

On the Complexity of Linear Programs with Parametric Constraint Matrices

Boveroux Laurie¹, Carvalho Margarida², Lodi Andrea³, and Louveaux Quentin¹

¹Department of Computer Science and Electrical Engineering,
University of Liège

²Department of Computer Science and Operations Research,
Université de Montréal

³Cornell Tech and Technion

Abstract

While sensitivity analysis for linear programs with parametric right-hand side vectors is well-established, cases involving parametric constraint matrices remain less explored. To address this gap, we investigate linear programs where the constraint matrix depends linearly on a scalar parameter. Our goal is to compute the minimum and maximum optimal values within a fixed parameter range. The computation of each of these optimal values corresponds to solving a bilevel program. We show that both bilevel optimization problems are NP-hard in general. We additionally identify three sufficient conditions on the parametric matrix under which the problems are no longer NP-hard. Under a feasibility condition, we introduce a reformulation for both problems and compare the execution time for solving them on standard benchmark instances. The experiments highlight that computing the parameter value corresponding to the minimum of the linear program is fast to solve, whereas computing the value corresponding to the maximum is not.

1 Introduction

Parametric linear programming studies how the optimal value of a linear program evolves under perturbations. While classical settings consider perturbations of the objective or the right-hand side, much less is known when the constraint matrix is affected. We study the case where a single scalar parameter modifies several coefficients of the constraint matrix. For each value of the scalar

parameter λ in a given interval, we consider the parametric linear program

$$\begin{aligned} f(\lambda) = \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & (A + \lambda D)\mathbf{x} \leq \mathbf{b}, \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the decision vector, $A \in \mathbb{R}^{m \times n}$ is the initial constraint matrix, $D \in \mathbb{R}^{m \times n}$ is the perturbation matrix associated with λ and $\mathbf{b} \in \mathbb{R}^m$ is the right-hand side vector. The function $f(\lambda)$ is called the *value function* of the parametric linear program. For instance, this setting appears when a single physical or economic parameter influences multiple constraints simultaneously. In time-indexed models of production or energy systems, parameters such as efficiency or resource availability appear repeatedly across constraints at each time period. Varying such a parameter, therefore affects multiple coefficients in the constraint matrix.

In such models, the objective typically represents total operating or investment costs or a performance metric. Understanding how the objective value evolves with respect to the parameter λ is interesting to assess the impact of uncertainty on the objective value and to identify the worst-case and best-case scenario. In particular, determining the global extrema of the associated value function over a given interval,

$$\min_{\lambda \in [\lambda_1, \lambda_2]} f(\lambda) \quad \text{and} \quad \max_{\lambda \in [\lambda_1, \lambda_2]} f(\lambda),$$

provides exact bounds on the achievable cost or performance as the parameter varies. These extrema lead to bilevel formulations, where λ is selected at the upper level and the linear program is solved at the lower level. However, the function $f(\lambda)$ may be nonconvex, making these problems difficult to solve. An illustrative example of a nonconvex function is shown in Figure 1.

1.1 Related work

Parametric and sensitivity analysis in linear programming study how optimal solutions and objective values evolve when problem data are perturbed. Studies mainly focus on variations in the objective coefficients or in the right-hand side. Perturbations on the constraint matrix have received less attention. Among the works that explicitly consider parameter-dependent constraint matrices, Zuidwijk [1] studies linear programs whose constraint matrix depends on a scalar parameter and derives local expressions for the evolution of the optimal value. His analysis relies on the characterization of regions in which a given basis remains optimal, leading to a piecewise description of the value function. Similarly, Khalilpour and Karimi [2] consider uncertainty on the left-hand side and propose a parametric solution approach based on basis-dependent computations.

More recently, Derval et al. [3] propose bounding procedures to estimate the value function over a parameter interval, yielding computable upper and lower bounds. Derval et al. [4] study the evaluation of optimal solutions and objective

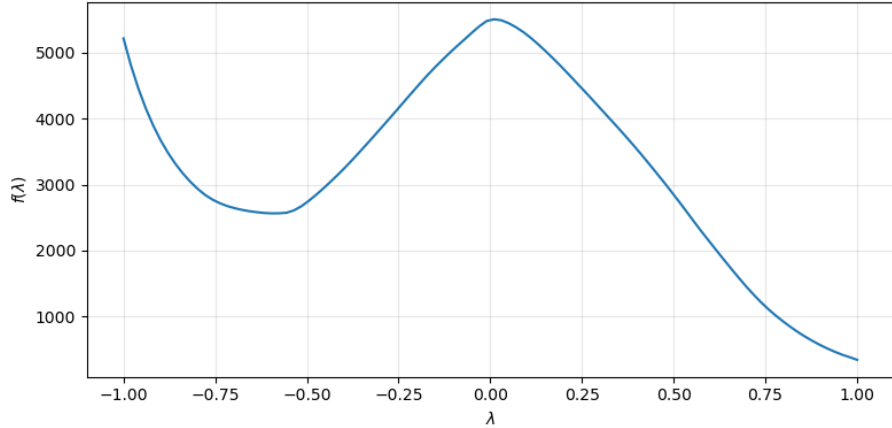


Figure 1: Illustrative example of a nonconvex function $f(\lambda)$.

values for a finite set of parameter values and develop warm-start strategies to accelerate repeated solves.

