Explaining First-Stage Decisions of Two-Stage Stochastic Programs: Modeling, Reformulations and Solution Methods

Adrien Belfer[1,*] Luce Brotcorne[1] Henri Lefebvre[2]
Marius Roland[1]

[1] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France.
[2] LIRMM, CNRS, University of Montpellier, Montpellier, France.

[*] Corresponding author: adrien.belfer@inria.fr

Abstract

Explainability is a growing concern for optimization models used as decision-making tools. It aims to help a stakeholder understand why a model recommends a given outcome, typically by exhibiting the elements of the input that account for this outcome. We study this question for linear two-stage stochastic programs, and explain the returned decision by a small set of scenarios under which it remains optimal. Selecting these scenarios amounts to reweighting the scenario probabilities, and leads to a bilevel model whose lower level is a two-stage stochastic program. We show that a decision can always be explained by at most one more scenario than the number of first-stage variables, whatever the size of the scenario set. We propose three bilevel models that generalize the only previous approach, and solve them with an exact single-level reformulation strengthened by valid inequalities, a decomposition method, and heuristics based on the optimality conditions of the lower level. Computational experiments show the benefits of the bilevel models and of the valid inequalities, which solve a third of the instances that the reformulation alone leaves unsolved. They also show the efficiency of the heuristic based on strong duality, which decides within seconds, on most instances, whether the decision can be explained with a prescribed number of scenarios.

Keywords: explainability, stochastic programming, bilevel optimization, scenario reduction

1 Introduction

Optimization models increasingly play a central role in supporting decisions with real-world consequences, ranging from production planning to the design and management of logistics

networks. As these models grow in scope and complexity, understanding the rationale behind a recommended decision becomes as important as computing the decision itself. A recent survey on practical advances of mixed-integer linear programming highlights the need for such understanding as one of the major open challenges in optimization-based decision support systems (Clautiaux and Ljubić 2025). Access to the objective function and constraints of a model does not suffice to understand its solutions when the model is large. Aydın et al. (2026) call explainability the additional operations that help a stakeholder understand, evaluate and challenge an outcome when the underlying rationale is too complex to grasp. This paper contributes to explainability in mathematical optimization.

Two-Stage Stochastic Programs (TSSPs) are widely used in practice, for instance in supply chain network design (Santoso et al. 2005) or disaster management (Graß and Fischer 2016). A TSSP accounts for parameter uncertainty by minimizing the expected cost over a finite set of scenarios, each associated with a given probability. Decisions are split into two stages. First-stage decisions are made before the uncertainty is revealed. Second-stage, or recourse, decisions adapt to the realized scenario. A first-stage decision therefore depends on the entire set of scenarios and their associated probabilities. This dependence is difficult to interpret, as the scenarios can be numerous.

This paper studies the explainability of first-stage decisions in linear TSSPs through retrospective scenario reduction. Scenario reduction is a classical idea in stochastic programming, and is usually applied before a TSSP is solved. The aim of such approaches is to replace the scenario set by a smaller one to reduce the effort required to solve the TSSP (Dupačová et al. 2003). Retrospective scenario reduction reverses this order. Given a first-stage solution $\tilde{x}$ obtained by solving the TSSP, we seek a probability vector supported on a small subset of scenarios, under which $\tilde{x}$, or a point close to it, is optimal. The selected scenarios then provide an explanation of $\tilde{x}$: they suffice, on their own, to justify the decision. Since only the probabilities are modified, the explanation is expressed in terms of scenarios already familiar to the decision-maker. This idea was introduced by Rathi et al. (2024). We here formulate it as a bilevel program, whose upper level selects a probability vector and whose lower level returns the induced optimal first-stage decision.

1.1 Related Literature

Explainability in mathematical optimization is a young research direction, surveyed by Lumbreras and Ciller (2025) and Aydın et al. (2026). One of the approaches identified by Lumbreras and Ciller (2025) is simplification and surrogate modeling. Our work belongs to this approach, as it provides a simplified representation of the uncertainty of a TSSP. Existing methods either build explainability into the model (Goerigk and Hartisch 2023, Ferrer-Mestres et al. 2024), or explain solutions of specific problem classes, such as scheduling or shortest path problems (Čyras et al. 2019, Schild et al. 2025). None of them addresses TSSPs. Two other directions, however, stand closer to our work: counterfactual explanations, which

also rely on bilevel optimization, and scenario reduction.

Counterfactual explanations seek a minimal modification of the input of a model that makes an alternative, preferred outcome optimal. The concept originates in explainable machine learning (Wachter et al. 2018), where such explanations can be computed via optimization models (Maragno et al. 2024). In optimization, such explanations were studied for binary problems in which only the objective coefficients are modified (Korikov and Beck 2021, 2023), for linear programs (Kurtz et al. 2026, Lefebvre and Schmidt 2026), and for mixed-integer linear programs (Engelhardt et al. 2026). These approaches typically involve bilevel models: the leader modifies the data of the problem, and the follower solves the resulting optimization problem. We depart from this line of work at both levels: the follower solves a TSSP rather than a deterministic program, while the leader modifies the scenario probabilities only. A related line of work considers counterfactual explanations in contextual optimization, where the data of an optimization problem is predicted from features, called the context, by a machine-learning model. In this setting, counterfactual explanations modify the context rather than the problem data (Forel et al. 2023, Ramírez-Ayerbe and Frejinger 2026). In particular, Forel et al. (2023) explain a first-stage decision of a TSSP. However, their explanation is expressed as a modification of the context rather than of the probabilities. Moreover, the new optimal decision is fixed, in contrast to our setting where it is left as a variable of the lower-level problem, making the problem bilevel and harder to solve. Finding a modification of the input under which a fixed solution becomes optimal is also the object of inverse optimization (Ahuja and Orlin 2001), from which the heuristics of Section 5 derive.

The second direction, scenario reduction, aims to lower the computational cost of solving a stochastic program while controlling the resulting loss in solution quality. Building on stability results, one line of work selects a reduced distribution that minimizes a probability distance to the original one (Dupačová et al. 2003, Heitsch and Römisch 2003, Rujeerapaiboon et al. 2022). A more recent line groups scenarios according to the decisions or costs they induce (Bertsimas and Mundru 2023, Hewitt et al. 2022, Keutchayan et al. 2023). Zhang et al. (2023) instead select a prescribed number of scenarios, so as to minimize the approximation error of the recourse function over a pool of first-stage solutions. These methods could be applied to retrospective scenario reduction. However, they target computational efficiency and ignore the decision to be explained, which is known in our setting. Section 5.1 uses the first line of work as a baseline.

Our first result on retrospective scenario reduction shows that a decision can always be recovered with a number of scenarios that does not depend on the size of the scenario set (Section 2.3). Bounds of a similar nature exist for reduced problems built from new scenarios, either a single aggregated scenario (Homem-de-Mello et al. 2026) or the clusters of a partition (Song and Luedtke 2015). Section 2.3 compares them with ours.

To the best of our knowledge, retrospective scenario reduction has only been studied

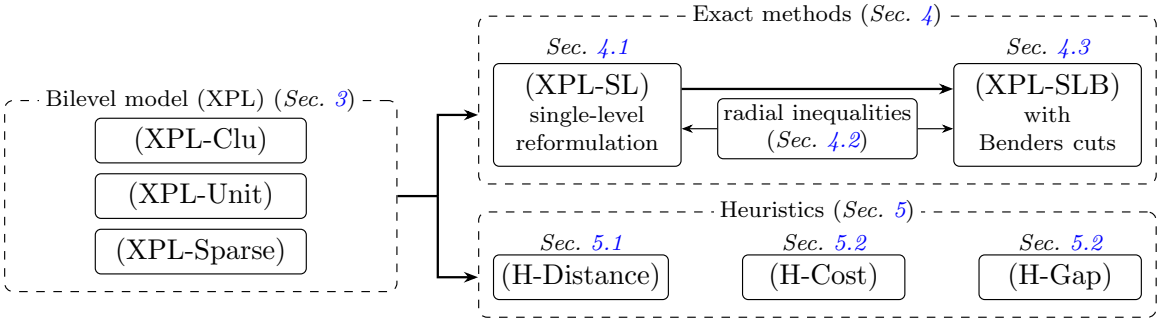


Figure 1: Models and solution methods developed in this paper, with the sections in which they are introduced.

by Rathi et al. (2024). Their approach consists of two steps. They first build a partition of the scenario set using the optimal second-stage decisions of each scenario. A bilevel model then selects one representative scenario per cluster, and assigns it the total probability of that cluster. We generalize this approach with models that require no partition beforehand, and propose new solution methods.

1.2 Contributions

Our contributions are the following.

1. We formulate retrospective scenario reduction as a bilevel problem. We show that there always exists a reduced distribution, supported on a number of scenarios independent of the size of the scenario set, under which $\tilde{x}$ remains optimal. We then propose three models that differ in how the probabilities may be redistributed.

2. We derive an exact single-level reformulation of the bilevel problem, and a decomposition method that exploits the scenario structure of the lower-level TSSP. For two of the three models, we strengthen the reformulation with radial inequalities, separated through a preprocessing performed scenario by scenario.

3. We propose three heuristics: a baseline inherited from classical scenario reduction, and two heuristics based on optimality conditions.

4. We assess these methods on three families of TSSPs adapted from the literature, and compare our models with the approach of Rathi et al. (2024).

Section 2 formulates the problem and bounds the number of scenarios needed to recover $\tilde{x}$. Figure 1 summarizes the models and methods developed in Sections 3 to 5, and the section in which each of them is introduced. Section 6 reports the computational study, and Section 7 concludes.

2 Retrospective Scenario Reduction

This section formulates the problem studied in this paper. Section 2.1 recalls TSSPs and introduces our notation. Section 2.2 defines retrospective scenario reduction. Section 2.3 bounds the number of scenarios that suffice to recover a first-stage decision.

2.1 Two-Stage Stochastic Programs

The scenario-based formulation of a linear TSSP is defined as

$$
\begin{aligned}
(\mathrm{TSSP}(p)) \quad \min_{x,y} \quad & c^\top x + \sum_{s\in S} p^s (f^s)^\top y^s && \text{(2.1a)}\\
\text{s.t.} \quad & Ax = b, && \text{(2.1b)}\\
& T^s x + W^s y^s = h^s, \quad s \in S, && \text{(2.1c)}\\
& x \geq 0, \quad y^s \geq 0, \quad s \in S. && \text{(2.1d)}
\end{aligned}
$$

Here, S denotes a finite set of scenarios. Each scenario $s \in S$ is associated with the data (f^s, T^s, W^s, h^s) and some probability $p^s \in [0,1]$. The probability vector $p := (p^s)_{s\in S}$ is an element of the probability simplex $\mathcal{P}_0 := \Big\{p \in [0,1]^{|S|} : \sum_{s\in S} p^s = 1\Big\}$.

The variables $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2\times|S|}$ represent the first-stage and second-stage decisions, respectively. In what follows, we let m_1 denote the number of first-stage constraints (2.1b), and m_2 the number of second-stage constraints (2.1c) per scenario. We also define the feasible set of first-stage decisions as $X := \{x \geq 0 : Ax = b\}$.

For a given first-stage decision $x \in X$, Problem (2.1) decomposes across scenarios. The recourse function for scenario $s \in S$ and first-stage decision x is

$$
Q^s(x) := \min_{y^s} \big\{ (f^s)^\top y^s : W^s y^s = h^s - T^s x,\ y^s \geq 0 \big\}. \tag{2.2}
$$

With this notation, the set of optimal first-stage decisions of $(\mathrm{TSSP}(p))$ is

$$
X^*(p) := \arg\min_{x\in X} \Big\{ c^\top x + \sum_{s\in S} p^s Q^s(x) \Big\}. \tag{2.3}
$$

To ensure that $Q^s(x)$ is finite for every $x \in X$, we make the following assumption. Note that this assumption does not depend on the probabilities and remains valid when p is modified.

Assumption 2.1 (Relatively complete and bounded recourse)**.** For every scenario $s \in S$ and every first-stage decision $x \in X$, the recourse problem (2.2) is feasible and bounded below.

2.2 Problem Statement

Throughout the remainder of this work, we assume that a TSSP has already been solved with probability vector $\tilde{p}$, yielding a solution $(\tilde{x}, \tilde{y})$. We refer to this problem as the initial TSSP.

Retrospective scenario reduction seeks a small set of scenarios that, on its own, justifies $\tilde{x}$. Given a target integer $k \in \{1, \ldots, |S| - 1\}$, it asks whether there exists a probability vector p supported on at most k scenarios such that $\tilde{x}$, or a point close to it, is optimal for (TSSP(p)). Under Assumption 2.1, any such vector defines a reduced TSSP: the scenarios carrying zero probability can be removed from (2.1) without changing the set of feasible first-stage decisions. Modifying the probabilities may, however, result in a TSSP without an optimal solution, or one whose optimal solutions lie arbitrarily far from $\tilde{x}$. We rule out both situations with the following assumption, which holds in particular when X is bounded.

Assumption 2.2 (Nonempty bounded optimal solution sets)**.** There exists a bounded set $B \subseteq \mathbb{R}^{n_1}$ such that, for every $p \in \mathcal{P}_0$, the set $X^*(p)$ is nonempty and contained in B.

To illustrate the problem and provide further insight, consider the following example.

Example 2.3 (The farmer's problem (Birge and Louveaux 2011))**.** A farmer allocates 500 acres among three crops before the yields are known, and buys or sells the crops once the yields are realized. We use the original data (Appendix D.1), with $|S| = 9$ equally likely scenarios representing good, average, and poor yield conditions. Each category contains three scenarios: yields are 1.45, 1.5 or 1.55 times their mean in the good scenarios, 0.95, 1 or 1.05 times in the average ones, and 0.45, 0.5 or 0.55 times in the poor ones. The optimal allocation $\tilde{x}$ balances these nine scenarios, making it hard to understand.

Reducing the distribution to three equally likely scenarios, one per category, leads to a simpler problem under which the same allocation remains optimal, provided the scenarios are well chosen: here, the good scenario of factor 1.45, the average one of factor 1, and the poor one of factor 0.5. The median scenario of each category, for instance, leads to a different allocation. The farmer can then understand $\tilde{x}$ by reasoning on three cases, a good, an average and a poor year, rather than on nine.

2.3 How Many Scenarios Suffice?

Although the initial problem may involve many scenarios, only a limited number may be needed to preserve the optimality of $\tilde{x}$. The following result provides an upper bound on this number.

Proposition 2.4. *Let Assumption 2.1 hold and let $\tilde{x} \in X^*(\tilde{p})$. Set*

$$\rho := \dim\big(M \ker A\big), \qquad \text{where} \quad M := \begin{bmatrix} T^1 \\ \vdots \\ T^{|S|} \end{bmatrix} \in \mathbb{R}^{|S|m_2 \times n_1},$$

and $M \ker A := \{Md : d \in \ker A\}$ denotes the image of $\ker A$ under M. Then there exists a probability vector p supported on at most $\rho + 1$ scenarios such that $\tilde{x} \in X^*(p)$.

The proof, given in Appendix A, applies Carathéodory's theorem to the first-order optimality condition of (TSSP($\tilde{p}$)) at $\tilde{x}$. Although M has $|S|$ blocks, we always have $\rho \leq \dim \ker A \leq n_1$, and even $\rho \leq n_1 - m_1$ when A has full row rank. The number of scenarios needed to explain a decision thus depends on the scenario set, through the matrices T^s, but it is bounded by a quantity independent of its size. Retrospective scenario reduction is therefore feasible as soon as $k \geq \rho + 1$. In Example 2.3, $\rho = 3$. Proposition 2.4 thus guarantees that $\tilde{x}$ can be explained by at most four scenarios, whereas the reduction given in Example 2.3 uses three. The bound is therefore not always attained.

