex Optimization and Its Applications, vol. 61, Kluwer Academic Publishers, Boston, http://dx.doi.org/10.1007/b101970, URL http://link.springer.com/10.1007/b101970.

Dempe, S., Kalashnikov, V., Pérez-Valdés, G.A., Kalashnykova, N., 2015. Bilevel Programming Problems. In: Energy Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-662-45827-3, URL https://link.springer.com/10.1007/978-3-662-45827-3.

Djelassi, H., Glass, M., Mitsos, A., 2019. Discretization-based algorithms for generalized semi-infinite and bilevel programs with coupling equality constraints. J. Global Optim. 75 (2), 341–392. http://dx.doi.org/10.1007/s10898-019-00764-3.

Faísca, N.P., Dua, V., Rustem, B., Saraiva, P.M., Pistikopoulos, E.N., 2007. Parametric global optimisation for bilevel programming. J. Global Optim. 38 (4), 609–623. http://dx.doi.org/10.1007/s10898-006-9100-6, URL http://link.springer.com/10.1007/s10898-006-9100-6.

Garcia-Herreros, P., Zhang, L., Misra, P., Arslan, E., Mehta, S., Grossmann, I.E., 2016. Mixed-integer bilevel optimization for capacity planning with rational markets. Comput. Chem. Eng. 86, 33–47. http://dx.doi.org/10.1016/j.compchemeng.2015.12.007, URL https://linkinghub.elsevier.com/retrieve/pii/S0098135415003713.

Gümüş, Z.H., Floudas, C.A., 2001. Global optimization of nonlinear bilevel programming problems. J. Global Optim. 20 (1), 1–31. http://dx.doi.org/10.1023/A:1011268113791, URL https://link.springer.com/10.1023/A:1011268113791.

Hestenes, M.R., 1969. Multiplier and gradient methods. J. Optim. Theory Appl. 4 (5), 303–320. http://dx.doi.org/10.1007/BF00927673, URL http://link.springer.com/10.1007/BF00927673.

Islam, M.M., Singh, H.K., Ray, T., Sinha, A., 2017. An enhanced memetic algorithm for single-objective bilevel optimization problems. Evol. Comput. 25 (4), 607–642. http://dx.doi.org/10.1162/evco_a_00198, URL https://direct.mit.edu/evco/article/25/4/607-642/1055.

Kleniati, P.-M., Adjiman, C.S., 2014. Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. Part I: Theoretical development. J. Global Optim. 60 (3), 425–458. http://dx.doi.org/10.1007/s10898-013-0121-7, URL http://link.springer.com/10.1007/s10898-013-0121-7.

Kleniati, P.-M., Adjiman, C.S., 2015. A generalization of the branch-and-sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. Comput. Chem. Eng. 72, 373–386. http://dx.doi.org/10.1016/j.compchemeng.2014.06.004, URL https://linkinghub.elsevier.com/retrieve/pii/S0098135414001835.

Kočvara, M., Outrata, J.V., 2004. Optimization problems with equilibrium constraints and their numerical solution. Math. Program. 101 (1), 119–149. http://dx.doi.org/10.1007/s10107-004-0539-2, URL http://link.springer.com/10.1007/s10107-004-0539-2.

Lucchetti, R., Mignanego, F., Pieri, G., 1987. Existence theorems of equilibrium points in stackelberg. Optimization 18 (6), 857–866. http://dx.doi.org/10.1080/02331938708843300, URL http://www.tandfonline.com/doi/abs/10.1080/02331938708843300.

Luo, Z.-Q., Pang, J.-S., Ralph, D., 1996. Mathematical programs with equilibrium constraints. Cambridge University Press, http://dx.doi.org/10.1017/CBO9780511983658, URL https://www.cambridge.org/core/product/identifier/9780511983658/type/book.

Mirrlees, J.A., 1999. The theory of moral hazard and unobservable behaviour: Part i. Rev. Econ. Stud. 66 (1), 3–21. http://dx.doi.org/10.1111/1467-937X.00075, URL https://academic.oup.com/restud/article-lookup/doi/10.1111/1467-937X.00075.

Mitsos, A., 2010. Global solution of nonlinear mixed-integer bilevel programs. J. Global Optim. 47 (4), 557–582. http://dx.doi.org/10.1007/s10898-009-9479-y, URL http://link.springer.com/10.1007/s10898-009-9479-y.

Mitsos, A., Bollas, G.M., Barton, P.I., 2009. Bilevel optimization formulation for parameter estimation in liquid–liquid phase equilibrium problems. Chem. Eng. Sci. 64 (3), 548–559. http://dx.doi.org/10.1016/j.ces.2008.09.034, URL https://linkinghub.elsevier.com/retrieve/pii/S0009250908005113.

Mitsos, A., Lemonidis, P., Barton, P.I., 2008. Global solution of bilevel programs with a nonconvex inner program. J. Global Optim. 42 (4), 475–513.

Nocedal, J., Wright, S., 2006. Numerical optimization, second ed. In: Springer Series in Operations Research and Financial Engineering, Springer New York, p. 664. http://dx.doi.org/10.1007/978-0-387-40065-5.

Outrata, J.V., 1990. On the numerical solution of a class of stackelberg problems. ZOR - Methods Model. Oper. Res. 34 (4), 255–277. http://dx.doi.org/10.1007/BF01416737, URL http://link.springer.com/10.1007/BF01416737.

Powell, M.J.D., 1969. A method for nonlinear constraints in minimization problems. In: Fletcher, R. (Ed.), Optimization. Academic Press, New York, pp. 283–298.

Rockafellar, R.T., 1973. The multiplier method of hestenes and Powell applied to convex programming. J. Optim. Theory Appl. 12 (6), 555–562. http://dx.doi.org/10.1007/BF00934777, URL http://link.springer.com/10.1007/BF00934777.

Scheel, H., Scholtes, S., 2000. Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. Math. Oper. Res. 25 (1), 1–22.

Sinha, A., Shaikh, V., 2022. Solving bilevel optimization problems using Kriging approximations. IEEE Trans. Cybern. 52 (10), 10639–10654. http://dx.doi.org/10.1109/TCYB.2021.3061551, URL https://ieeexplore.ieee.org/document/9382956/.

von Stackelberg, H., 1934. Marktform und Gleichgewicht. Technical Report, Springer-Verlag, Berlin, English tr.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106 (1), 25–57. http://dx.doi.org/10.1007/s10107-004-0559-y, URL http://link.springer.com/10.1007/s10107-004-0559-y.

Ward, S., Zemkoho, A., Fisher, J., Benfield, D., Qian, Y., 2025. BOLIB: Bilevel optimisation library for benchmarking. Technical Report., University of Southampton.

Wiesemann, W., Tsoukalas, A., Kleniati, P.-M., Rustem, B., 2013. Pessimistic bilevel optimization. SIAM J. Optim. 23 (1), 353–380. http://dx.doi.org/10.1137/120864015, URL http://epubs.siam.org/doi/10.1137/120864015.

Yue, D., You, F., 2014. Game-theoretic modeling and optimization of multi-echelon supply chain design and operation under stackelberg game and market equilibrium. Comput. Chem. Eng. 71, 347–361. http://dx.doi.org/10.1016/j.compchemeng.2014.08.010.