A related line of research is robust optimization [5], which seeks solutions that remain feasible under all realizations of uncertain data. While this framework is also motivated by uncertainty in the constraint matrix coefficients, its goal is different from ours. In robust optimization, the decision maker searches for a solution that performs well against all admissible perturbations. However, our objective is to characterize the best- and worst-case values reached by the optimal value function of the linear program when the scalar parameter varies over a given interval.

Our work is also connected to bilevel optimization. Indeed, optimizing the value function with respect to the parameter yields a bilevel formulation, where the upper-level variable is the scalar parameter λ and the lower level corresponds to a linear program. From a complexity perspective, the decision version of bilevel linear optimization problems are NP-complete. Jeroslow established NP-hardness [6] and Buchheim [7] showed that the problem belongs to NP via polynomial-size KKT-based reformulations. This NP-hardness persists under other settings such as min-max variants of bilevel optimization problems (add ref). Moreover, bilevel optimization problems remains NP-hard even when the upper level involves a single decision variable [8]. However, our problem does not fall directly into the class of bilevel linear optimization problems. Indeed, the dependence on λ appears in the constraint matrix of the lower-level problem and breaks the linearity. Thus, existing complexity results for bilevel linear optimization cannot be directly transferred to our setting. Related complexity analyses have been conducted for bilevel problems involving bilinear terms, such as pricing [9]. In these formulations, the bilinearities arise in the objective function from interactions between distinct decision variables, whereas in our

setting they involve a single scalar parameter λ that affects the feasible region.

1.2 Our contributions

In this paper, we study parametric linear programs in which the constraint matrix depends linearly on a scalar parameter and investigate the problem of computing the parameter values that minimize and maximize the optimal value of the resulting linear program over a given interval. We show that both corresponding bilevel formulations are NP-hard in general by providing polynomial reductions. We then identify three sufficient conditions on the parametric matrix under which the problem is no longer NP-hard. We also derive single-level reformulations for both problems, including a formulation based on strong duality for the max–min problem and a feasibility check relying on Farkas’ lemma to ensure its validity. Finally, we provide computational evidence on benchmark instances proposed by Derval et al. [3], showing that computing the minimum can be done efficiently, whereas computing the maximum remains significantly more challenging.

2 Problem statement

We consider a parametric linear optimization problem in which the constraint matrix depends linearly on a scalar parameter λ . For each $\lambda \in \Lambda \subset \mathbb{R}$, we define the problem

$$\begin{aligned} \mathcal{P}(\lambda) \equiv \quad & \min \mathbf{c}^T \mathbf{x} \\ & \text{s.t. } A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned} \tag{1}$$

and denote by $f(\lambda)$ its optimal value. We are interested in computing the global extrema over $\Lambda = [\lambda_1, \lambda_2]$. This leads to the following optimization problems:

$$\min_{\lambda \in [\lambda_1, \lambda_2]} \min_{\mathbf{x} \geq 0} \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b} \} \tag{2}$$

and

$$\max_{\lambda \in [\lambda_1, \lambda_2]} \min_{\mathbf{x} \geq 0} \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b} \}. \tag{3}$$

These formulations can be interpreted as bilevel optimization problems in which the upper level selects the parameter λ , while the lower level solves the linear program $\mathcal{P}(\lambda)$.

2.1 Auxiliary reformulations

We first observe that the min–min problem (2) can be rewritten as a single-level problem. Since the upper and lower levels have the same objective function, the bilevel formulation reduces to a joint minimization over $(\mathbf{x}, \lambda)$:

$$\min_{\mathbf{x}, \lambda} \mathbf{c}^T \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \lambda_1 \leq \lambda \leq \lambda_2.$$

In contrast, the max–min problem cannot be reduced to a joint optimization problem, since the lower-level variable must be optimal for the value of λ chosen at the upper level. To handle this structure, we rely on a dual reformulation of the lower-level problem. For a fixed value of λ , assume that the linear program $\mathcal{P}(\lambda)$ is feasible for all λ and has a finite optimal value. Its dual is

$$\max_{\mathbf{y} \geq \mathbf{0}} \{ \mathbf{b}^\top \mathbf{y} \mid (A + \lambda D)^\top \mathbf{y} \leq \mathbf{c} \}.$$

By strong duality, we therefore have

$$\min_{\mathbf{x} \geq \mathbf{0}} \{ \mathbf{c}^\top \mathbf{x} \mid (A + \lambda D)\mathbf{x} \leq \mathbf{b} \} = \max_{\mathbf{y} \geq \mathbf{0}} \{ \mathbf{b}^\top \mathbf{y} \mid (A + \lambda D)^\top \mathbf{y} \leq \mathbf{c} \}.$$

These two auxiliary reformulations will be used throughout the paper. The joint reformulation of the min–min problem is used in the complexity analysis and in the computational experiments. The dual reformulation of the lower-level problem is used in the complexity analysis of the max–min problem and in the derivation of its single-level formulation.

3 Complexity

In this section, we show that both the min–min (2) and max–min (3) problems are NP-hard in general. Our proofs rely on polynomial reductions from the linear multiplicative programming problem. We then identify restricted assumptions on D under which the min–min problem is no longer NP-hard.

3.1 NP-hardness of the min–min problem

As shown in Section 2.1, the bilevel formulation (2) is equivalent to the following single-level problem

$$\min_{\mathbf{x}, \lambda} \mathbf{c}^\top \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \lambda_1 \leq \lambda \leq \lambda_2.$$

Therefore, it is sufficient to prove NP-hardness of this single-level formulation.