Two results of a similar nature exist, but they rely on new scenarios. Homem-de-Mello et al. (2026) show that, for TSSPs with fixed recourse matrix and second-stage costs, any optimal first-stage decision is also optimal for the problem defined by a single scenario, which need not belong to S. Adaptive partition-based methods (van Ackooij et al. 2018, Song and Luedtke 2015, Roland et al. 2025) aggregate the scenarios of each cluster of a partition into one. For programs with simple recourse, Song and Luedtke (2015) exhibit a partition whose size does not depend on $|S|$ and that preserves the optimal value. Proposition 2.4 differs from both results as it keeps a subset of the original scenarios, with modified probabilities, under which a given decision remains optimal. The selected scenarios are thus familiar to the decision-maker, which is the purpose of retrospective scenario reduction. In addition, our result applies to any TSSP satisfying Assumption 2.1, whatever the structure of the recourse problem and the position of the uncertainty.

3 Bilevel Models for Retrospective Scenario Reduction

Proposition 2.4 has two limitations. First, it provides no guarantee below the threshold: when $k < \rho + 1$, a probability vector supported on at most k scenarios that preserves the optimality of $\tilde{x}$ may not exist. Second, the existence result does not ensure that the resulting probability vector is interpretable. In particular, it may differ substantially from $\tilde{p}$, and the probabilities may lack a structure that is meaningful to the decision-maker. This section therefore addresses retrospective scenario reduction through a bilevel problem.

Let $\mathcal{P} \subseteq \mathcal{P}_0$ be a set of admissible probability vectors, typically supported on at most k scenarios but possibly having additional structure. Let $d : \mathcal{P} \times X \to \mathbb{R}$ be a function,

to be minimized, that measures the quality of a reduction, typically through the distance between the induced first-stage decision and $\tilde{x}$. We propose to solve

$$\begin{aligned} (\text{XPL}) \quad \min_{p,\, x} \quad & d(p, x) && \text{(3.1a)} \\ \text{s.t.} \quad & p \in \mathcal{P}, && \text{(3.1b)} \\ & x \in X^*(p). && \text{(3.1c)} \end{aligned}$$

Problem (XPL) is a bilevel optimization problem (Dempe 2002). Such problems describe a hierarchy between two decision-makers with distinct objectives. A leader decides first, here by selecting a probability vector p, and a follower reacts by solving an optimization problem parameterized by that decision, here (TSSP(p)). This hierarchy is imposed by the question of interest, since evaluating $p \in \mathcal{P}$ requires solving (TSSP(p)).

A difficulty arises when the follower has several optimal decisions, that is, when $X^*(p)$ is not a singleton. The value $d(p, x)$ then depends on which one is considered. We adopt the optimistic perspective (Dempe 2002, Sec. 3.3): the minimization in (3.1a) runs over both p and x, so (XPL) considers the decision $x \in X^*(p)$ of smallest value $d(p, x)$. Section 7 discusses an alternative perspective. The next proposition, proved in Appendix A, gives sufficient conditions for (XPL) to be well posed.

Proposition 3.1. *Let Assumptions 2.1 and 2.2 hold, let $\mathcal{P} \subseteq \mathcal{P}_0$ be nonempty and closed, and let d be lower semicontinuous. Then, Problem (3.1) has a solution.*

We now specify the objective function d and the admissible set $\mathcal{P}$ we consider.

Objective function. Retrospective scenario reduction asks for a probability vector under which the new optimal first-stage decision is $\tilde{x}$, or a point close to it. We therefore measure the distance of the induced decision to $\tilde{x}$, through $d_x(x, \tilde{x}) := \|x - \tilde{x}\|_1$. However, when asking for a reduction, a decision-maker may want to keep a reduced problem faithful to the initial one. We therefore also measure the distance between p and $\tilde{p}$, through the Wasserstein distance d_p of order 1, a standard tool in scenario reduction (Dupačová et al. 2003). The Wasserstein distance is defined by a cost matrix, whose entries give the cost of transporting probability mass from scenario s to scenario s'. We here consider the cost $\|\xi^s - \xi^{s'}\|_2$ between their data, where ξ^s stacks the entries of f^s, T^s, W^s and h^s, each standardized across the scenarios so that no entry dominates because of its units. We propose the objective function

$$d : (p, x) \mapsto \alpha\, d_x(x, \tilde{x}) + (1 - \alpha)\, d_p(p, \tilde{p}), \tag{3.2}$$

where $\alpha \in (0, 1]$ balances the two terms.

An alternative model would require the induced decision to equal $\tilde{x}$, or to lie in a neighborhood of it. Such a constraint couples the two levels, and renders (XPL) infeasible

whenever no admissible p recovers $\tilde{x}$, which Proposition 2.4 does not exclude when $k < \rho+1$. We therefore measure the distance to $\tilde{x}$ in the objective function.

Admissible sets. Different decision-makers may favor different structures for the reduced probability distribution. To accommodate these preferences, we introduce three admissible sets, defined in terms of the support of a probability vector p, given by $\text{supp}(p) := \{s \in S : p^s > 0\}$, which identifies the scenarios retained in the reduction.

The first set follows the construction of Rathi et al. (2024), and starts from an *a priori* partition of the scenarios. One scenario can be selected per cluster, and receives the total probability of that cluster. The resulting explanation reads directly: each selected scenario stands for its cluster. Such structure suits a decision-maker who already has a grouping in mind. In Example 2.3, for instance, the farmer may want the explanation to preserve the distinction between good, average and poor yields.

Let $\mathcal{C} = \{C_1, \dots, C_k\}$ be a partition of S into k clusters. This admissible set reads

$$\mathcal{P}^{\mathcal{C}}_{\text{cluster}} := \Big\{p \in \mathcal{P}_0 : |\text{supp}(p) \cap C| = 1 \text{ and } \textstyle\sum_{s\in C} p^s = \sum_{s\in C} \tilde{p}^s,\ \forall C \in \mathcal{C}\Big\}.$$

A decision-maker may have no such grouping in mind. The second set keeps the intuition of a partition yet without fixing it beforehand: each selected scenario represents a subset of the original scenarios, without requiring this subset to be specified in advance. This interpretation relies on the following assumption, standard in the sample average approximation (Shapiro et al. 2021, Chap. 5).

Assumption 3.2 (Uniform probabilities)**.** The initial probability vector is uniform, that is, $\tilde{p}^s = \frac{1}{|S|}$ for all $s \in S$.

Each selected scenario then receives the total probability of the scenarios it represents, that is, an integer multiple of $\frac{1}{|S|}$:

$$\mathcal{P}_{\text{unit}} := \big\{p \in \mathcal{P}_0 : |\text{supp}(p)| \le k,\ |S|\, p^s \in \mathbb{N},\ s \in S\big\}.$$

A decision-maker may finally only care about which scenarios matter, and not how they represent the others. The third set therefore removes the restrictions above on the redistribution of the probability mass, and only bounds the number of selected scenarios:

$$\mathcal{P}_{\text{sparse}} := \big\{p \in \mathcal{P}_0 : |\text{supp}(p)| \le k\big\}.$$

The explanation then no longer reads as a set of representatives, but as the k scenarios that matter, each weighted by its relative importance in supporting $\tilde{x}$.

We denote by (XPL-Clu), (XPL-Unit) and (XPL-Sparse) the model (XPL) equipped with the objective function (3.2) and the set $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$, $\mathcal{P}_{\text{unit}}$ and $\mathcal{P}_{\text{sparse}}$, respectively. Each admissible set, as well as the two distances $d_x(x, \tilde{x})$ and $d_p(p, \tilde{p})$, can be described with

Table 1: Number of variables and linear constraints describing the admissible sets (left) and the distances (right), detailed in Appendix B. The constant $\tilde{p}^s_{\mathcal{C}}$ is the initial probability of the cluster containing s.

Model	#Bin.	#Int.	#Cont.	#Constr.	p^s
(XPL-Clu)	$\lvert S\rvert$ (v^s)	–	–	k	$\tilde{p}^s_{\mathcal{C}}\, v^s$
(XPL-Unit)	$\lvert S\rvert$ (v^s)	$\lvert S\rvert$ $(0 \le u^s \le \lvert S\rvert)$	–	$\lvert S\rvert + 2$	$u^s/\lvert S\rvert$
(XPL-Sparse)	$\lvert S\rvert$ (v^s)	–	$\lvert S\rvert$ (p^s)	$\lvert S\rvert + 2$	p^s

Distance	#Cont.	#Constr.
$\lVert x - \tilde{x}\rVert_1$	n_1	$2n_1$
$d_p(p, \tilde{p})$	$\lvert S\rvert^2$	$2\lvert S\rvert$

auxiliary variables, binary, integer or continuous, and linear constraints, of size polynomial in $|S|$ and n_1. Table 1 summarizes these descriptions, which are detailed in Appendix B. In particular, it gives the expression of p^s in each set. In the remainder of the paper, these variables and constraints are added implicitly. The three models are thus bilevel problems with a mixed-integer linear upper level and a linear lower level. Each of the three models admits an optimal solution, by Proposition 3.1, since the three admissible sets are nonempty and closed (Appendix B), and the objective function (3.2) is continuous. The three sets are also nested, as proved in Appendix A.4.

Proposition 3.3. *Under Assumption 3.2, for every partition $\mathcal{C}$ of S into k subsets, $\mathcal{P}^{\mathcal{C}}_{\text{cluster}} \subseteq \mathcal{P}_{\text{unit}} \subseteq \mathcal{P}_{\text{sparse}}$, and both inclusions are strict if $2 \le k < |S|$.*

Model (XPL-Sparse) thus explores the largest set of probability vectors. When $k \ge \rho + 1$, $\mathcal{P}_{\text{sparse}}$ is moreover the only set guaranteed to contain a probability vector under which $\tilde{x}$ is optimal (Proposition 2.4). The restrictions of $\mathcal{P}_{\text{unit}}$ and $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ may, however, improve interpretability. Moreover, the theoretical advantage of (XPL-Sparse) comes at a computational cost, as discussed in the following sections.

4 Exact Solution Methods

In this section, we propose exact methods for (XPL), which therefore apply to (XPL-Clu), (XPL-Unit) and (XPL-Sparse). Throughout this section, Assumptions 2.1 and 2.2 hold, $\mathcal{P}$ is nonempty and closed and d is lower semicontinuous, so that (XPL) admits an optimal solution (Proposition 3.1). Section 4.1 derives a single-level reformulation, denoted (XPL-SL), and linearizes it for (XPL-Clu) and (XPL-Unit). Section 4.2 strengthens the linear relaxation of the linearized model with valid inequalities. Section 4.3 exploits the structure of the lower-level TSSP through a decomposition method.

4.1 A Single-Level Reformulation

The lower-level problem of (XPL) is (TSSP(p)), which is a linear program. A first-stage decision x is thus optimal for (TSSP(p)) if and only if it is part of a primal feasible solution (x, y) whose objective value equals that of some dual feasible solution (λ, μ). Replacing the

lower-level problem by these conditions (Dempe 2002, Sec. 3.5) yields the reformulation

$$
\begin{aligned}
\text{(XPL-SL)} \quad \min_{p,x,y,\lambda,\mu} \quad & d(p,x) && \text{(4.1a)}\\
\text{s.t.} \quad & p \in \mathcal{P}, && \text{(4.1b)}\\
& A^\top \lambda + \sum_{s\in S} (T^s)^\top \mu^s \le c, && \text{(4.1c)}\\
& (W^s)^\top \mu^s \le p^s f^s, \quad s \in S, && \text{(4.1d)}\\
& Ax = b, && \text{(4.1e)}\\
& T^s x + W^s y^s = h^s, \quad s \in S, && \text{(4.1f)}\\
& c^\top x + \sum_{s\in S} p^s (f^s)^\top y^s \le b^\top \lambda + \sum_{s\in S} (h^s)^\top \mu^s, && \text{(4.1g)}\\
& x \ge 0, \quad y^s \ge 0, \quad s \in S, && \text{(4.1h)}\\
& \lambda,\ \mu \text{ free.} && \text{(4.1i)}
\end{aligned}
$$

Constraints (4.1c)–(4.1d) enforce dual feasibility of (λ, μ), (4.1e)–(4.1f) primal feasibility of (x, y), and (4.1g) strong duality.

With the variables and constraints of Table 1, the reformulations of (XPL-Clu), (XPL-Unit) and (XPL-Sparse) are nonconvex Mixed-Integer Bilinear Programs (MIBLPs). The only nonlinear constraint is (4.1g), through the products $p^s(f^s)^\top y^s$. For (XPL-Clu) and (XPL-Unit), p^s is defined with a binary or integer variable (see Table 1). These products can be linearized exactly, provided that bounds on $(f^s)^\top y^s$ are available.

Linearizing the bilinear terms. We now derive such bounds, then detail the linearizations.

Proposition 4.1. *Let Assumptions 2.1 and 2.2 hold, and let* $X_{\mathcal{P}} := \{x \in X : \exists\, p \in \mathcal{P},\ x \in X^*(p)\}$ *be the set of first-stage decisions that are optimal for at least one* $p \in \mathcal{P}$*. Define*

$$
\underline{Q}^s := \inf_{x\in X_{\mathcal{P}}} Q^s(x), \qquad \overline{Q}^s := \sup_{x\in X_{\mathcal{P}}} Q^s(x), \qquad s \in S.
$$

These bounds are finite, and at least one optimal solution (p, x, y, λ, μ) *of (XPL-SL) satisfies*

$$
\underline{Q}^s \le (f^s)^\top y^s \le \overline{Q}^s, \qquad s \in S. \tag{4.2}
$$

By Proposition 4.1, constraints (4.2) can be added to (XPL-SL) without changing its optimal value. Since $X_{\mathcal{P}}$ has no explicit description, $\underline{Q}^s$ and $\overline{Q}^s$ are not directly computable. Any finite bounds $L^s \le \underline{Q}^s$ and $U^s \ge \overline{Q}^s$ can be used instead. For example, when X is bounded, the infimum and supremum of Q^s over X can be used. In our experiments, both bounds are derived from the structure of each instance family (Appendix D).