Theorem 1. *The problem*

$$(\mathcal{P}_{min}) \quad \min_{\mathbf{x}, \lambda} \mathbf{c}^\top \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \lambda_1 \leq \lambda \leq \lambda_2,$$

is NP-hard.

Proof. We prove NP-hardness via a polynomial reduction from Matsui’s NP-hard family for linear multiplicative programming [10]:

$$(P_1(M)) \quad \min z_1 z_2$$

with decision variables

$$\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{R}^n, \quad \mathbf{v} = (v_{ij})_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}, \quad u^0, v^0, z_1, z_2 \in \mathbb{R},$$

subject to

$$0 \leq u_i \leq 1 \quad (i = 1, \dots, n), \quad (4)$$

$$0 \leq v_{ij} \leq 1 \quad (i, j = 1, \dots, n), \quad (5)$$

$$v_{ij} \leq u_i, \quad v_{ij} \leq u_j, \quad v_{ij} \geq u_i + u_j - 1 \quad (i \neq j), \quad (6)$$

$$v_{ii} = u_i \quad (i = 1, \dots, n), \quad (7)$$

$$M\mathbf{u} = \mathbf{1}, \quad (8)$$

$$u^0 = \sum_{i=1}^n p^i u_i, \quad (9)$$

$$v^0 = \sum_{i=1}^n \sum_{j=1}^n p^{i+j} v_{ij}, \quad (10)$$

$$z_1 = v^0 - p + 2p^{4n} + 2p^{2n}u^0, \quad (11)$$

$$z_2 = v^0 - p + 2p^{4n} - 2p^{2n}u^0, \quad (12)$$

where M an $m \times n$ binary matrix with $n > m$ and $p = n^{n^4}$.

Matsui proves that

$$\text{opt}(P_1(M)) \leq 4p^{8n} \iff M\mathbf{u} = \mathbf{1} \text{ has a 0-1 solution,}$$

and, because the problem on the right-hand side of the equivalence is NP-complete, the linear multiplicative program in the left-hand side is also NP-hard.

From this instance we construct an instance of $(\mathcal{P}_{min})$ with decision vector

$$\mathbf{x} = (s, u_1, \dots, u_n, v_{11}, \dots, v_{nn}, u^0, v^0, z_2, t).$$

and parameter λ representing z_1 . The objective is $\mathbf{c}^\top \mathbf{x} = t$.

All linear constraints (4)–(10) and (12) are included in $A\mathbf{x} \leq \mathbf{b}$, together with $s = 1$ enforced by $s \leq 1, -s \leq -1$.

The relation (11) is encoded as

$$\lambda s = v^0 - p + 2p^{4n} + 2p^{2n}u^0,$$

i.e.,

$$\begin{aligned} \lambda s - v^0 - 2p^{2n}u^0 &\leq -p + 2p^{4n}, \\ -\lambda s + v^0 + 2p^{2n}u^0 &\leq p - 2p^{4n}. \end{aligned}$$

The epigraph constraint

$$-t + \lambda z_2 \leq 0$$

enforces $t \geq \lambda z_2$.

From the bounds on u and v , we obtain

$$-p + 2p^{4n} \leq \lambda \leq n^2 p^{2n} - p + 2p^{4n} + 2np^{3n}.$$

For $n \geq 5$, this yields $\lambda \in [p^{4n}, 5p^{4n}]$ and we set $\lambda_1 = p^{4n}$, $\lambda_2 = 5p^{4n}$. These bounds preserve all feasible values arising from $(P_1(M))$. Moreover, λ_1 and λ_2 are polynomially encodable, since $\log p = O(n^4 \log n)$ [10].

Given any feasible solution of $(P_1(M))$, define $\mathbf{x} = (1, \mathbf{u}, \mathbf{v}, u^0, v^0, z_2, z_1 z_2)$ and $\lambda = z_1$. Then $(\mathbf{x}, \lambda)$ is feasible for $(\mathcal{P}_{\min})$ and $\mathbf{c}^\top \mathbf{x} = z_1 z_2$, hence

$$\text{opt}(\mathcal{P}_{\min}) \leq \text{opt}(P_1(M)).$$

Conversely, let $(\mathbf{x}, \lambda)$ be optimal for $(\mathcal{P}_{\min})$. Then $s = 1$ and the constraints imply

$$\lambda = v^0 - p + 2p^{4n} + 2p^{2n}u^0.$$

Setting $z_1 = \lambda$, we obtain a feasible solution of $(P_1(M))$. Since t appears only in the objective and in $-t + \lambda z_2 \leq 0$, optimality implies $t = \lambda z_2 = z_1 z_2$, hence

$$\text{opt}(P_1(M)) \leq \text{opt}(\mathcal{P}_{\min}).$$

Thus $\text{opt}(\mathcal{P}_{\min}) = \text{opt}(P_1(M))$ and NP-hardness follows. $\square$

3.2 NP-hardness of the max–min problem

In contrast with the min–min formulation, which reduces to a single-level optimization problem, the max–min formulation remains bilevel. Since the min–min problem is effectively single-level, the max–min problem, which involves a true bilevel structure, is expected to be at least NP-hard. The following theorem establishes this result.

Theorem 2. *The problem*

$$\max_{\lambda \in [\lambda_1, \lambda_2]} \min_{\mathbf{x} \geq \mathbf{0}} \{ \mathbf{c}^\top \mathbf{x} \mid A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b} \} \quad (13)$$

is NP-hard.