With these bounds, the products $p^s(f^s)^\top y^s$ of (4.1g) can be linearized exactly for (XPL-Clu) and (XPL-Unit). In both models, each probability can be written as a weighted sum of binary variables, $p^s = \sum_{j\in J} \omega_j^s \beta_j^s$, where J is a finite set and $\omega_j^s > 0$. For (XPL-Clu), J is a singleton, β^s is the variable v^s of Table 1, and $\omega^s = \tilde{p}_{\mathcal{C}}^s$. For (XPL-Unit), the integer variable u^s of Table 1 is first expanded into $N_{\text{bin}} := \lfloor \log_2 |S| \rfloor + 1$ binary variables (Glover 1975), so that $J = \{0, \dots, N_{\text{bin}} - 1\}$ and $\omega_j^s = 2^j/|S|$ (Appendix B.3). Each product $\beta_j^s(f^s)^\top y^s$, of a binary variable with a bounded expression, is then replaced by a variable z_j^s subject to four linear inequalities (Glover 1975). This yields the model

$$\min_{p,x,y,\lambda,\mu,\beta,z} \quad d(p,x) \tag{4.3a}$$

$$\text{s.t.} \quad (4.1b)\text{–}(4.1f),\ (4.1h),\ (4.1i), \tag{4.3b}$$

$$p^s = \textstyle\sum_{j\in J} \omega_j^s \beta_j^s, \qquad s \in S, \tag{4.3c}$$

$$c^\top x + \textstyle\sum_{s\in S}\sum_{j\in J} \omega_j^s z_j^s \le b^\top \lambda + \sum_{s\in S} (h^s)^\top \mu^s, \tag{4.3d}$$

$$L^s \beta_j^s \le z_j^s \le U^s \beta_j^s, \qquad s \in S,\ j \in J, \tag{4.3e}$$

$$(f^s)^\top y^s - U^s(1-\beta_j^s) \le z_j^s \le (f^s)^\top y^s - L^s(1-\beta_j^s), \qquad s \in S,\ j \in J, \tag{4.3f}$$

$$\beta_j^s \in \{0,1\}, \qquad s \in S,\ j \in J. \tag{4.3g}$$

Compared with (XPL-SL), the linearization adds $|S||J|$ continuous variables and $4|S||J|$ constraints, and, for (XPL-Unit), $|S|N_{\text{bin}}$ binary variables. Both models are then Mixed-Integer Linear Programs (MILPs). For (XPL-Sparse), p^s is a continuous variable, and we do not carry out linearization, so the model remains a MIBLP. All three formulations have a size polynomial in that of the TSSP and can be handled directly by solvers such as Gurobi (Gurobi Optimization, LLC 2026).

4.2 Radial Inequalities

In what follows, we derive valid inequalities that strengthen the linear relaxation of model (4.3).

In constraint (4.3d), the left-hand side is a linearization of the primal objective $c^\top x + \sum_{s\in S} p^s(f^s)^\top y^s$ of (TSSP(p)), performed with constraints (4.3e)–(4.3f). When $\beta \in \{0,1\}$, these constraints give $z_j^s = \beta_j^s (f^s)^\top y^s$, and the linearized expression equals the primal objective. However, when β is fractional, the linearized expression may underestimate the primal objective, and constraint (4.3d) then no longer forces $x \in X^*(p)$. The following inequalities bound the linearized expression from below by a function of the distance $\|x - \tilde{x}\|_1$.

Proposition 4.2 (Radial inequalities)**.** *For $\kappa \ge 0$ and $p \in \mathcal{P}_0$, let*

$$\phi^s(\kappa) := \min_{x\in X} \left\{ c^\top x + Q^s(x) + \kappa \|x - \tilde{x}\|_1 \right\}, \quad s \in S, \qquad \Phi(\kappa, p) := \sum_{s\in S} p^s \phi^s(\kappa).$$

These values are finite, and every feasible solution of (4.3) satisfies, for every $\kappa \geq 0$,

$$c^\top x + \sum_{s\in S}\sum_{j\in J}\omega_j^s z_j^s \;\geq\; \Phi(\kappa, p) - \kappa\|x - \tilde{x}\|_1. \tag{4.4}$$

The proof, given in Appendix A.6, follows from the definition of the functions ϕ^s. For fixed κ and p, the right-hand side of (4.4) is an affine function of the distance $\|x - \tilde{x}\|_1$, with slope $-\kappa$ and value $\Phi(\kappa, p)$ at $x = \tilde{x}$. We therefore call (4.4) a radial inequality, and κ its slope. The radial inequalities form an infinite family, with one inequality per slope $\kappa \geq 0$. We now analyze the properties of the function Φ to characterize this family.

Proposition 4.3. *There exists a finite set $\Gamma \subset (0, +\infty)$ such that, for every $p \in \mathcal{P}_0$:*

(i) the function $\Phi(\cdot, p)$ is concave, nondecreasing and piecewise linear on $\mathbb{R}_+$, with breakpoints in Γ;

(ii) $\Phi(\kappa, p) = c^\top \tilde{x} + \sum_{s\in S} p^s Q^s(\tilde{x})$ for every $\kappa \geq \bar{\kappa} := \max(\Gamma \cup \{0\})$.

The set Γ is the union of the sets Γ^s of breakpoints of the functions ϕ^s, $s \in S$ (Appendix A.7). Proposition 4.3 has three consequences.

Corollary 4.4. *The radial inequality (4.4) with $\kappa = \bar{\kappa}$ is tight at $\tilde{x}$, that is, every feasible solution of (4.3) with $x = \tilde{x}$ satisfies*

$$c^\top \tilde{x} + \sum_{s\in S}\sum_{j\in J}\omega_j^s z_j^s \;=\; \Phi(\bar{\kappa}, p) \;=\; c^\top \tilde{x} + \sum_{s\in S} p^s Q^s(\tilde{x}).$$

The right-hand side of this equation is the cost of $\tilde{x}$ in (TSSP(p)). In the linear relaxation of (4.3), the radial inequality with $\kappa = \bar{\kappa}$ keeps the linearized primal objective above this cost at $x = \tilde{x}$, even when β is fractional. By (4.3d) and weak duality, every point of the relaxation with $x = \tilde{x}$ then satisfies $\tilde{x} \in X^*(p)$.

Corollary 4.5. *Let (p, x, z) be a point of the linear relaxation of (4.3). Separating the radial inequalities amounts to maximizing $\Phi(\kappa, p) - \kappa\|x - \tilde{x}\|_1$ over $\kappa \geq 0$. This maximum is attained at a slope of $\Gamma \cup \{0\}$.*

Indeed, by Proposition 4.3, the function $\kappa \mapsto \Phi(\kappa, p) - \kappa\|x - \tilde{x}\|_1$ is concave and piecewise linear, with breakpoints in Γ, and nonincreasing beyond $\bar{\kappa}$.

In model (4.3), the distance $\|x - \tilde{x}\|_1$ is represented by auxiliary variables whose sum $\sum_i t_i$ is at least $\|x - \tilde{x}\|_1$ (Appendix B.1). Since $\kappa \geq 0$, every radial inequality remains valid with $\sum_i t_i$ in place of $\|x - \tilde{x}\|_1$, and then becomes linear. Corollaries 4.4 and 4.5 still hold after this replacement.

Corollary 4.6. *The closure of the linear radial inequalities, that is, the set of points (p, x, z, t) of the linear relaxation of (4.3) that satisfy (4.4) with $\sum_i t_i$ in place of $\|x - \tilde{x}\|_1$ for every $\kappa \geq 0$, is a polyhedron. The $|\Gamma| + 1$ linear radial inequalities with $\kappa \in \Gamma \cup \{0\}$ describe this polyhedron.*

We now detail how we separate the radial inequalities for model (4.3). By Corollary 4.5, separating at a point $(\bar{p}, \bar{x}, \bar{z}, \bar{t})$ of the linear relaxation amounts to maximizing $\Phi(\kappa, \bar{p}) - \kappa \sum_i \bar{t}_i$ over $\kappa \geq 0$. The separation problem can be expressed as a single linear program (Appendix C). This linear program involves all scenarios at once, and is at least as large as the TSSP. Solving this linear program at every node of the branch-and-bound would be costly.

We instead separate the radial inequalities in two steps. Before the branch-and-bound, we compute the sets Γ^s and the values $\phi^s(\kappa)$ for every $s \in S$ and $\kappa \in \Gamma \cup \{0\}$, with the sandwich algorithm of Eisner and Severance (1976) (Appendix C).

During the branch-and-bound, at each solution $(\bar{p}, \bar{x}, \bar{z}, \bar{t})$ of a node relaxation, we evaluate the radial inequalities with $\kappa \in \Gamma \cup \{0\}$ from these values. If the inequality with $\kappa = 0$ is violated, we add it. Otherwise, we add the violated inequality with the largest

$$\eta(\kappa) := \frac{1}{\kappa}\Big(\Phi(\kappa, \bar{p}) - c^\top \bar{x} - \sum_{s \in S} \sum_{j \in J} \omega_j^s \bar{z}_j^s\Big), \qquad \kappa \in \Gamma. \tag{4.5}$$

At fixed $\bar{p}$ and fixed linearized primal objective, the radial inequality with slope κ forces $\sum_i t_i \geq \eta(\kappa)$, and is violated if and only if $\eta(\kappa) > \sum_i \bar{t}_i$. Since $\sum_i t_i$ appears in the objective, the largest $\eta(\kappa)$ raises the lower bound the most for the given solution. By Proposition 4.3, η is monotone between consecutive slopes of $\Gamma \cup \{0\}$ and beyond $\bar{\kappa}$. When the inequality with $\kappa = 0$ is satisfied, the largest $\eta(\kappa)$ over $\kappa > 0$ is thus attained in Γ.

This separation solves no linear program during the branch-and-bound, and the preprocessing is performed only once. For each scenario s, the sandwich algorithm solves a number of linear programs linear in $|\Gamma^s|$, each of the size of scenario s alone. We know no polynomial bound on $|\Gamma^s|$, but $|\Gamma^s|$ is bounded independently of $|S|$, so the preprocessing grows linearly with $|S|$.

4.3 A Decomposition Method

In (XPL-SL), the second-stage variables y^s of each scenario appear only through their recourse cost $(f^s)^\top y^s$ and the scenario-specific constraints (4.1f), as in a TSSP. This separability is exploited by the L-shaped method (Van Slyke and Wets 1969), which keeps the first-stage decision x in a master problem and handles the second-stage decisions through separate scenario subproblems. We extend this approach to (XPL-SL), accounting for two differences: the recourse costs enter the strong duality constraint (4.1g) rather than the objective, and they are weighted by the decision variables p^s.

A formulation with Benders cuts. In (XPL-SL), we replace each recourse cost $(f^s)^\top y^s$ by a variable θ^s, and remove y^s together with constraints (4.1f). For a first-stage decision x, θ^s should equal $Q^s(x)$ defined in (2.2). Enforcing $\theta^s \geq Q^s(x)$ is enough: θ^s only appears on the left-hand side of (4.1g), with a nonnegative coefficient p^s.

By linear programming duality, $Q^s(x)$ is the maximum of $(h^s - T^s x)^\top \pi$ over the dual feasible region of the recourse problem (2.2). This region does not depend on x. Assuming without loss of generality that W^s has full row rank, its set V^s of extreme points is nonempty and finite. The maximum is attained at one of the extreme points (Birge and Louveaux 2011, Sec. 5.1). We can therefore reformulate (XPL-SL) as

$$\begin{aligned}
\text{(XPL-SLB)} \quad \min_{p,x,\theta,\lambda,\mu} \quad & d(p,x) && \text{(4.6a)}\\
\text{s.t.} \quad & \text{(4.1b)–(4.1e), (4.1i)}, && \text{(4.6b)}\\
& x \geq 0, && \text{(4.6c)}\\
& c^\top x + \sum_{s\in S} p^s \theta^s \leq b^\top \lambda + \sum_{s\in S} (h^s)^\top \mu^s, && \text{(4.6d)}\\
& \theta^s \geq (h^s - T^s x)^\top \pi, \quad s \in S,\ \pi \in V^s. && \text{(4.6e)}
\end{aligned}$$

Inequalities (4.6e) are the Benders optimality cuts of the underlying TSSP. Under Assumption 2.1, no feasibility cuts are needed. The proof of the following result is given in Appendix A.

Proposition 4.7. *Under Assumption 2.1, problems (XPL-SLB) and (XPL-SL) have the same optimal value. If $(p, x, \theta, \lambda, \mu)$ is an optimal solution of (XPL-SLB) and, for each $s \in S$, y^s is an optimal solution of the recourse problem (2.2) at x, then (p, x, y, λ, μ) is an optimal solution of (XPL-SL).*

Constraint (4.6d) now carries the products $p^s\theta^s$. As in Proposition 4.1, the bounds $L^s \leq \theta^s \leq U^s$ can be added to (XPL-SLB) without changing its optimal value. The proof is analogous to that of Proposition 4.1, and is omitted. The products are thus linearized as in Section 4.1. After linearization, (XPL-SLB) is again a MILP for (XPL-Clu) and (XPL-Unit), and a MIBLP for (XPL-Sparse).

A branch-and-cut algorithm. Cuts (4.6e) are in general exponentially many. We therefore propose a branch-and-cut method to solve (XPL-SLB) that adds the cuts (4.6e) dynamically (Maher 2021). Initially, the solver is given (XPL-SLB) without any constraints (4.6e). Whenever an integer-feasible incumbent $(\bar{p}, \bar{x}, \bar{\theta}, \bar{\lambda}, \bar{\mu})$ is found, the solver invokes the separation procedure detailed in Algorithm 1.

Each simplex multiplier $\bar{\pi}^s$ is an extreme point of the dual feasible region of (2), that is, $\bar{\pi}^s \in V^s$. Every cut returned by Algorithm 1 belongs to the finite family (4.6e). If no cut is returned, then $\bar{\theta}^s \geq Q^s(\bar{x}) = \max_{\pi\in V^s}(h^s - T^s\bar{x})^\top\pi$ for all $s \in S$, so the incumbent is feasible for (XPL-SLB). The search then continues, as a better solution may still be found. Since each rejected incumbent violates a cut not yet in the model, only finitely many cuts are added. For (XPL-Clu) and (XPL-Unit), whose reformulations are MILPs, the branch-and-cut method thus terminates with an optimal solution of (XPL-SLB) (Maher

Algorithm 1 Separation procedure for the cuts (4.6e)

Require: an integer-feasible incumbent $(\bar{p}, \bar{x}, \bar{\theta}, \bar{\lambda}, \bar{\mu})$
$\mathcal{G} \leftarrow \emptyset$
for $s \in S$ **do**
    solve the recourse problem (2.2) at $\bar{x}$ with the simplex method
    let $\bar{\pi}^s$ be the simplex multipliers, so that $Q^s(\bar{x}) = (h^s - T^s\bar{x})^\top \bar{\pi}^s$
    if $\bar{\theta}^s < Q^s(\bar{x})$ **then**
        $\mathcal{G} \leftarrow \mathcal{G} \cup \{\, \theta^s \geq (h^s - T^s x)^\top \bar{\pi}^s \,\}$
    end if
end for
if $\mathcal{G} = \emptyset$ **then**
    accept the incumbent
else
    reject the incumbent and return the cuts in $\mathcal{G}$ to the solver
end if

2021). For (XPL-Sparse), the decomposition applies unchanged to the bilinear formulation, which the solver handles by spatial branch-and-bound.

Restricting the separation. The cuts only need to be separated for scenarios with positive probability. Indeed, θ^s appears in (XPL-SLB) only through the product $p^s\theta^s$ in (4.6d) and the cuts (4.6e) of scenario s. When $\bar{p}^s = 0$, the product vanishes. Setting $\bar{\theta}^s := Q^s(\bar{x})$ then satisfies these cuts, and changes neither the objective value nor (4.6d). We can therefore run the loop of Algorithm 1 over $\mathrm{supp}(\bar{p})$ instead of S, which solves at most k recourse problems instead of $|S|$. An incumbent accepted after this restricted separation can still be made feasible for (XPL-SLB), and the termination argument above remains valid.

The presented decomposition projects out the second-stage primal variables y^s, but not the dual multipliers μ^s, also indexed by scenario. These multipliers are coupled through constraint (4.1c), so they cannot be separated scenario by scenario. A Benders decomposition over them remains possible, but would have to handle this coupling. We leave this direction for future work.

5 Heuristic Solution Methods

Solving (XPL-Clu), (XPL-Unit) and (XPL-Sparse) to optimality may be computationally prohibitive for large TSSP instances, whereas a non-optimal explanation may suffice in practice. This section therefore presents three heuristic approaches. Their common principle is to avoid the difficulty of the exact models, where the first-stage decision is a variable of the lower level. The first one ignores the bilevel structure (Section 5.1). The other two reverse the viewpoint of (XPL). Instead of searching for a probability vector whose induced decision is as close as possible to $\tilde{x}$, they search for a probability vector under which $\tilde{x}$

is as close as possible to optimal. They measure this distance to optimality through the duality gap or through a perturbation of the first-stage costs (Section 5.2). In this section, $\mathcal{P}$ denotes one of the three sets $\mathcal{P}_{\text{sparse}}$, $\mathcal{P}_{\text{unit}}$, or $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ of Section 3.

5.1 Baseline: Classical Scenario Reduction (H-Distance)

The first heuristic serves as a baseline. It follows the principle of classical scenario reduction (Dupačová et al. 2003, Heitsch and Römisch 2003), and selects the admissible distribution $p \in \mathcal{P}$ closest to $\tilde{p}$ by solving the following optimization problem:

$$\text{(H-Distance)} \quad \min_{p \in \mathcal{P}} \quad d_p(p, \tilde{p}).$$

With the reformulations of Table 1, (H-Distance) is a MILP. The distance d_p is the Wasserstein distance of Section 3, whose cost matrix is built on the data of the scenarios, as is standard in classical scenario reduction. Other cost matrices could be used. Note, however, that (H-Distance) does not account for how $X^*(p)$ depends on p. The next two heuristics address this limitation by specifically targeting the optimality of $\tilde{x}$.

5.2 Fixing the First-Stage Decision (H-Gap, H-Cost)

The optimality conditions of the lower level are bilinear. In (XPL-SL), the strong duality constraint (4.1g) contains the products $p^s(f^s)^\top y^s$, since both p and y are variables. Retrospective scenario reduction seeks a probability vector under which the induced decision is $\tilde{x}$ itself. We therefore fix $x = \tilde{x}$ and $y = \tilde{y}$, and search for probabilities under which this given solution is optimal, in the spirit of inverse optimization (Ahuja and Orlin 2001). Since $\tilde{y}^s$ is an optimal recourse decision at $\tilde{x}$, the products reduce to $p^s Q^s(\tilde{x})$, where $Q^s(\tilde{x})$ is a constant, and the conditions become linear.

However, the resulting linear system may be infeasible, since there may be no $p \in \mathcal{P}$ making $\tilde{x}$ optimal. We therefore relax the optimality conditions with a gap $\varepsilon \in \mathbb{R}$ and a perturbation $\delta \in \mathbb{R}^{n_1}$ of the first-stage costs:

$$A^\top \lambda + \sum_{s \in S} (T^s)^\top \mu^s \leq c + \delta, \tag{5.1a}$$

$$(W^s)^\top \mu^s \leq p^s f^s, \quad s \in S, \tag{5.1b}$$

$$(c + \delta)^\top \tilde{x} + \sum_{s \in S} p^s Q^s(\tilde{x}) \leq b^\top \lambda + \sum_{s \in S} (h^s)^\top \mu^s + \varepsilon. \tag{5.1c}$$

With $\delta = 0$ and $\varepsilon = 0$, constraints (5.1) reduce to (4.1c)–(4.1d) and (4.1g) at $x = \tilde{x}$, and admit a solution if and only if $\tilde{x} \in X^*(p)$. With $\delta = 0$, constraints (5.1) state that $\tilde{x}$ is ε-optimal for (TSSP(p)). With $\varepsilon = 0$, constraints (5.1) state that $\tilde{x}$ is optimal for (TSSP(p)) with first-stage costs $c + \delta$. Minimizing one violation while setting the other to zero yields

two heuristics:

$$\text{(H-Gap)} \qquad \min_{p,\lambda,\mu,\varepsilon} \Big\{ \varepsilon : p \in \mathcal{P},\ (5.1) \text{ with } \delta = 0 \Big\},$$

$$\text{(H-Cost)} \qquad \min_{p,\lambda,\mu,\delta} \Big\{ \|\delta\|_1 : p \in \mathcal{P},\ (5.1) \text{ with } \varepsilon = 0 \Big\}.$$

The heuristic (H-Gap) seeks an admissible probability vector under which the suboptimality gap of $\tilde{x}$ is smallest. The heuristic (H-Cost) seeks an admissible probability vector under which $\tilde{x}$ is optimal for the smallest perturbation of the first-stage costs. With the descriptions of Table 1 and the standard linearization of the ℓ_1-norm, both problems are feasible MILPs that admit an optimal solution (Appendix A.9). The optimal value of each is zero if and only if some $p \in \mathcal{P}$ makes $\tilde{x}$ optimal. Both heuristics thus decide exactly whether $\tilde{x}$ can be recovered, and are heuristic only with respect to the objective function of (XPL).

All three heuristics return a probability vector $p^* \in \mathcal{P}$, but not the associated first-stage decision. The decision is recovered by solving (TSSP(p^*)), which always admits an optimal solution under Assumption 2.2. If several optimal solutions exist, we select the one minimizing $\|x - \tilde{x}\|_1$, by solving a second linear program over the set of optimal solutions of (TSSP(p^*)).

6 Computational Experiments

This section answers five questions, each stated in the title of Sections 6.2–6.6. We first describe the setup common to all experiments.

6.1 Experimental Setup

All experiments are run on the *chiclet* cluster of the Grid5000 testbed, whose nodes have two AMD EPYC 7301 CPUs (32 cores) and 128 GiB of RAM. All models are implemented in Julia (Bezanson et al. 2017) with JuMP (Lubin et al. 2023), and solved with Gurobi 13.0.2 (Gurobi Optimization, LLC 2026) under a time limit of 360 s. To mitigate the numerical instabilities caused by the linearizations of Section 4.1, we set `IntFeasTol` to 10^{-7}, `IntegralityFocus` to 1 and `NumericFocus` to 2 for the models of Section 4. For the same models, we set `PreCrush` to 1, as required by the cut callbacks of Sections 4.2 and 4.3. The heuristics of Section 5 use the default parameters of Gurobi. The code and the instances are available at `https://gitlab.inria.fr/abelfer/tssp-xpl`.

We consider three families of TSSPs of the form (2.1), whose formulations, sizes and scenario generation are detailed in Appendix D. The farmer's problem (FP) extends Example 2.3 to 100 crops, with uncertain yields in T^s. The supply chain planning problem (SCP) is based on Rathi et al. (2024, Sec. 4.1) and has uncertain demands in h^s, which are

independent and uniform (SCP0), independent and mixed Bernoulli–uniform (SCP1), or highly correlated (SCP2). The resource allocation problem (RAP) adapts a test problem of Lefebvre et al. (2026) to the stochastic setting, with uncertain demands in h^s. For each of the five groups and each number of scenarios in $\{15, 30, 50, 100\}$, we generate five TSSPs with uniform probabilities, which gives 100 TSSPs. An instance of retrospective scenario reduction is a pair formed by a TSSP and a target $k \in \{2, 5, 8\}$, which gives 300 instances. On our instances, ρ equals 100 for FP, 10 for SCP, and 10, 15 or 20 for RAP. Every target k is thus smaller than $\rho + 1$, and Proposition 2.4 does not guarantee that $\tilde{x}$ can be recovered.

For each TSSP, an initial solution $(\tilde{x}, \tilde{y})$ is computed once by solving (TSSP($\tilde{p}$)), so that all methods explain the same decision. An instance is solved when the final relative optimality gap is at most 10^{-4}. When the time limit is reached or the solver stops early, we report the best solution found, and we state explicitly when a comparison is restricted to the solved instances. On a few runs, the final solution of Gurobi violates its integrality or feasibility tolerance by at most $2 \cdot 10^{-5}$, or Gurobi discards a final solution for this reason; we then report the best solution returned by Gurobi, if any. Every method returns a feasible point (p, x) within the time limit on every instance, except one run of Section 6.4 and the runs of the decomposition reported in Section 6.5. For every returned point (p, x), we solve (TSSP(p)) a posteriori and check that x is optimal, up to a relative tolerance of 10^{-6} on the objective value. We consider that a method recovers $\tilde{x}$ when $\|x - \tilde{x}\|_1 \leq 10^{-6}$.

We use (H-Distance) run on $\mathcal{P}_{\text{sparse}}$, which implements classical scenario reduction, as the baseline of all comparisons. When only T^s and h^s are random, as in our TSSP instances, stability results bound the change in the optimal value of the TSSP by a multiple of the Wasserstein distance of order 1 between the distributions (Dupačová et al. 2003, Heitsch and Römisch 2003, Römisch 2003), which (H-Distance) minimizes.

6.2 Which Clustering Should (XPL-Clu) Be Built On?

Model (XPL-Clu) introduced in Section 3 takes as input a partition of the scenarios, as in the approach of Rathi et al. (2024). Before comparing the three models, we determine which partition we should use to best recover $\tilde{x}$.

We compare four partitions. Three of them cluster the scenarios according to their data ξ^s (Data), their optimal second-stage decisions $\tilde{y}^s$ in the initial TSSP (Recourse, as in Rathi et al. 2024), or a subgradient of Q^s at $\tilde{x}$ (Subgradient). This subgradient is $-(T^s)^\top \bar{\pi}^s$, where $\bar{\pi}^s$ is an optimal dual solution of the recourse problem (2.2) at $\tilde{x}$ (Shapiro et al. 2021, Sec. 2.1.1). The clusters are built with multi-start K-means (Lloyd 1982). For Data and Recourse, the vectors are first standardized as in Section 3. The fourth partition (Random) assigns each scenario to a random cluster, while ensuring that no cluster is empty.

All partitions are used in (XPL-Clu) with $\alpha = 1$, so that the cost function reduces to $d_x(x, \tilde{x})$. We report $\bar{d}_x := d_x / d_x^{\text{base}}$, where d_x^{base} is the decision distance of the baseline. For 6 instances, we have $d_x^{\text{base}} = 0$. On 5 of them, all partitions also recover $\tilde{x}$, and we set

Table 2: Comparison of partitions for (XPL-Clu). "Exact": percentage of instances on which $\tilde{x}$ is recovered. $\bar{d}_x$: median [first quartile, third quartile]. "Worst": percentage of instances on which the partition attains the largest $\bar{d}_x$ without recovering $\tilde{x}$; ties are counted for each tied partition.

k		Random	Recourse	Data	Subgradient
2	Exact	9	9	8	**13**
	$\bar{d}_x$	0.21 [0.06, 0.43]	0.34 [0.08, 0.58]	0.23 [0.08, 0.43]	**0.15** [0.04, 0.43]
	Worst	**17**	43	23	19
5	Exact	**28**	16	25	**28**
	$\bar{d}_x$	0.09 [0.01, 0.26]	0.21 [0.04, 0.52]	**0.06** [0.01, 0.31]	**0.06** [0.00, 0.19]
	Worst	**5**	60	10	14
8	Exact	27	22	27	**40**
	$\bar{d}_x$	0.09 [0.01, 0.25]	0.15 [0.04, 0.46]	0.08 [0.01, 0.27]	**0.03** [0.00, 0.14]
	Worst	**8**	55	14	**8**

$\bar{d}_x = 0$, as for any exact recovery. The sixth instance is discarded. Solution times do not differ noticeably across partitions and are not reported; they are compared across models in Section 6.3.

Table 2 shows that the Subgradient partition performs best: for every k, it attains the lowest median $\bar{d}_x$ and the highest percentage of exact recoveries, possibly tied. Unlike the other criteria, it accounts for the optimality of $\tilde{x}$: scenarios with similar subgradients play similar roles in its optimality condition, so a representative of a group can stand for the others.

Grouping scenarios by their data or by their recourse decisions, in contrast, does not beat the Random partition. The Recourse partition of Rathi et al. (2024) has the highest median $\bar{d}_x$ for every k, and never recovers $\tilde{x}$ more often than the Random one. The Data partition performs similarly to the Random one.

All partitions, however, do much better than the baseline. Their median $\bar{d}_x$ never exceeds 0.34: (XPL-Clu) reduces the decision distance of classical scenario reduction by at least 66% in median, whatever the partition. Most of the improvement over the baseline thus comes from the bilevel model itself.

Finally, no partition consistently outperforms the others. Each achieves the largest $\bar{d}_x$ on some instances, although the Random partition does so least often overall. Thus, any partition fixed in advance may be outperformed by another in recovering $\tilde{x}$. This motivates (XPL-Unit) and (XPL-Sparse), which do not require a predefined partition. We note that other clustering methods were also tested, including K-medoids (Kaufman and Rousseeuw 1987), without altering these conclusions.

6.3 Which Model Should Be Used?

By Proposition 3.3, the optimal value of (XPL-Sparse) is no larger than that of (XPL-Unit), which is no larger than that of (XPL-Clu). We measure how much the restrictions of $\mathcal{P}_{\text{unit}}$ and $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ increase the optimal value, and whether this ordering still holds under the time limit.

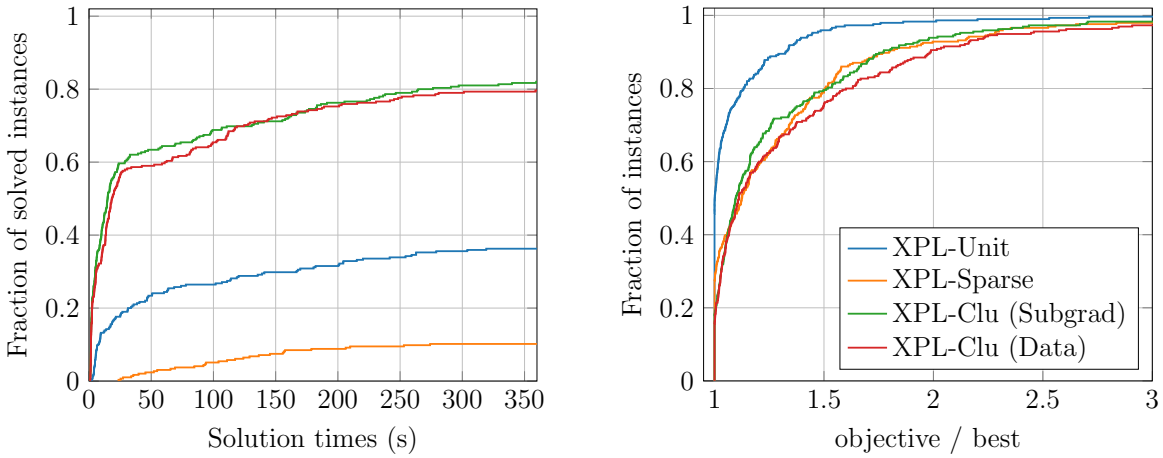
We now consider the cost function (3.2), with d_p as defined in Section 3. On each instance, d_x and d_p are normalized by their values for the baseline, which therefore has objective value 1. We set $\alpha = 0.75$: recovering $\tilde{x}$ is the primary goal of retrospective scenario reduction, and d_p mainly serves to discriminate between distributions that induce the same decision. The 6 instances on which the baseline already recovers $\tilde{x}$ are discarded, as the normalization of d_x is then undefined. Model (XPL-Clu) is built on two partitions of Section 6.2: Subgradient, which attains the lowest median $\bar{d}_x$, and Data, which groups scenarios with the same features as those used by d_p. This gives four models.