Proof. We prove the NP-hardness via a polynomial reduction from the min–min problem (Theorem 1). Consider an instance of the min–min problem of the form

$$\min_{\lambda \in [\lambda_1, \lambda_2]} \min_{\boldsymbol{\mu} \geq \mathbf{0}} \{ \mathbf{c}_M^\top \boldsymbol{\mu} \mid A_M \boldsymbol{\mu} + \lambda D_M \boldsymbol{\mu} \leq b_M \}.$$

By Matsui’s construction [10, Corollary 2.3], the instances derived from $P_1(M)$ are always feasible and admit a finite optimal value. Indeed, for any instance M of the set partition problem, Matsui constructs a corresponding optimization problem $P_1(M)$ whose feasible region is nonempty and bounded. Consider now the max–min problem

$$\max_{\lambda \in [\lambda_1, \lambda_2]} \min_{\mathbf{x} \geq \mathbf{0}} \{ \mathbf{c}^\top \mathbf{x} \mid (A + \lambda D)\mathbf{x} \leq \mathbf{b} \}.$$

For a fixed value of λ , the inner minimization problem is a linear program. Using the dual reformulation introduced in Section 2.1, it can be expressed equivalently as

$$-\min_{\boldsymbol{\mu} \geq \mathbf{0}} \{ \mathbf{b}^\top \boldsymbol{\mu} \mid -A^\top \boldsymbol{\mu} + \lambda(-D)^\top \boldsymbol{\mu} \leq \mathbf{c} \}.$$

We now choose the data $(A, D, \mathbf{b}, \mathbf{c})$ so that this dual problem coincides with the min-min instance above. Setting

$$\mathbf{b} := \mathbf{c}_M, \quad -A^\top := A_M, \quad -D^\top := D_M, \quad \mathbf{c} := \mathbf{b}_M,$$

the inner problem becomes

$$-\min_{\boldsymbol{\mu} \geq \mathbf{0}} \{c_M^\top \boldsymbol{\mu} \mid A_M \boldsymbol{\mu} + \lambda D_M \boldsymbol{\mu} \leq \mathbf{b}_M\}.$$

By substituting this expression into the max-min formulation, we have

$$\begin{aligned} & \max_{\lambda \in [\lambda_1, \lambda_2]} \min_{\mathbf{x} \geq \mathbf{0}} \{c^\top \mathbf{x} \mid (A + \lambda D)\mathbf{x} \leq \mathbf{b}\} \\ &= \max_{\lambda \in [\lambda_1, \lambda_2]} \left(-\min_{\boldsymbol{\mu} \geq \mathbf{0}} \{c_M^\top \boldsymbol{\mu} \mid A_M \boldsymbol{\mu} + \lambda D_M \boldsymbol{\mu} \leq \mathbf{b}_M\} \right) \\ &= -\min_{\lambda \in [\lambda_1, \lambda_2]} \min_{\boldsymbol{\mu} \geq \mathbf{0}} \{c_M^\top \boldsymbol{\mu} \mid A_M \boldsymbol{\mu} + \lambda D_M \boldsymbol{\mu} \leq \mathbf{b}_M\}. \end{aligned}$$

This shows the max-min problem reconstructs the negative of the NP-hard min-min problem. $\square$

3.3 Tractable cases

We now identify restricted assumptions on D under which the problem of finding either the minimum or the maximum is no longer NP-hard.

Uniform sign in D

We consider the case where all coefficients of D have the same sign. In this setting, the value function $f(\lambda)$ is monotone on the interval $[\lambda_1, \lambda_2]$. Then, the optimal values are attained at the endpoints of the interval.

Proposition 1. *Assume that all entries of D have the same sign.*

- *If $D_{ij} \geq 0 \forall i, j$, then $f(\lambda)$ is nondecreasing on $[\lambda_1, \lambda_2]$.*
- *If $D_{ij} \leq 0 \forall i, j$, then $f(\lambda)$ is nonincreasing on $[\lambda_1, \lambda_2]$.*

Proof. Let $\lambda', \lambda'' \in [\lambda_1, \lambda_2]$ with $\lambda' \leq \lambda''$.

Assume first that $D_{ij} \geq 0 \forall i, j$. Since $\mathbf{x} \geq \mathbf{0}$, we have $D\mathbf{x} \geq \mathbf{0}$ and therefore

$$\lambda' D\mathbf{x} \leq \lambda'' D\mathbf{x}.$$

Hence,

$$A\mathbf{x} + \lambda' D\mathbf{x} \leq A\mathbf{x} + \lambda'' D\mathbf{x}.$$

It follows that every feasible solution for λ'' is also feasible for λ' , i.e.,

$$F(\lambda'') \subseteq F(\lambda'),$$

where $F(\lambda)$ denotes the feasible set of $\mathcal{P}(\lambda)$. Therefore,

$$f(\lambda') \leq f(\lambda''),$$

showing that f is nondecreasing on $[\lambda_1, \lambda_2]$.

The case $D_{ij} \leq 0 \forall i, j$ follows similarly and yields that f is nonincreasing on $[\lambda_1, \lambda_2]$. $\square$

Corollary 1. *Assume that all entries of D have the same sign.*

- *If $D_{ij} \geq 0 \forall i, j$, then*

$$\min_{\lambda \in [\lambda_1, \lambda_2]} f(\lambda) = f(\lambda_1), \quad \max_{\lambda \in [\lambda_1, \lambda_2]} f(\lambda) = f(\lambda_2).$$

- *If $D_{ij} \leq 0 \forall i, j$, then*

$$\min_{\lambda \in [\lambda_1, \lambda_2]} f(\lambda) = f(\lambda_2), \quad \max_{\lambda \in [\lambda_1, \lambda_2]} f(\lambda) = f(\lambda_1).$$

Consequently, problems (2) and (3) are polynomially solvable.

Exactly one non-zero row in D

We consider the case where D has exactly one non-zero row. This implies that only one constraint, denoted by i_0 , is affected by the parameter λ . The problem can then be written as

$$\begin{aligned} (\mathcal{P}_{min, row}) \quad & \min_{\mathbf{x}, \lambda} \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & B\mathbf{x} \leq \boldsymbol{\beta}, \\ & \mathbf{a}_{i_0}\mathbf{x} + \lambda \mathbf{d}_{i_0}\mathbf{x} \leq b_{i_0}, \\ & \mathbf{x} \geq \mathbf{0}, \\ & \lambda \in [\lambda_1, \lambda_2]. \end{aligned}$$

In this setting, the value function $f(\lambda)$ is quasi-concave on $[\lambda_1, \lambda_2]$. As a consequence, the minimum over λ is reached at one of the bounds λ_1 or λ_2 .

Definition 1. *A function f is quasi-concave on an interval if for any λ', λ'' in the interval and any $\alpha \in [0, 1]$,*

$$f(\alpha\lambda' + (1 - \alpha)\lambda'') \geq \min\{f(\lambda'), f(\lambda'')\}.$$

Proposition 2. *If D has exactly one non-zero row, then the value function $f(\lambda)$ is quasi-concave on $[\lambda_1, \lambda_2]$.*

Proof. Let $\lambda', \lambda'' \in [\lambda_1, \lambda_2]$ and define

$$\hat{\lambda} = \alpha\lambda' + (1 - \alpha)\lambda'', \quad \alpha \in [0, 1].$$

Let $\mathbf{x}^* \geq \mathbf{0}$ be an optimal solution at $\hat{\lambda}$. Then

$$B\mathbf{x}^* \leq \beta, \quad (\mathbf{a}_{i_0} + \hat{\lambda}\mathbf{d}_{i_0})\mathbf{x}^* \leq b_{i_0}.$$

Suppose, by contradiction, that $\mathbf{x}^*$ is infeasible for both λ' and λ'' . Then

$$(\mathbf{a}_{i_0} + \lambda'\mathbf{d}_{i_0})\mathbf{x}^* > b_{i_0} \quad \text{and} \quad (\mathbf{a}_{i_0} + \lambda''\mathbf{d}_{i_0})\mathbf{x}^* > b_{i_0}$$

Multiplying the first inequality by $\alpha \geq 0$ and the second by $1 - \alpha \geq 0$, and summing, leads to

$$(\mathbf{a}_{i_0} + \hat{\lambda}\mathbf{d}_{i_0})\mathbf{x}^* > b_{i_0},$$

which contradicts feasibility at $\hat{\lambda}$.

Hence $\mathbf{x}^*$ is feasible for at least one of the problems associated with λ' or λ'' . If it is feasible for λ' , then by optimality of $f(\lambda')$ in a minimization problem,

$$f(\lambda') \leq \mathbf{c}^\top \mathbf{x}^* = f(\hat{\lambda}).$$

Similarly, if it is feasible for λ'' , then

$$f(\lambda'') \leq \mathbf{c}^\top \mathbf{x}^* = f(\hat{\lambda}).$$

Therefore,

$$f(\hat{\lambda}) \geq \min\{f(\lambda'), f(\lambda'')\},$$

which proves quasi-concavity. $\square$

Since $f(\lambda)$ is quasi-concave on the interval $[\lambda_1, \lambda_2]$, any minimum of f over this interval is reached at a bound of the interval. Therefore,

$$\min_{\lambda \in [\lambda_1, \lambda_2]} f(\lambda) = \min\{f(\lambda_1), f(\lambda_2)\}.$$

As a result, the optimal value can be obtained by evaluating the problem only at the values λ_1 and λ_2 and selecting the best solution.

Observation: the proof extends to the mixed-integer case.

Exactly one non-zero column in D

We consider the case where D has exactly one non-zero column. This implies that only one variable, denoted by x_0 , is affected by the parameter λ . We decompose the variable vector as $(x_0, \mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^{n-1}$. Let A_0 denote the corresponding column of A and $\mathbf{d}$ the non-zero column of D . The problem can then be written as

$$\begin{aligned} (\mathcal{P}_{min,col}) \quad & \min_{x_0, \mathbf{x}, \lambda} \mathbf{c}^\top \mathbf{x} + c_0 x_0 \\ \text{s.t.} \quad & A\mathbf{x} + x_0(A_0 + \lambda\mathbf{d}) \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \quad x_0 \geq 0, \\ & \lambda \in [\lambda_1, \lambda_2]. \end{aligned}$$

In this setting, the value function $f(\lambda)$ is quasi-convex on $[\lambda_1, \lambda_2]$, which implies that the problem is no longer NP-hard.

Definition 2. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is called quasi-convex if its domain $\text{dom}(f)$ and all its sublevel sets

$$S_t = \{x \in \text{dom}(f) \mid f(x) \leq t\}, \quad \forall t \in \mathbb{R},$$

are convex.