We first compare optimal values. Gurobi solves all models to optimality, within a gap of 0.01%, on only 25 instances, mostly with $k = 2$. On these instances, the optimal value of (XPL-Unit) exceeds that of (XPL-Sparse) by 4% in median, and by less than 7% on three quarters of them. Even with the better of its two partitions on each instance, the optimal value of (XPL-Clu) exceeds it by 11% in median, and by less than 45% on three quarters of them. Restricting the probabilities to integer multiples of $1/|S|$ thus increases the optimal value only slightly, whereas fixing a partition in advance increases it more. These 25 instances are, however, the easiest of the test bed, so these conclusions must be qualified.

We then consider all instances, under the time limit (Figure 2). Gurobi solves (XPL-Sparse), a nonconvex MIBLP, on far fewer instances than the other models (panel (a)). Panel (b) is a performance profile: for each factor $\tau \geq 1$, it gives the fraction of instances on which the objective value of a model is at most τ times the best value found by the four models. The curve of (XPL-Unit) lies above the others: (XPL-Unit) returns the best or a near-best solution on more instances than the other three models. Under the time limit, the returned solutions thus do not follow the ordering of Proposition 3.3. Overall, (XPL-Unit) returns better solutions within the time limit, at a small cost in optimal value, and keeps a direct interpretation: a selected scenario carrying u^s units represents u^s scenarios. We therefore recommend (XPL-Unit) for retrospective scenario reduction and keep it for the next experiments.

6.4 Do the Radial Inequalities Pay Off?

We now evaluate the radial inequalities of Section 4.2 on (XPL-Unit). We set $\alpha = 1$ to isolate the distance $d_x(x, \tilde{x})$, on which the inequalities act. We solve model (4.3) with Gurobi, with and without radial inequalities in a user-cut callback. The time limit of 360 s



(a) ECDF of the solution times.

(b) Performance profile of the objective.

Figure 2: Comparison of the four models over the whole test bed. In both panels, a curve that rises early indicates a model performing better. The legend of panel (b) applies to both panels.

Table 3: Radial inequalities on (XPL-Unit), with $\alpha = 1$, by status and solution time of the instances without the inequalities. Speedups and node reductions divide the solution time and the number of explored nodes without the inequalities by those with the inequalities.

	Solved without, in			Not solved	All
	< 10 s	10–100 s	≥ 100 s	without	
Number of instances					
Total	65	49	40	146	300
Solved with the inequalities	65	49	38	48	200
Slower with the inequalities	29	7	3	–	39
Unsolved, with a smaller final gap	–	–	–	12	12
Ratios on the instances solved in both cases					
Speedup, geometric mean	1.02	1.95	3.35	–	1.70
Speedup, maximum	3.7	15.9	48.7	–	48.7
Node reduction, geometric mean	2.28	3.89	7.32	–	3.62

includes the preprocessing. Many runs without the inequalities reach the time limit. Table 3 therefore groups the instances by their status without the inequalities: solved, by solution time, or not solved. The table first counts the instances solved with the inequalities, the slowdowns and, for the unsolved instances, the smaller final gaps, then gives the speedups and node reductions on the 152 instances solved in both cases.

The inequalities are cheap to compute and to separate. The preprocessing takes 1 s in median, and at most 9 s. At each call, the separation procedure examines the $|\Gamma| + 1$ slopes of $\Gamma \cup \{0\}$, that is, $2.9\,|S|$ slopes in median, and at most $29\,|S|$. The separation takes 4% of the solution time in median, and at most 26%.

On the instances solved without the inequalities, the speedup and the node reduction grow with the solution time without the inequalities. Slowdowns occur mostly on the instances

solved without the inequalities in less than 10 s. On these instances, the inequalities still reduce the number of explored nodes, but the preprocessing and the separation take a significant share of a short run. Two instances, solved without the inequalities in more than 100 s, are no longer solved with them.

With the inequalities, Gurobi solves a third of the instances not solved without the inequalities. On the 98 instances that remain unsolved, the final gap decreases on 12 of them, and never increases. The objective value of the incumbent improves on 67 of these 98 instances, and worsens on 23. The inequalities cut off points of the linear relaxation where the linearized primal objective underestimates the primal objective. The primal heuristics of Gurobi, which start from the solutions of the relaxation, thus start from better points, which likely explains most improvements. The incumbent may still worsen, since the inequalities also change the search tree.

Overall, the radial inequalities pay off, and we use them in Section 6.6.

6.5 Does the Decomposition Pay Off?

We now apply the decomposition of Section 4.3 to (XPL-Unit), with the cost function of Section 6.3, and discard the same six instances. We compare the branch-and-cut method with (XPL-SL) solved by Gurobi. We test two options of the branch-and-cut method. *Active* restricts the separation to the scenarios with positive probability (Section 4.3). *Init* departs from the initialization of Section 4.3: instead of starting without any Benders cut, the solver is given from the start all the cuts generated by the L-shaped method when solving the initial TSSP to optimality. Enabling each option or not gives four configurations.

We use the measures of Section 6.4. On the instances solved by both methods, we compare the solution times through speedups, aggregated by their geometric mean. On the instances solved by neither, we compare the final incumbents. On 7 to 17 runs, depending on the configuration, the decomposition returns no solution, as Gurobi rejects its incumbents for constraint violations. Most of these runs concern FP and RAP. We count them as unsolved, without incumbent. Table 4 summarizes the results.

Table 4 shows that whether the decomposition pays off depends mainly on the instance group. On the three SCP groups, every configuration is faster than (XPL-SL). On the instances solved by neither, every configuration also finds a better incumbent more often than a worse one. On FP and RAP, in contrast, every configuration is slower than (XPL-SL), or as fast at best. Overall, every configuration solves 111 to 113 instances, against 104 for (XPL-SL).

Active yields the largest speedups on four groups, but also the largest number of runs without solution (17). Init + Active keeps speedups between 1.70 and 2.31 on the SCP groups, finds the best incumbents on SCP0 and SCP2, and returns no solution on 9 runs only.

The L-shaped iterations reported in Table 4 explain the difference between the groups.

Table 4: Decomposition applied to (XPL-Unit), against (XPL-SL), by group of instances. L-shaped iterations: mean number of iterations of the L-shaped method on the initial TSSP. Incumbents are compared on the instances solved by neither; a run without solution counts as a worse incumbent. Bold: largest speedup of each column.

	FP	RAP	SCP0	SCP1	SCP2	All
Instances	60	60	59	57	58	294
L-shaped iterations, mean	44.1	80.7	10.5	9.8	9.7	31.4
Solved instances						
(XPL-SL)	8	26	20	22	28	104
Branch-and-cut	10	25	23	23	30	111
+ Init	10	25	25	23	30	113
+ Active	10	24	25	22	30	111
+ Init + Active	10	27	23	22	29	111
Speedup, geometric mean						
Branch-and-cut	0.29	0.61	1.67	1.24	1.15	0.97
+ Init	0.26	0.46	2.04	1.74	1.43	1.07
+ Active	**0.77**	**1.00**	**2.37**	1.65	**1.73**	**1.50**
+ Init + Active	0.48	0.57	2.31	**1.91**	1.70	1.28
Better / worse incumbent, on the instances solved by neither						
Branch-and-cut	2 / 46	5 / 21	19 / 14	19 / 12	20 / 8	65 / 101
+ Init	0 / 48	7 / 19	26 / 6	22 / 8	20 / 7	75 / 88
+ Active	3 / 45	6 / 22	22 / 9	17 / 16	19 / 8	67 / 100
+ Init + Active	1 / 48	8 / 16	28 / 5	21 / 11	24 / 4	82 / 84

The L-shaped method needs far fewer iterations on the SCP groups than on FP and RAP. On FP and RAP, the recourse functions are thus poorly approximated by a small number of Benders cuts. The decomposition relies on the same cuts, and inherits this slow convergence.

Overall, the decomposition is not uniformly faster, but its benefit can be anticipated before solving any bilevel model: it pays off when the L-shaped method solves the underlying TSSP in few iterations. In that case, we recommend the Init + Active configuration.

6.6 Can the Heuristics Replace the Exact Models?

We run the heuristics of Section 5 on $\mathcal{P}_{\text{unit}}$, the admissible set recommended in Section 6.3. Since (H-Gap) and (H-Cost) do not account for d_p, we compare all methods on d_x only. The study has two steps. We first compare the heuristics with one another. We then solve (XPL-Unit) with $\alpha = 1$ and the radial inequalities of Section 4.2, warm-started with the probability vector of the best heuristic, to measure how far each heuristic is from the optimum. We consider that this model improves a heuristic when it reduces d_x by more than 1%. Table 5 summarizes the results.

Selecting a distribution close to $\tilde{p}$ does not explain $\tilde{x}$: the baseline recovers $\tilde{x}$ on only 2% of the instances.

(H-Gap) and (H-Cost) recover $\tilde{x}$ on 54% of the instances, in about 3 s in median. On

Table 5: Heuristics and warm-started (XPL-Unit) on $\mathcal{P}_{\text{unit}}$, by group of instances. The rows below "Not recovered by any heuristic" are restricted to these instances. $\bar{d}_x$ is the distance d_x divided by that of (H-Distance) on the same instance. (XPL-Unit) improves a heuristic when it reduces d_x by more than 1%.

	FP	RAP	SCP0	SCP1	SCP2	All
All instances	60	60	60	60	60	300
$\tilde{x}$ recovered (number)						
(H-Distance)	0	0	1	3	2	6
(H-Cost)	0	20	43	49	49	161
(H-Gap)	0	20	43	49	49	161
Recovery proved impossible (number)	43	36	16	11	11	117
Undecided (number)	17	4	1	0	0	22
Median time (s)						
(H-Distance)	0.8	1.0	0.8	0.7	0.7	0.8
(H-Cost)	41.0	2.8	2.6	2.9	2.6	3.3
(H-Gap)	98.3	2.7	2.3	2.4	2.5	2.9
Not recovered by any heuristic	60	40	17	11	11	139
Improved by (XPL-Unit) (number)						
over (H-Cost)	60	38	16	10	11	135
over (H-Gap)	54	31	13	8	5	111
Median $\bar{d}_x$						
(H-Cost)	0.74	0.32	0.52	0.35	0.31	0.39
(H-Gap)	0.07	0.18	0.25	0.15	0.07	0.12
(XPL-Unit), warm-started	0.03	0.12	0.12	0.04	0.02	0.06

FP, they recover no instance, and (H-Cost) and (H-Gap) take 41 s and 98 s in median. On 39% of the instances, they prove that $\tilde{x}$ cannot be recovered with k scenarios. Only 7% of the instances remain undecided within the time limit, most of them in FP, and the warm-started model recovers $\tilde{x}$ on none of them.

On the 139 instances that no heuristic recovers, the warm-started model improves (H-Gap) on 80% of them and (H-Cost) on 97% of them. The median $\bar{d}_x$ drops from 0.12 for (H-Gap) and 0.39 for (H-Cost) to 0.06.

Overall, the heuristics suffice when $\tilde{x}$ can be recovered: they recover it in a few seconds, and on most other instances they prove that it cannot be. When $\tilde{x}$ cannot be recovered, the bilevel model substantially improves their explanations. We therefore recommend a two-stage approach: (H-Gap) first decides whether $\tilde{x}$ can be explained with k scenarios and, if not, its solution warm-starts (XPL-Unit).

7 Discussion and Future Work

In this paper, we propose a general framework to explain the first-stage decisions of TSSPs through retrospective scenario reduction. Computing such explanations amounts to solving a challenging bilevel problem whose lower level is a TSSP. For this problem, we propose

exact solution methods that exploit the structure of the underlying TSSP, as well as heuristic methods based on the optimality conditions of the lower level. Our numerical results identify the most effective approaches, and show that they provide good-quality explanations for instances with up to one hundred scenarios.

Several directions remain open. First, the bilevel framework of (XPL) is not restricted to scenario reduction. Other choices of the objective function d and of the admissible set $\mathcal{P}$ lead to other explanations. In such applications, it may be preferable to constrain the induced decision to lie in a set $\mathcal{D}$, as in counterfactual explanations (Korikov and Beck 2021, Kurtz et al. 2026), rather than to penalize its distance to a target. When $\mathcal{D}$ is described by linear constraints, the single-level reformulation, the valid inequalities and the decomposition we propose still apply. The feasibility of (XPL) is, however, no longer guaranteed, and the impact of this coupling constraint on the solution methods remains to be studied.

Second, under the optimistic perspective, a reduction is considered successful as soon as $\tilde{x}$ is one of the optimal decisions of the reduced TSSP. Thus, our model can output a reduction that does not uniquely identify $\tilde{x}$. An extreme case arises when the reduced program has X itself as its optimal set: every decision is optimal, and the resulting explanation provides no useful information. A pessimistic perspective, in which the reduction is evaluated with respect to the worst decision in the optimal set, would rule out such cases. The resulting model is more challenging, and we leave its analysis to future work.

Acknowledgments

This research is supported by the project "Seamless Shared Urban Mobility (SUM)". This project has received funding from the European Union's Horizon Europe Research and Innovation Programme under grant agreement No. 101103646.

Experiments presented in this paper were carried out using Grid5000, one of the precursor infrastructures of the SLICES-FR testbed, operated by a scientific interest group including 14 major French academic institutes and Universities (see https://slices-fr.eu/).

A Proofs

A.1 Preliminaries on the Recourse Functions

Several proofs of this appendix rely on the following properties of the recourse functions. Let $s \in S$. By linear programming duality, for every $x \in X$,

$$Q^s(x) = \max\{(h^s - T^s x)^\top \pi : \pi \in \Pi^s\}, \qquad \Pi^s := \{\pi \in \mathbb{R}^{m_2} : (W^s)^\top \pi \leq f^s\}. \tag{A.1}$$

Under Assumption 2.1, the recourse problem (2.2) is feasible and bounded below for every $x \in X$. Hence Π^s is nonempty, the maximum in (A.1) is attained, and Q^s is finite on X. Moreover, Q^s is convex and polyhedral (Shapiro et al. 2021, Sec. 2.1.1): its restriction to X is the maximum of finitely many affine functions. In particular, Q^s is continuous on X.

For $x \in X$, let $\Pi^s(x)$ denote the set of optimal solutions of the maximization in (A.1). The subdifferential of Q^s at x is (Shapiro et al. 2021, Sec. 2.1.1)

$$\partial Q^s(x) = \big\{-(T^s)^\top \pi : \pi \in \Pi^s(x)\big\}, \tag{A.2}$$

and the normal cone to X at x reads

$$N_X(x) = \big\{A^\top \nu - \sigma : \nu \in \mathbb{R}^{m_1},\ \sigma \in \mathbb{R}^{n_1}_+,\ \sigma_i = 0 \text{ for all } i \text{ such that } x_i > 0\big\}. \tag{A.3}$$

Finally, let $p \in \mathcal{P}_0$. A decision $x \in X$ belongs to $X^*(p)$ if and only if (Shapiro et al. 2021, Sec. 2.1)

$$0 \in c + \sum_{s \in S} p^s\, \partial Q^s(x) + N_X(x). \tag{A.4}$$

A.2 Proof of Proposition 2.4

Proof. Since $\tilde{x} \in X^*(\tilde{p})$, Equations (A.4) and (A.2) show that there exist $\pi^s \in \Pi^s(\tilde{x})$, $s \in S$, and $\tilde{v} \in N_X(\tilde{x})$ such that

$$c + \sum_{s \in S} \tilde{p}^s g^s + \tilde{v} = 0, \qquad \text{where } g^s := -(T^s)^\top \pi^s \in \partial Q^s(\tilde{x}). \tag{A.5}$$

The proof looks for a vector p with reduced support satisfying the same condition, with the subgradients g^s unchanged. The vector $\tilde{v}$, in contrast, is adjusted: the expression (A.3) of $N_X(\tilde{x})$ shows that any component in $(\ker A)^\perp = \mathrm{Im}(A^\top)$ can be added to $\tilde{v}$ without leaving the cone. We therefore look for a vector p such that $\sum_{s\in S} p^s g^s$ equals $\sum_{s\in S} \tilde{p}^s g^s$ after projection onto $\ker A$, and Carathéodory's theorem yields one. What is left out by the projection lies in $(\ker A)^\perp$, and is added to $\tilde{v}$ to recover the full condition.