Proof. Let

$$S_t = \{\lambda \in [\lambda_1, \lambda_2] \mid f(\lambda) \leq t\}.$$

We show that S_t is convex.

Take $\lambda', \lambda'' \in S_t$. Then $f(\lambda') \leq t$ and $f(\lambda'') \leq t$. Hence, there exist feasible solutions $(\mathbf{x}', x'_0)$ for λ' and $(\mathbf{x}'', x''_0)$ for λ'' , such that

$$\mathbf{c}^\top \mathbf{x}' + c_0 x'_0 \leq t, \quad \mathbf{c}^\top \mathbf{x}'' + c_0 x''_0 \leq t,$$

and

$$A\mathbf{x}' + (A_0 + \lambda' \mathbf{d})x'_0 \leq \mathbf{b}, \quad A\mathbf{x}'' + (A_0 + \lambda'' \mathbf{d})x''_0 \leq \mathbf{b}.$$

Let $\alpha \in [0, 1]$ and define $\hat{\lambda} = \alpha\lambda' + (1 - \alpha)\lambda''$. We now construct a feasible solution associated with $\hat{\lambda}$.

Define

$$\theta = \frac{\alpha x''_0}{\alpha x''_0 + (1 - \alpha)x'_0}, \quad \text{and therefore} \quad 1 - \theta = \frac{(1 - \alpha)x'_0}{\alpha x''_0 + (1 - \alpha)x'_0}.$$

Set

$$\hat{\mathbf{x}} = \theta \mathbf{x}' + (1 - \theta) \mathbf{x}'' \quad \text{and} \quad \hat{x}_0 = \theta x'_0 + (1 - \theta) x''_0.$$

By construction,

$$\hat{x}_0 = \frac{x'_0 x''_0}{\alpha x''_0 + (1 - \alpha)x'_0}.$$

Moreover,

$$\hat{\lambda} \hat{x}_0 = (\alpha \lambda' + (1 - \alpha) \lambda'') \frac{x'_0 x''_0}{\alpha x''_0 + (1 - \alpha) x'_0}.$$

On the other hand,

$$\theta \lambda' x'_0 + (1 - \theta) \lambda'' x''_0 = \frac{\alpha \lambda' x'_0 x''_0 + (1 - \alpha) \lambda'' x'_0 x''_0}{\alpha x''_0 + (1 - \alpha) x'_0}.$$

Thus,

$$\hat{\lambda} \hat{x}_0 = \theta \lambda' x'_0 + (1 - \theta) \lambda'' x''_0.$$

We now verify feasibility:

$$A\hat{\mathbf{x}} + (A_0 + \hat{\lambda} \mathbf{d})\hat{x}_0.$$

Using the definitions above,

$$\begin{aligned} A\hat{\mathbf{x}} + (A_0 + \hat{\lambda} \mathbf{d})\hat{x}_0 &= A(\theta \mathbf{x}' + (1 - \theta) \mathbf{x}'') \\ &\quad + A_0(\theta x'_0 + (1 - \theta) x''_0) \\ &\quad + \mathbf{d} \hat{\lambda} \hat{x}_0. \end{aligned}$$

Since

$$\hat{\lambda}\hat{x}_0 = \theta\lambda'x'_0 + (1-\theta)\lambda''x''_0,$$

we obtain

$$\begin{aligned} A\hat{\mathbf{x}} + (A_0 + \hat{\lambda}\mathbf{d})\hat{x}_0 &= \theta(A\mathbf{x}' + (A_0 + \lambda'\mathbf{d})x'_0) \\ &\quad + (1-\theta)(A\mathbf{x}'' + (A_0 + \lambda''\mathbf{d})x''_0). \end{aligned}$$

Because both terms are feasible,

$$A\mathbf{x}' + (A_0 + \lambda'\mathbf{d})x'_0 \leq \mathbf{b}, \quad \text{and} \quad A\mathbf{x}'' + (A_0 + \lambda''\mathbf{d})x''_0 \leq \mathbf{b},$$

we conclude that

$$A\hat{\mathbf{x}} + (A_0 + \hat{\lambda}\mathbf{d})\hat{x}_0 \leq \mathbf{b}.$$

Hence, $(\hat{\mathbf{x}}, \hat{x}_0)$ is feasible for $\hat{\lambda}$.

Finally,

$$\begin{aligned} \mathbf{c}^\top \hat{\mathbf{x}} + c_0 \hat{x}_0 &= \mathbf{c}^\top (\theta\mathbf{x}' + (1-\theta)\mathbf{x}'') + c_0(\theta x'_0 + (1-\theta)x''_0) \\ &= \theta(\mathbf{c}^\top \mathbf{x}' + c_0 x'_0) + (1-\theta)(\mathbf{c}^\top \mathbf{x}'' + c_0 x''_0) \\ &\leq \theta t + (1-\theta)t \\ &= t. \end{aligned}$$

Therefore,

$$f(\hat{\lambda}) \leq t,$$

which shows that $\hat{\lambda} \in S_t$.

Hence, every sublevel set S_t is convex and therefore f is quasi-convex.