Let Z be a matrix whose columns form a basis of $\ker A$. We look for p such that

$$\sum_{s \in S} p^s Z^\top g^s = \sum_{s \in S} \tilde{p}^s Z^\top g^s.$$

The points $Z^\top g^s = -(T^s Z)^\top \pi^s$ lie in the image of $(T^s Z)^\top$, hence in the image of $(MZ)^\top$. The dimension of this subspace is $\mathrm{rank}(MZ) = \rho$, because the columns of MZ span $M \ker A$. As $\tilde{p} \in \mathcal{P}_0$, $\sum_{s\in S} \tilde{p}^s Z^\top g^s$ is a convex combination of the $|S|$ points $Z^\top g^s$, all lying in that subspace of dimension ρ. Carathéodory's theorem therefore provides a probability vector $p \in \mathcal{P}_0$, supported on a subset $I \subseteq S$ with $|I| \le \rho + 1$, such that

$$\sum_{s\in S} p^s Z^\top g^s = \sum_{s\in S} \tilde{p}^s Z^\top g^s. \tag{A.6}$$

Set $r := \sum_{s\in S}(\tilde{p}^s - p^s)g^s$. Equality (A.6) gives $Z^\top r = 0$. As the columns of Z span $\ker A$, the vector r belongs to $(\ker A)^\perp$.

Since $\sum_{s\in S} \tilde{p}^s g^s = \sum_{s\in S} p^s g^s + r$, condition (A.5) reads

$$c + \sum_{s\in S} p^s g^s + \tilde{v} + r = 0.$$

Define $v := \tilde{v} + r$. Because r belongs to $(\ker A)^\perp$ and $\tilde{v}$ to $N_X(\tilde{x})$, the vector v is in $N_X(\tilde{x})$. Condition (A.4) therefore gives $\tilde{x} \in X^*(p)$. □

A.3 Proof of Proposition 3.1

Proof. Let $V(p,x) := c^\top x + \sum_{s\in S} p^s Q^s(x)$ defined on $\mathcal{P}_0 \times X$. This function is continuous on $\mathcal{P}_0 \times X$ since each Q^s is finite-valued and continuous on X (Appendix A.1). Because the lower-level feasible set X does not depend on p, the feasible set of (3.1) can be written as

$$\mathcal{F} = \bigcap_{z\in X} \big\{(p,x) \in \mathcal{P} \times X : V(p,x) - V(p,z) \le 0\big\}.$$

This set is closed because it is an intersection of closed sets. It is nonempty and bounded by Assumption 2.2, since $\mathcal{P} \subseteq [0,1]^{|S|}$ and $X^*(p) \subseteq B$ for every $p \in \mathcal{P}$. Since d is lower semicontinuous, it attains its minimum on the nonempty compact set $\mathcal{F}$. □

A.4 Proof of Proposition 3.3

Proof. The result follows from the definitions of the three sets. Every $p \in \mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ has exactly k positive components, one per cluster, each equal to the total probability of its cluster, which is an integer multiple of $1/|S|$ under Assumption 3.2; hence $p \in \mathcal{P}_{\text{unit}}$. Every $p \in \mathcal{P}_{\text{unit}}$ has at most k nonzero components, hence belongs to $\mathcal{P}_{\text{sparse}}$.

For strictness, note that every vector of $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ has exactly one positive value per cluster. If $2 \le k < |S|$, there is a cluster containing two distinct scenarios $s_1 \ne s_2$. The vector assigning $1/|S|$ to s_1, $(|S|-1)/|S|$ to s_2, and 0 elsewhere belongs to $\mathcal{P}_{\text{unit}}$ but not to $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$. The vector assigning $1/(2|S|)$ to s_1, $1 - 1/(2|S|)$ to s_2, and 0 elsewhere belongs to $\mathcal{P}_{\text{sparse}}$ but not to $\mathcal{P}_{\text{unit}}$, since $1/(2|S|)$ is not an integer multiple of $1/|S|$. □

A.5 Proof of Proposition 4.1

Proof. We first prove that $\underline{Q}^s$ and $\overline{Q}^s$ are finite by a compactness argument. By Assumption 2.2, $X_{\mathcal{P}}$ is bounded. Its closure is therefore a compact subset of the closed set X.

By Appendix A.1, Q^s is finite and continuous on X. It is hence bounded on this closure. Consequently $\underline{Q}^s$ and $\overline{Q}^s$ are finite for every $s \in S$.

We now prove that constraints (4.2) do not cut off every optimal solution. To do so, we take an optimal solution of (XPL-SL) and replace its second-stage decisions by optimal recourse decisions. We show that the resulting point is feasible, then optimal, and finally that it satisfies (4.2). Let (p, x, y, λ, μ) be an optimal solution of (XPL-SL). For each $s \in S$, let $\hat{y}^s$ be an optimal solution of the recourse problem (2.2) at x, which exists under Assumption 2.1.

Constraints (4.1b)–(4.1e) and (4.1i) do not involve y and remain satisfied. Constraints (4.1f) and (4.1h) hold since $\hat{y}^s$ is feasible for the recourse problem (2.2) of scenario s. For the strong duality constraint (4.1g), optimality of $\hat{y}^s$ for the recourse problem of scenario s gives $(f^s)^\top \hat{y}^s \leq (f^s)^\top y^s$. Multiplying by $p^s \geq 0$ and summing over $s \in S$,

$$c^\top x + \sum_{s \in S} p^s (f^s)^\top \hat{y}^s \;\leq\; c^\top x + \sum_{s \in S} p^s (f^s)^\top y^s \;\leq\; b^\top \lambda + \sum_{s \in S} (h^s)^\top \mu^s,$$

where the second inequality is (4.1g) at the original point. Hence $(p, x, \hat{y}, \lambda, \mu)$ satisfies (4.1g), and is feasible for (XPL-SL). Since the objective function $d(p, x)$ does not depend on y, $(p, x, \hat{y}, \lambda, \mu)$ has the same value as (p, x, y, λ, μ), and is therefore optimal.

It remains to check the bounds. By (4.1g) and weak duality, $x \in X^*(p)$ with $p \in \mathcal{P}$, hence $x \in X_{\mathcal{P}}$. Since $(f^s)^\top \hat{y}^s = Q^s(x)$ by construction,

$$\underline{Q}^s \leq Q^s(x) = (f^s)^\top \hat{y}^s \leq \overline{Q}^s, \qquad s \in S. \quad \square$$

A.6 Proof of Proposition 4.2

Proof. Let $\kappa \geq 0$ and $s \in S$. Taking $x = \tilde{x}$ in the definition of ϕ^s gives $\phi^s(\kappa) \leq c^\top \tilde{x} + Q^s(\tilde{x})$, which is finite by Assumption 2.1. The penalty $\kappa \|x - \tilde{x}\|_1$ is nonnegative, so $\phi^s(\kappa)$ is at least the optimal value of the TSSP restricted to scenario s, which is finite by Assumption 2.2. Hence $\phi^s(\kappa)$ and $\Phi(\kappa, p)$ are finite.

By definition of ϕ^s, every $x \in X$ satisfies $c^\top x + Q^s(x) \;\geq\; \phi^s(\kappa) - \kappa \|x - \tilde{x}\|_1$ for every $s \in S$. Let p be a probability vector. Since $\sum_{s \in S} p^s = 1$, multiplying these inequalities by $p^s \geq 0$ and summing over $s \in S$ gives

$$c^\top x + \sum_{s \in S} p^s Q^s(x) \;\geq\; \Phi(\kappa, p) - \kappa \|x - \tilde{x}\|_1. \tag{A.7}$$

Consider now a feasible solution of (4.3). Since β is binary, constraints (4.3e)–(4.3f) give $z^s_j = \beta^s_j (f^s)^\top y^s$. By (4.3c), the linearized primal objective then equals the primal objective of (TSSP(p)). Each y^s is a feasible solution of the recourse problem at x, so

$(f^s)^\top y^s \geq Q^s(x)$, and

$$c^\top x + \sum_{s\in S}\sum_{j\in J} \omega_j^s z_j^s \;=\; c^\top x + \sum_{s\in S} p^s (f^s)^\top y^s \;\geq\; c^\top x + \sum_{s\in S} p^s Q^s(x). \tag{A.8}$$

Combining (A.7) and (A.8) gives (4.4). □

A.7 Proof of Proposition 4.3

Proof. We first establish similar properties for each function ϕ^s, $s \in S$, then combine them.

Let $s \in S$. With the recourse problem (2.2) in place of Q^s and auxiliary variables t for $\|x - \tilde{x}\|_1$, $\phi^s(\kappa)$ reads

$$\phi^s(\kappa) = \min\left\{c^\top x + (f^s)^\top y + \kappa \mathbf{1}^\top t \;:\; (x, y, t) \in \Omega^s\right\}, \tag{A.9}$$

where $\Omega^s := \{(x, y, t) : Ax = b,\ T^s x + W^s y = h^s,\ t \geq x - \tilde{x},\ t \geq \tilde{x} - x,\ x \geq 0,\ y \geq 0\}$ does not depend on κ. Since $t \geq 0$ on Ω^s, the objective of (A.9) is nondecreasing in κ at every point of Ω^s, so ϕ^s is nondecreasing. Since $\phi^s(\kappa)$ is finite (Appendix A.6), the minimum in (A.9) is attained at an extreme point of Ω^s. The function ϕ^s is thus the minimum of finitely many affine functions of κ, one per extreme point of Ω^s. Hence ϕ^s is concave and piecewise linear on $\mathbb{R}_+$, with a finite set Γ^s of breakpoints in $(0, +\infty)$.

Let $\bar{\kappa}^s := \max(\Gamma^s \cup \{0\})$. We now show that $\phi^s(\kappa) = c^\top \tilde{x} + Q^s(\tilde{x})$ for every $\kappa \geq \bar{\kappa}^s$. On $[\bar{\kappa}^s, +\infty)$, ϕ^s is affine, nondecreasing and bounded from above by $c^\top \tilde{x} + Q^s(\tilde{x})$, hence constant. Let $\kappa > \bar{\kappa}^s$, and let (x, y, t) be an optimal solution of (A.9) for κ. This solution is also feasible for (A.9) with $\bar{\kappa}^s$, with value $\phi^s(\kappa) - (\kappa - \bar{\kappa}^s)\mathbf{1}^\top t \geq \phi^s(\bar{\kappa}^s)$. Since $\phi^s(\kappa) = \phi^s(\bar{\kappa}^s)$, we obtain $t = 0$, which forces $x = \tilde{x}$. Hence $\phi^s(\kappa) = c^\top \tilde{x} + Q^s(\tilde{x})$. The function ϕ^s being constant on $[\bar{\kappa}^s, +\infty)$, we obtain

$$\phi^s(\kappa) = c^\top \tilde{x} + Q^s(\tilde{x}), \qquad \kappa \geq \bar{\kappa}^s. \tag{A.10}$$

Let $\Gamma := \bigcup_{s\in S} \Gamma^s$, a finite subset of $(0, +\infty)$, and let $p \in \mathcal{P}_0$. The function $\Phi(\cdot, p) = \sum_{s\in S} p^s \phi^s$ is a convex combination of concave, nondecreasing and piecewise linear functions, with breakpoints in Γ, which proves (i). By definition of Γ, $\bar{\kappa} = \max(\Gamma \cup \{0\}) = \max_{s\in S} \bar{\kappa}^s$. For $\kappa \geq \bar{\kappa}$, equalities (A.10) hold for every $s \in S$, and $\sum_{s\in S} p^s = 1$ gives (ii). □

A.8 Proof of Proposition 4.7

Proof. Let $x \in X$. By (A.1), $Q^s(x)$ is the maximum of $(h^s - T^s x)^\top \pi$ over Π^s. Since W^s has full row rank, Π^s contains no line, so its set V^s of extreme points is nonempty and finite. Under Assumption 2.1, this maximum is finite, so it is attained at one of these extreme points. Constraints (4.6e) are therefore equivalent to $\theta^s \geq Q^s(x)$, $s \in S$.

We show that every feasible point of either problem yields a feasible point of the other with the same objective value. The objective $d(p, x)$ and the constraints (4.1b)–(4.1e), (4.1i) and $x \geq 0$ are common to both problems and involve neither y nor θ. Only the remaining constraints need to be checked.

Let $(p, x, \theta, \lambda, \mu)$ be feasible for (XPL-SLB). For each $s \in S$, let y^s be an optimal solution of the recourse problem (2.2) at x, which exists under Assumption 2.1. Since y^s is feasible for this problem, (x, y) satisfies (4.1f) and (4.1h). Moreover, $(f^s)^\top y^s = Q^s(x) \leq \theta^s$ for every $s \in S$. Since $p^s \geq 0$, we obtain $\sum_{s\in S} p^s (f^s)^\top y^s \leq \sum_{s\in S} p^s \theta^s$, so (4.1g) follows from (4.6d).

Conversely, let (p, x, y, λ, μ) be feasible for (XPL-SL), and set $\theta^s := (f^s)^\top y^s$ for every $s \in S$. Then (4.6d) coincides with (4.1g). Moreover, y^s is feasible for the recourse problem at x by (4.1f) and (4.1h), so $\theta^s = (f^s)^\top y^s \geq Q^s(x)$, and (4.6e) holds.

In both directions, p and x are unchanged, so the objective value is preserved, and the two problems have the same optimal value. In particular, completing an optimal solution of (XPL-SLB) with optimal recourse decisions y^s, as in the first direction, yields an optimal solution of (XPL-SL). □

A.9 Feasibility of (H-Gap) and (H-Cost)

Let $p \in \mathcal{P}$. We show that (5.1) admits a solution with $\delta = 0$, and a solution with $\varepsilon = 0$, so that both problems are feasible. Both objective values are bounded below by zero, by weak duality for (H-Gap). For each of the finitely many values of the integer variables describing $\mathcal{P}$, the remaining problem is a linear program bounded below, so both problems admit an optimal solution.

With $\delta = 0$, (TSSP(p)) admits an optimal solution under Assumption 2.2. By strong duality, its dual admits an optimal solution (λ, μ), which satisfies (5.1) with $\delta = 0$ and ε equal to the suboptimality gap of $\tilde{x}$.

With $\varepsilon = 0$, let π^s be an optimal dual solution of the recourse problem (2.2) at $\tilde{x}$, which exists under Assumption 2.1. Set $\lambda = 0$, $\mu^s = p^s \pi^s$ for $s \in S$, and $\delta = \sum_{s\in S} (T^s)^\top \mu^s - c$. Constraint (5.1a) holds with equality, and (5.1b) follows from $(W^s)^\top \pi^s \leq f^s$ and $p^s \geq 0$. Note that $(h^s - T^s\tilde{x})^\top \pi^s = Q^s(\tilde{x})$ by strong duality for (2.2). Hence $(c + \delta)^\top \tilde{x} + \sum_{s\in S} p^s Q^s(\tilde{x}) = \sum_{s\in S} p^s (\pi^s)^\top T^s \tilde{x} + \sum_{s\in S} p^s (h^s - T^s\tilde{x})^\top \pi^s = \sum_{s\in S} (h^s)^\top \mu^s$, which is (5.1c) with $\varepsilon = 0$. □

B Mixed-Integer Linear Descriptions

Recall that (XPL) reads $\min_{p,x}\{d(p, x) : p \in \mathcal{P},\ x \in X^*(p)\}$, with the objective function d defined in Equation (3.2). This appendix shows that, for the three admissible sets of Section 3, the upper level of (XPL) can be written with the auxiliary variables of Table 1 and linear constraints. Section B.1 replaces the objective function, Section B.2 the admissible sets, and Section B.3 presents the binary expansion used to linearize (XPL-Unit).