Remark. If $x'_0 = x''_0 = 0$, the definition of θ is not valid. However, in this case the bilinear term disappears and feasibility follows directly by convexity of the remaining linear constraints. $\square$

Observation: The proof does not extend to the mixed-integer case. The constructed point $(\hat{\mathbf{x}}, \hat{x}_0)$ is obtained as a convex combination of two feasible solutions. Hence, even if the initial solutions are integer, there is no guarantee that $(\hat{\mathbf{x}}, \hat{x}_0)$ is integer as well.

4 Single-level Reformulations

The bilevel formulations (2) and (3) are not directly usable by standard optimization solvers. We therefore rely on the reformulations introduced in Section 2.1 to derive equivalent single-level problems that can be handled by non-linear optimization solvers.

We first consider the min-min problem. As established in Section 2.1, it is equivalent to a joint minimization problem in $(\mathbf{x}, \lambda)$. We therefore directly use the following single-level formulation.

The min–min problem is equivalent to

$$\min_{\mathbf{x}, \lambda} \mathbf{c}^\top \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \lambda_1 \leq \lambda \leq \lambda_2. \quad (14)$$

We then consider the max–min problem (3). In contrast with the min–min case, this formulation remains bilevel and cannot be reduced to a joint optimization problem. Following the reformulation introduced in Section 2.1, we replace the lower-level problem by its dual and enforce optimality via strong duality. This yields the following single-level formulation.

Assume that, for every $\lambda \in [\lambda_1, \lambda_2]$, the lower-level problem is feasible and bounded. Then the max–min problem is equivalent to

$$\begin{aligned} \max_{\lambda, \mathbf{x}, \mathbf{y}} \quad & \mathbf{b}^\top \mathbf{y} \\ \text{s.t.} \quad & A\mathbf{x} + \lambda D\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \\ & (A + \lambda D)^\top \mathbf{y} \leq \mathbf{c}, \quad \mathbf{y} \leq \mathbf{0}, \\ & \mathbf{c}^\top \mathbf{x} = \mathbf{b}^\top \mathbf{y}, \\ & \lambda_1 \leq \lambda \leq \lambda_2. \end{aligned} \quad (15)$$

The above reformulation relies on the lower-level problem being feasible for all values of λ and having a finite optimal value. To verify this, we use Farkas' lemma [11], which provides a certificate of infeasibility. Applying the lemma to the system

$$(A + \lambda D)\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

we obtain that infeasibility occurs for some λ if and only if there exist λ and $\mathbf{y}$ such that

$$(A + \lambda D)^\top \mathbf{y} \geq \mathbf{0}, \quad \mathbf{y}^\top \mathbf{b} = -1, \quad \mathbf{y} \geq \mathbf{0}.$$

This leads to a feasibility problem over the interval $[\lambda_1, \lambda_2]$, whose solvability determines whether such a certificate exists.

The lower-level is feasible for all $\lambda \in [\lambda_1, \lambda_2]$ if and only if the problem

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & (A + \lambda D)^\top \mathbf{y} \geq \mathbf{0}, \\ & \mathbf{y}^\top \mathbf{b} = -1, \\ & \mathbf{y} \geq \mathbf{0}, \\ & \lambda_1 \leq \lambda \leq \lambda_2 \end{aligned} \quad (16)$$

is infeasible.

Statistic	Vars.	Total Constr.	$Ax = b$	$Ax \leq b$	$(A + \lambda D)x = b$	$(A + \lambda D)x \leq b$
Min	79	153	0	14	0	0
Median	1175	1966	287	1486	18	82
Mean	2028	3240	495	2644	35	65
Max	13525	24025	3115	21025	100	100

Table 1: Size statistics for the 191 benchmark instances.

If this problem (16) is infeasible, no certificate of infeasibility exists and the lower-level problem is feasible for every $\lambda \in [\lambda_1, \lambda_2]$. In that case, the reformulation (15) is valid.

5 Experiments

The following experiments provide computational evidence on benchmark instances proposed by Derval et al. [3]. The benchmark contains 191 parametric linear bilevel instances derived from Netlib problems. The constraints are of the forms The constraints of the problems are of the forms

$$Ax = b, \quad Ax \leq b, \quad (A + \lambda D)x = b, \quad (A + \lambda D)x \leq b,$$

with exactly 100 constraints depending on the parameter λ . The instances cover a broad range of dimensions. The number of variables ranges from 79 to 13 525, with a median of 1 175 variables. The total number of constraints ranges from 14 to more than 24 000. Table 1 summarizes the distribution of equality and inequality constraints, distinguishing between constraints that are independent of λ and constraints with λ -dependent coefficients. For each instance, we consider the same interval for the parameter λ and solve the three following models:

- i. the min–min formulation (14), which computes the minimum of the value function over the interval,
- ii. the feasibility problem (16), which searches for a Farkas certificate proving that the lower-level problem is infeasible for some value of λ ,
- iii. the max–min formulation (15), which computes the maximum of the value function over the interval.

All models are implemented in Python and solved with Gurobi. Bilinear terms of the form λx_i and λy_i are modeled explicitly and handled by the solver. Each run has a time limit of 1200 seconds. The outcome of the feasibility problem is reported below directly in terms of the lower-level problem. When (16) is infeasible, the lower-level problem is feasible for all values of λ in the interval and the strong-duality reformulation can be used. When (16) is optimal, the lower-level problem is infeasible for at least one value of λ in the interval. In this case, the strong-duality reformulation is not valid over the whole interval and we conclude that the upper-bound is $+\infty$.