B.1 Distances

In (3.2), both $d_x(x, \tilde{x})$ and $d_p(p, \tilde{p})$ are optimal values of linear minimization problems. The decision distance satisfies

$$d_x(x, \tilde{x}) = \|x - \tilde{x}\|_1 = \min_{t \in \mathbb{R}^{n_1}} \Big\{ \sum_{i=1}^{n_1} t_i : t \geq x - \tilde{x},\ t \geq \tilde{x} - x \Big\}.$$

The Wasserstein distance of order 1 between p and $\tilde{p}$, with cost matrix $(c_{ss'})_{s,s' \in S}$, is the optimal value of the transportation problem

$$d_p(p, \tilde{p}) = \min_{\gamma \geq 0} \Big\{ \sum_{s,s' \in S} c_{ss'} \gamma_{ss'} : \sum_{s' \in S} \gamma_{ss'} = p^s,\ s \in S,\ \sum_{s \in S} \gamma_{ss'} = \tilde{p}^{s'},\ s' \in S \Big\}.$$

Since both terms enter (3.2) with nonnegative weights and (XPL) is a minimization problem, the variables t and γ and the constraints above can be added to the upper level, and $d(p, x)$ replaced by $\alpha \sum_i t_i + (1 - \alpha) \sum_{s,s'} c_{ss'} \gamma_{ss'}$. This requires n_1 variables and $2n_1$ constraints for d_x, and $|S|^2$ variables and $2|S|$ constraints for d_p.

B.2 Admissible Sets

Each set can be described with binary variables $v \in \{0, 1\}^{|S|}$, where $v^s = 1$ allows scenario s to be selected.

For $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$, let $\tilde{p}^s_{\mathcal{C}} := \sum_{s' \in C(s)} \tilde{p}^{s'}$ denote the total probability of the cluster $C(s) \in \mathcal{C}$ containing s. Then $p \in \mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ if and only if there exists $v \in \{0, 1\}^{|S|}$ such that

$$\sum_{s \in C} v^s = 1, \quad C \in \mathcal{C}, \qquad p^s = \tilde{p}^s_{\mathcal{C}} v^s, \quad s \in S.$$

For $\mathcal{P}_{\text{unit}}$, integer variables $u \in \{0, \ldots, |S|\}^{|S|}$ count the scenarios that each selected scenario represents. Then $p \in \mathcal{P}_{\text{unit}}$ if and only if there exist $v \in \{0, 1\}^{|S|}$ and $u \in \{0, \ldots, |S|\}^{|S|}$ such that

$$\sum_{s \in S} v^s \leq k, \qquad u^s \leq |S|\, v^s, \quad s \in S, \qquad \sum_{s \in S} u^s = |S|, \qquad p^s = u^s / |S|, \quad s \in S.$$

For $\mathcal{P}_{\text{sparse}}$, $p \in \mathcal{P}_{\text{sparse}}$ if and only if there exists $v \in \{0, 1\}^{|S|}$ such that

$$\sum_{s \in S} v^s \leq k, \qquad 0 \leq p^s \leq v^s, \quad s \in S, \qquad \sum_{s \in S} p^s = 1.$$

The three sets are closed: $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ and $\mathcal{P}_{\text{unit}}$ are finite, and $\mathcal{P}_{\text{sparse}}$ is the finite union, over the subsets $K \subseteq S$ with $|K| \leq k$, of the faces $\{p \in \mathcal{P}_0 : p^s = 0,\ s \notin K\}$ of $\mathcal{P}_0$.

As an example, the complete formulation of (XPL-Unit) reads

$$\text{(XPL-Unit)} \quad \min_{p,u,v,x,t,\gamma} \quad \alpha \sum_{i=1}^{n_1} t_i + (1-\alpha) \sum_{s,s' \in S} c_{ss'} \gamma_{ss'} \tag{B.1a}$$

$$\text{s.t.} \quad t \geq x - \tilde{x}, \quad t \geq \tilde{x} - x, \tag{B.1b}$$

$$\sum_{s' \in S} \gamma_{ss'} = p^s, \quad s \in S, \qquad \sum_{s \in S} \gamma_{ss'} = \tilde{p}^{s'}, \quad s' \in S, \qquad \gamma \geq 0, \tag{B.1c}$$

$$\sum_{s \in S} v^s \leq k, \quad u^s \leq |S|\, v^s, \quad s \in S, \quad \sum_{s \in S} u^s = |S|, \tag{B.1d}$$

$$p^s = u^s / |S|, \quad s \in S, \tag{B.1e}$$

$$v \in \{0,1\}^{|S|}, \quad u \in \{0, \ldots, |S|\}^{|S|}, \tag{B.1f}$$

$$x \in X^*(p). \tag{B.1g}$$

Models (XPL-Clu) and (XPL-Sparse) are obtained by replacing constraints (B.1d)–(B.1f) with those of $\mathcal{P}^{\mathcal{C}}_{\text{cluster}}$ or $\mathcal{P}_{\text{sparse}}$.

B.3 Binary Expansion for (XPL-Unit)

In (XPL-Unit), each integer variable is expanded as $u^s = \sum_{j=0}^{N_{\text{bin}}-1} 2^j \beta^s_j$, with $\beta^s_j \in \{0,1\}$ and $N_{\text{bin}} = \lfloor \log_2 |S| \rfloor + 1$. Constraint $u^s \leq |S|\, v^s$ is kept, since the expansion can represent values up to $2^{N_{\text{bin}}} - 1 \geq |S|$. The bilinear terms of constraint (4.1g) then read

$$p^s (f^s)^\top y^s = \frac{1}{|S|} \sum_{j=0}^{N_{\text{bin}}-1} 2^j\, \beta^s_j (f^s)^\top y^s,$$

and each product $\beta^s_j (f^s)^\top y^s$ is replaced by a variable z^s_j subject to the four inequalities of Section 4.1, with β^s_j in place of v^s. The same expansion applies to the products $p^s \theta^s$ of (XPL-SLB).

C Preprocessing of the Radial Inequalities

For each scenario $s \in S$, the preprocessing computes the value $c^\top \tilde{x} + Q^s(\tilde{x})$, the threshold $\bar{\kappa}^s$, the set Γ^s of breakpoints of ϕ^s, and the values $\phi^s(\kappa)$ for $\kappa \in \Gamma \cup \{0\}$. Every step solves linear programs of the size of a single scenario TSSP. The last subsection gives the linear program that separates the radial inequalities without preprocessing, and explains why we do not use it.

C.1 A Dual Formula for the Functions ϕ^s

Let $s \in S$ and $\kappa \geq 0$. Since $\phi^s(\kappa)$ is finite, linear programming duality applied to (A.9) gives

$$\phi^s(\kappa) = \max\left\{b^\top\lambda + (h^s)^\top\pi - \tilde{x}^\top\zeta \;:\; (\lambda,\pi,\zeta) \in D^s,\; -\kappa\mathbf{1} \leq \zeta \leq \kappa\mathbf{1}\right\}, \tag{C.1}$$

where $D^s := \{(\lambda,\pi,\zeta) : A^\top\lambda + (T^s)^\top\pi \leq c + \zeta,\; (W^s)^\top\pi \leq f^s\}$ does not depend on κ.

C.2 Preprocessing Procedure

For each scenario $s \in S$, the preprocessing first computes $\bar{\kappa}^s$. Since ϕ^s equals $c^\top\tilde{x} + Q^s(\tilde{x})$ from its last breakpoint $\bar{\kappa}^s$ on (Appendix A.7), $\bar{\kappa}^s$ is the smallest $\kappa \geq 0$ such that $\phi^s(\kappa) \geq c^\top\tilde{x} + Q^s(\tilde{x})$, and (C.1) gives the linear program

$$\bar{\kappa}^s = \min_{\kappa\geq 0,\; (\lambda,\pi,\zeta)\in D^s} \left\{\kappa \;:\; -\kappa\mathbf{1} \leq \zeta \leq \kappa\mathbf{1},\; b^\top\lambda + (h^s)^\top\pi - \tilde{x}^\top\zeta \geq c^\top\tilde{x} + Q^s(\tilde{x})\right\}, \tag{C.2}$$

where $Q^s(\tilde{x})$ is computed beforehand.

The preprocessing then computes the other breakpoints of ϕ^s on $[0, \bar{\kappa}^s]$, with the sandwich algorithm of Eisner and Severance (1976). The algorithm only evaluates ϕ^s, and each evaluation solves the linear program (C.1), of the size of a single-scenario TSSP. The number of evaluations is linear in the number $|\Gamma^s|$ of breakpoints of ϕ^s. At the end, every value of $\phi^s(\kappa)$ is known for $\kappa \in \Gamma^s \cup \{0\}$.

The preprocessing finally computes the values $\phi^s(\kappa)$, $\kappa \in \Gamma \cup \{0\}$. Up to $\bar{\kappa}^s$, these values follow by linear interpolation between the two closest κ at which ϕ^s is known. Since ϕ^s is known at 0 and at every breakpoint of ϕ^s, the interpolation is exact, and requires no further linear program. Beyond the greatest breakpoint $\bar{\kappa}^s$, $\phi^s(\kappa) = c^\top\tilde{x} + Q^s(\tilde{x})$ by Appendix A.7.

The linear programs are solved up to numerical tolerances, and an overestimated value $\phi^s(\kappa)$ could cut off feasible solutions. We therefore decrease each value $\phi^s(\kappa)$ by $10^{-5}\big(1 + |c^\top\tilde{x} + Q^s(\tilde{x})|\big)$. The sandwich algorithm detects breakpoints up to a relative tolerance of 10^{-7}. The callback adds a radial inequality only when the violation exceeds $10^{-6}\big(1 + |c^\top\bar{x} + \sum_{s\in S}\sum_{j\in J}\omega_j^s\bar{z}_j^s|\big)$.

C.3 Separation by a Linear Program

Let $(\bar{p}, \bar{x}, \bar{z}, \bar{t})$ be a solution of the linear relaxation of (4.3). By (C.1), the largest violation over $\kappa \geq 0$ of inequality (4.4), with $\|x - \tilde{x}\|_1$ linearized by $\sum_i t_i$, is the optimal value of the linear program

$$\begin{aligned} \max_{\kappa\geq 0,\; (\lambda^s,\pi^s,\zeta^s)_{s\in S}} \quad & \sum_{s\in S}\bar{p}^s\big(b^\top\lambda^s + (h^s)^\top\pi^s - \tilde{x}^\top\zeta^s\big) - \kappa\sum_i \bar{t}_i - c^\top\bar{x} - \sum_{s\in S}\sum_{j\in J}\omega_j^s\bar{z}_j^s \\ \text{s.t.} \quad & (\lambda^s,\pi^s,\zeta^s) \in D^s, \quad -\kappa\mathbf{1} \leq \zeta^s \leq \kappa\mathbf{1}, \qquad s \in S. \end{aligned} \tag{C.3}$$

Table 6: The three TSSP instance families. n_1 and n_2: numbers of first-stage and second-stage variables per scenario.

Family	Source	n_1	n_2	Uncertainty
FP	Birge and Louveaux (2011)	100	300	T^s
SCP	Rathi et al. (2024)	10	220	h^s
RAP	Lefebvre et al. (2026)	10–20	220–840	h^s

Some radial inequality is violated if and only if this value is positive, and an optimal κ then gives the most violated one. Linear program (C.3) contains one copy of the variables and constraints of (C.1) per scenario, coupled by κ. Linear program (C.3) is thus $|S|$ times larger than (C.1), and at least as large as the TSSP. We do not use (C.3), since the separation is called many times during the branch-and-bound, and would then solve such a linear program at every call.

D Instance Families

This appendix details the three families of TSSPs of Section 6.1, summarized in Table 6. Each family is described in three parts: its formulation, the bounds L^s and U^s, and the generation of its data. The formulations are stated with inequality constraints for readability. They take the form (2.1) once slack variables are introduced, and the sizes of Table 6 refer to the formulations below. In this appendix, the symbols A, b, C, d, F, g, K, r, t, u, z, β, γ and τ are redefined locally and are unrelated to their use elsewhere in the paper. The bounds are those required by the linearizations of Section 4.1. They bound the second-stage cost of each scenario: $L^s \leq Q^s(x) \leq U^s$ must hold for every $x \in X_{\mathcal{P}}$ (Proposition 4.1). The bounds derived below hold for every $x \in X$, hence on $X_{\mathcal{P}}$.

D.1 Farmer's Problem

The example of Birge and Louveaux (2011). A farmer allocates 500 acres among wheat, corn and sugar beets, with the data of Table 7. The farmer must feed the cattle with 200 T of wheat and 240 T of corn, which can be grown or bought. Any surplus is sold. Sugar beets are sold at 36 per ton up to a quota of 6000 T, and at 10 per ton beyond. They cannot be bought in the original example; we set their purchase price to 37, just above their selling price, so that buying them is never profitable. Example 2.3 considers nine equally likely scenarios, in which all yields are multiplied by 1.45, 1.5 or 1.55 (good scenarios), 0.95, 1 or 1.05 (average scenarios), and 0.45, 0.5 or 0.55 (poor scenarios). The optimal allocation is $\tilde{x} = (213.1, 80.0, 206.9)$ acres of wheat, corn and sugar beets.

Model. The FP family generalizes this example to a set C of n_C crops. Crop c costs g_c per unit of area and yields γ_c^s units in scenario s. Once the yields are known, the farmer

Table 7: Data of the farmer's problem (Birge and Louveaux 2011).

	Wheat	Corn	Sugar beets
Mean yield (T/acre)	2.5	3	20
Planting cost (per acre)	150	230	260
Selling price (per T)	170	150	36 (10 beyond 6000 T)
Purchase price (per T)	238	210	37
Requirement (T)	200	240	0

sells at price r_c up to a quantity $\bar{\ell}_c$, and at price $\bar{r}_c \le r_c$ beyond it, and can buy at price $b_c \ge r_c$ to meet a requirement R_c. The problem reads

$$\begin{array}{lll} \min\limits_{x} & \sum\limits_{c\in C} g_c x_c + \sum\limits_{s\in S} p^s Q^s(x) \quad \text{with} \quad Q^s(x) = \min\limits_{\ell,u,q} & \sum\limits_{c\in C} \big(b_c q_c - r_c \ell_c - \bar{r}_c u_c\big) \\ \text{s.t.} & \sum\limits_{c\in C} x_c \le A, & \text{s.t.} \quad \ell_c \le \bar{\ell}_c, \qquad c \in C, \\ & x \ge 0, & \gamma_c^s x_c + q_c \ge R_c + \ell_c + u_c, \quad c \in C, \\ & & \ell, u, q \ge 0, \end{array}$$

where x_c is the area allocated to crop c, and ℓ_c, u_c and q_c are the limited sales, unlimited sales and purchases of crop c in scenario s. The example above corresponds to $n_C = 3$, $A = 500$, $\bar{r}_c = r_c$ for wheat and corn, which makes their sale limit irrelevant, and $R_c = 0$ for sugar beets. The uncertainty enters through the yields, hence through the matrices T^s of formulation (2.1).