Minimum	Lower-level feasibility	Maximum	#instances
optimal	feasible for all λ	optimal	90
optimal	time limit	time limit	49
optimal	feasible for all λ	time limit	31
infeasible or unbounded	infeasible for some λ	skipped	10
optimal	time limit	optimal	3
optimal	infeasible for some λ	skipped	3
optimal	time limit	unbounded	2
optimal	feasible for all λ	suboptimal	1
infeasible	infeasible for some λ	skipped	1
suboptimal	feasible for all λ	time limit	1

Table 2: Joint solver outcomes for the minimum formulation, the feasibility check, and the maximum formulation over the 191 instances.

Model and status	#runs	Median time (s)	Mean time (s)	Max time (s)
Minimum, optimal	179	0.23	2.11	53.03
Feasibility, lower-level feasible	123	0.06	10.68	667.03
Feasibility, lower-level infeasible	14	0.07	7.02	95.07
Feasibility, time limit	54	1200	1200	1200
Maximum, optimal	93	1.77	38.69	1052.72
Maximum, time limit	81	1200	1200	1200

Table 3: Runtime statistics by model and solver status.

Table 2 shows that the minimum problem is usually easy: 179 instances (93.7%) are solved to proven optimality and one run ends with a suboptimal status. The remaining 11 instances (5.7%) correspond to instances that are infeasible or unbounded for the minimum formulation. The feasibility check proves that the lower-level problem is feasible for all values of λ in 123 instances (64.4%), proves that it is infeasible for some value of λ in 14 instances (7.3%) and reaches the time limit in 54 instances (28.3%). The maximum formulation is solved to optimality in 93 instances (48.7 %), reaches the time limit in 81 instances (42.4%), is unbounded in 2 instances (1.0%) and is skipped in the 14 instances (7.3%) where the lower-level problem is not feasible over the whole interval. These results show that the minimum formulation is solved reliably, while the maximum formulation is much harder, with many time limits. The feasibility check can also be a bottleneck, since it fails to conclude for 28.3% of the instances.

The runtime statistics in Table 3 confirm the same contrast. When the minimum formulation is solved to optimality, the median runtime is 0.23 seconds and all optimal minimum runs finish in less than one minute. The feasibility check is also often very fast when it terminates: both the median time for proving feasibility over the whole interval and the median time for proving infeasibility

for some value of λ are below one tenth of a second. However, the feasibility check is not uniformly easy: 54 runs reach the 20-minute time limit. The maximum formulation is much harder. Even among the 93 optimal maximum runs, the mean runtime is 38.69 seconds and the slowest solved instance takes around 18 minutes. Among the 81 runs that were stopped at the time limit, only 33 ended with a finite optimality gap. The remaining 48 runs still had an infinite gap at termination.

In conclusion, the experiments indicate that the min–min formulation is well handled by the solver on this benchmark set. The max–min formulation is less consistent. It can be solved to optimality on many instances, but many runs remain unsolved after 20 minutes.

6 Conclusion

In this paper, we consider parametric linear programs with constraint matrices depending linearly on a scalar parameter. We study the problem of computing the minimum and maximum of the associated value function over a bounded interval. We prove that both corresponding bilevel problems are NP-hard in general and we identify restricted assumptions under which they are no longer NP-hard. We then derive single-level formulations for both problems. The minimum problem admits a direct bilinear reformulation. The maximum problem requires a strong duality-based reformulation together with a feasibility condition that must hold over the entire parameter interval. The computational study shows a difference in computational difficulty between the two problems. On the tested instances, the minimum problem is solved efficiently in nearly all cases. In contrast, the maximum problem remains harder. Many instances cannot be solved within the time limit and verifying the feasibility condition may itself become a computational bottleneck.

References

- [1] Rob Zuidwijk. Linear parametric sensitivity analysis of the constraint co-efficient matrix in linear programs. Technical report, 2005.
- [2] Rajab Khalilpour and IA Karimi. Parametric optimization with uncertainty on the left hand side of linear programs. *Computers & chemical engineering*, 60:31–40, 2014.
- [3] Guillaume Derval, Damien Ernst, Quentin Louveaux, and Bardhyl Miftari. Sensitivity analysis for linear changes of the constraint matrix of a linear program. *arXiv preprint arXiv:2410.14443*, 2024.
- [4] Guillaume Derval, Bardhyl Miftari, Damien Ernst, and Quentin Louveaux. Efficient lp warmstarting for linear modifications of the constraint matrix. *arXiv preprint arXiv:2501.04151*, 2025.

- [5] Aharon Ben-Tal, Arkadi Nemirovski, and Laurent El Ghaoui. Robust optimization. 2009.
- [6] Robert G Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical programming*, 32(2):146–164, 1985.
- [7] Christoph Buchheim. Bilevel linear optimization belongs to np and admits polynomial-size kkt-based reformulations. *Operations Research Letters*, 51(6):618–622, 2023.
- [8] Nagisa Sugishita and Margarida Carvalho. Complexity of bilevel linear programming with a single upper-level variable. *arXiv preprint arXiv:2510.21126*, 2025.
- [9] Christoph Grüne, Dorothee Henke, Eva Rotenberg, and Lasse Wulf. The complexity of stackelberg pricing games. *arXiv preprint arXiv:2511.05700*, 2025.
- [10] Tomomi Matsui. Np-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization*, 9(2):113–119, 1996.
- [11] David P. Williamson. Orie 6300: Mathematical programming i - lecture 7, 2008. Course lecture notes.