Bounds. Since $b_c \ge r_c \ge \bar{r}_c$, buying a unit to resell it is never profitable. An optimal recourse thus buys the deficit of each crop, and sells its surplus at price r_c up to $\bar{\ell}_c$ and at price $\bar{r}_c$ beyond. Hence, with $(a)^+ := \max\{a, 0\}$,

$$Q^s(x) = \sum_{c\in C} \Big(b_c (R_c - \gamma_c^s x_c)^+ - r_c \min\big\{(\gamma_c^s x_c - R_c)^+, \bar{\ell}_c\big\} - \bar{r}_c (\gamma_c^s x_c - R_c - \bar{\ell}_c)^+ \Big).$$

Each term of the sum is nonincreasing in x_c and equals $b_c R_c$ at $x_c = 0$, so $Q^s(x) \le U^s := \sum_{c\in C} b_c R_c$. Each term is also at least $-r_c \gamma_c^s x_c$. As $\sum_{c\in C} x_c \le A$, this gives $Q^s(x) \ge L^s := -A \max_{c\in C} \gamma_c^s r_c$.

Instance generation. The FP instances have $n_C = 100$ crops and $A = 1000$. Crops are partitioned into 33 blocks of three or four crops, and each crop is either a high-volume crop or a grain crop. Crops are drawn in order: a crop becomes high-volume with probability 0.4 if no crop of its block is high-volume yet, and grain otherwise. Each block thus contains at most one high-volume crop. High-volume crops have a mean yield uniform on $[15, 25]$, low unit prices and a binding sale limit. Grain crops have a mean yield uniform on $[2, 4]$, high unit prices and no sale limit. Scenario yields are log-normal with the prescribed mean and a log-standard deviation of 0.2, that is, a coefficient of variation of about 20%. They

Table 8: Costs and prices of the FP instances, drawn uniformly on the given ranges and rounded to integers.

	High-volume crops	Grain crops
Planting cost g_c	$[220, 300]$	$[170, 250]$
Selling price r_c	$[28, 43]$	$[140, 180]$
Price beyond the limit $\bar{r}_c$	$r_c \times [0.2, 0.3]$	$= r_c$
Purchase price b_c	$r_c + [1, 6]$	$r_c + [20, 60]$

are correlated through a Gaussian copula with pairwise correlation 0.85 within a block and 0.20 across blocks, and rounded to two decimals. Costs and prices are drawn on the ranges of Table 8. The sale limit $\bar{\ell}_c$ of a high-volume crop is its maximal production, over the scenarios, on an area drawn uniformly between $2A/n_C$ and $4A/n_C$. Requirements are set on 33 crops, taken among grain crops first. For such a crop, R_c is between 40% and 70% (uniformly) of its worst-case production on an area A/n_C.

D.2 Supply Chain Planning

Model. The SCP family generalizes the supply chain example of Rathi et al. (2024, Sec. 4.1), which has two plants and three customers. A set F of n_F plants supplies a set C of n_C customers. Plant f produces at cost g_f per unit, up to a capacity K_f. Once the demands are known, it ships to customer c at cost τ_{fc} per unit. The demand d_c^s of customer c in scenario s can also be met by external purchases, at cost b_c per unit. The problem reads

$$\begin{aligned}
\min_{x} \quad & \sum_{f\in F} g_f x_f + \sum_{s\in S} p^s Q^s(x) \quad \text{with} \quad Q^s(x) = \min_{t,q} \quad \sum_{f\in F}\sum_{c\in C} \tau_{fc} t_{fc} + \sum_{c\in C} b_c q_c \\
\text{s.t.} \quad & 0 \le x_f \le K_f, \quad f \in F, \qquad\qquad \text{s.t.} \quad \sum_{c\in C} t_{fc} \le x_f, \qquad f \in F, \\
& \qquad\qquad\qquad\qquad\qquad\quad \sum_{f\in F} t_{fc} + q_c \ge d_c^s, \qquad c \in C, \\
& \qquad\qquad\qquad\qquad\qquad\quad t, q \ge 0,
\end{aligned}$$

where x_f is the production of plant f. In scenario s, t_{fc} is the quantity shipped from plant f to customer c, and q_c the quantity bought externally for customer c. The uncertainty enters through the demands, hence through the right-hand sides h^s of formulation (2.1).

Bounds. All costs are nonnegative, so $Q^s(x) \ge L^s := 0$. Buying the whole demand externally is feasible for every $x \in X$, so $Q^s(x) \le U^s := \sum_{c\in C} b_c d_c^s$.

Instance generation. The SCP instances have $n_F = 10$ plants and $n_C = 20$ customers. Production costs g_f, transportation costs τ_{fc} and external purchase costs b_c are integers drawn uniformly on $\{80, \dots, 140\}$, $\{50, \dots, 100\}$ and $\{250, \dots, 350\}$, respectively. Producing

and shipping a unit thus costs at most 240, less than any external purchase. The expected demand of each customer is 20 in all three groups. All plants have the same capacity $K_f = 80$, so that the total capacity is twice the expected total demand. The three groups differ only in the demand distribution. In SCP0, demands are independent and uniform on $[5, 35]$. In SCP1, each demand is independently low with probability $1/2$, uniform on $[5, 15]$, and high otherwise, uniform on $[25, 35]$. In SCP2, demands keep the uniform marginals of SCP0, but are correlated through a Gaussian copula with pairwise correlation 0.9. All demands are rounded to two decimals.

D.3 Resource Allocation

Model. The RAP family adapts a test problem of Lefebvre et al. (2026) to the stochastic setting. A set F of n_F facilities serves a set C of n_C customers. Facility f produces at cost g_f per unit. Once the demands are known, each unit of production of facility f allocated to customer c covers β_{fc} units of its demand. The demand d_c^s of customer c in scenario s can also be met by external purchases, at cost b_c per unit. The problem reads

$$\begin{aligned}
\min_{x} \quad & \sum_{f\in F} g_f x_f + \sum_{s\in S} p^s Q^s(x) \quad \text{with} \quad Q^s(x) = \min_{z,q} \quad \sum_{c\in C} b_c q_c \\
\text{s.t.} \quad & x \geq 0, && \text{s.t.} \quad \sum_{c\in C} z_{fc} \leq x_f, && f \in F, \\
& && \sum_{f\in F} \beta_{fc} z_{fc} + q_c \geq d_c^s, && c \in C, \\
& && z, q \geq 0,
\end{aligned}$$

where x_f is the production of facility f. In scenario s, z_{fc} is the production of facility f allocated to customer c, and q_c the quantity bought externally for customer c. As in SCP, the uncertainty enters through the demands, hence through the right-hand sides h^s. RAP differs from SCP in three respects: production is not capacitated, allocating it to customers has no cost, and a unit allocated to customer c covers β_{fc} units of its demand, instead of exactly one. Without capacities, X is unbounded. Assumption 2.2 still holds: since production costs are positive, producing beyond what the largest demands require is never optimal.

Bounds. The arguments of Appendix D.2 apply unchanged, and give $L^s := 0$ and $U^s := \sum_{c\in C} b_c d_c^s$.

Instance generation. Each RAP TSSP is built from a deterministic instance of this test problem, drawn at random among the available ones, which have $n_F \in \{10, 15, 20\}$ facilities and $n_C \in \{20, 30, 40\}$ customers. The selected instances are listed in the repository. We use their production costs g_f, service rates β_{fc} and nominal demands $\bar{d}_c$, and neglect

congestion. Demands are drawn independently and uniformly on $[\bar{d}_c, 1.2\,\bar{d}_c]$, and rounded to four decimals. External purchase costs are set to $b_c = 5 \min\{g_f/\beta_{fc} : \beta_{fc} > 0\}$, five times the cheapest production cost of one unit of demand of customer c, so that external purchase is a last resort.

References

Ravindra K Ahuja and James B Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, 2001. doi: 10.1287/opre.49.5.771.10607.

Nurşen Aydın, Ş İlker Birbil, İlker Küçükparlak, and Altuğ Yalçıntaş. Explainable optimization: A call for interdisciplinary action, 2026. Preprint, arXiv:2606.08675.

Dimitris Bertsimas and Nishanth Mundru. Optimization-based scenario reduction for data-driven two-stage stochastic optimization. *Operations Research*, 71(4):1343–1361, 2023. doi: 10.1287/opre.2022.2265.

Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi: 10.1137/141000671.

John R Birge and Francois Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 2nd edition, 2011. doi: 10.1007/978-1-4614-0237-4.

François Clautiaux and Ivana Ljubić. Last fifty years of integer linear programming: A focus on recent practical advances. *European Journal of Operational Research*, 324(3):707–731, 2025. doi: 10.1016/j.ejor.2024.11.018.

Kristijonas Čyras, Dimitrios Letsios, Ruth Misener, and Francesca Toni. Argumentation for explainable scheduling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):2752–2759, 2019. doi: 10.1609/aaai.v33i01.33012752.

Stephan Dempe. *Foundations of Bilevel Programming*, volume 61 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, 2002. doi: 10.1007/b101970.

Jitka Dupačová, Nicole Gröwe-Kuska, and Werner Römisch. Scenario reduction in stochastic programming: An approach using probability metrics. *Mathematical Programming*, 95(3): 493–511, 2003. doi: 10.1007/s10107-002-0331-0.

Mark J Eisner and Dennis G Severance. Mathematical techniques for efficient record segmentation in large shared databases. *Journal of the ACM (JACM)*, 23(4):619–635, 1976. doi: 10.1145/321978.321982.

Felix Engelhardt, Jannis Kurtz, Ş İlker Birbil, and Ted Ralphs. Counterfactual explanations for integer optimization problems, 2026. Preprint, arXiv:2510.17624.

Jonathan Ferrer-Mestres, Thomas G. Dietterich, Olivier Buffet, and Iadine Chadès. Interpretable solutions for stochastic dynamic programming, 2024. Preprint, bioRxiv.

Alexandre Forel, Axel Parmentier, and Thibaut Vidal. Explainable data-driven optimization: From context to decision and back again. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10170–10187. PMLR, 2023. URL https://proceedings.mlr.press/v202/forel23a.html.

Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975. doi: 10.1287/mnsc.22.4.455.

Marc Goerigk and Michael Hartisch. A framework for inherently interpretable optimization models. *European Journal of Operational Research*, 310(3):1312–1324, 2023. doi: 10.1016/j.ejor.2023.04.013.

Emilia Graß and Kathrin Fischer. Two-stage stochastic programming in disaster management: A literature survey. *Surveys in Operations Research and Management Science*, 21(2):85–100, 2016. doi: 10.1016/j.sorms.2016.11.002.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026. URL https://www.gurobi.com.

Holger Heitsch and Werner Römisch. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24(2):187–206, 2003. doi: 10.1023/A:1021805924152.

Mike Hewitt, Janosch Ortmann, and Walter Rei. Decision-based scenario clustering for decision-making under uncertainty. *Annals of Operations Research*, 315(2):747–771, 2022. doi: 10.1007/s10479-020-03843-x.

Tito Homem-de-Mello, Juan Valencia, Felipe Lagos, and Guido Lagos. Forecasting outside the box: Application-driven optimal pointwise forecasts for a class of two-stage stochastic programs. *Mathematical Programming*, 2026. doi: 10.1007/s10107-026-02359-9. Forthcoming.

Leonard Kaufman and Peter Rousseeuw. Clustering by means of medoids. In Yadolah Dodge, editor, *Statistical Data Analysis Based on the L_1-Norm and Related Methods*, pages 405–416. North-Holland, 1987.

Julien Keutchayan, Janosch Ortmann, and Walter Rei. Problem-driven scenario clustering in stochastic optimization. *Computational Management Science*, 20(1):13, 2023. doi: 10.1007/s10287-023-00446-2.

Anton Korikov and J Christopher Beck. Counterfactual explanations via inverse constraint programming. In *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*, volume 210 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.CP.2021.35.

Anton Korikov and J Christopher Beck. Objective-based counterfactual explanations for linear discrete optimization. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2023)*, volume 13884 of *Lecture Notes in Computer Science*, pages 18–34. Springer, 2023. doi: 10.1007/978-3-031-33271-5_2.

Jannis Kurtz, Ş. İlker Birbil, and Dick den Hertog. Counterfactual explanations for linear optimization. *European Journal of Operational Research*, 329(1):24–41, 2026. doi: 10.1016/j.ejor.2025.06.016.

Henri Lefebvre and Martin Schmidt. Computing weak counterfactual explanations for linear optimization: A new class of bilevel models and a tailored penalty alternating direction

method. *European Journal of Operational Research*, 333(2):534–544, 2026. doi: 10.1016/j.ejor.2026.02.007.

Henri Lefebvre, Enrico Malaguti, and Michele Monaci. Exact approaches for convex adjustable robust optimization. *Mathematical Programming*, 218(1):439–476, 2026. doi: 10.1007/s10107-025-02311-3.

Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982. doi: 10.1109/TIT.1982.1056489.

Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 15(3):581–589, 2023. doi: 10.1007/s12532-023-00239-3.

Sara Lumbreras and Pedro Ciller. Interpretable optimization: Why and how we should explain optimization models. *Applied Sciences*, 15(10):5732, 2025. doi: 10.3390/app15105732.

Stephen Maher. Implementing the branch-and-cut approach for a general purpose Benders' decomposition framework. *European Journal of Operational Research*, 290(2):479–498, 2021. doi: 10.1016/j.ejor.2020.08.037.

Donato Maragno, Jannis Kurtz, Tabea E. Röber, Rob Goedhart, Ş. İlker Birbil, and Dick den Hertog. Finding regions of counterfactual explanations via robust optimization. *INFORMS Journal on Computing*, 36(5):1316–1334, 2024. doi: 10.1287/ijoc.2023.0153.

Jasone Ramírez-Ayerbe and Emma Frejinger. Relative explanations for contextual problems with endogenous uncertainty: An application to competitive facility location, 2026. Preprint, arXiv:2506.19155.

Tushar Rathi, Rishabh Gupta, Jose Pinto, and Qi Zhang. Enhancing explainability of stochastic programming solutions via scenario and recourse reduction. *Optimization and Engineering*, 25(2):795–820, 2024. doi: 10.1007/s11081-023-09825-9.

Marius Roland, Alexandre Forel, and Thibaut Vidal. Adaptive partitioning for chance-constrained problems with finite support. *SIAM Journal on Optimization*, 35(1):476–505, 2025. doi: 10.1137/24M1632772.

Werner Römisch. Stability of stochastic programming problems. In Andrzej Ruszczyński and Alexander Shapiro, editors, *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 483–554. Elsevier, 2003. doi: 10.1016/S0927-0507(03)10008-4.

Napat Rujeerapaiboon, Kilian Schindler, Daniel Kuhn, and Wolfram Wiesemann. Scenario reduction revisited: Fundamental limits and guarantees. *Mathematical Programming*, 191(1): 207–242, 2022. doi: 10.1007/s10107-018-1269-1.

Tjendera Santoso, Shabbir Ahmed, Marc Goetschalckx, and Alexander Shapiro. A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115, 2005. doi: 10.1016/j.ejor.2004.01.046.

Aaron Schild, Sreenivas Gollapudi, Anupam Gupta, Kostas Kollias, and Ali Sinop. *Why is My*

Route Different Today? An Algorithm for Explaining Route Selection, pages 251–263. SIAM, 2025. doi: 10.1137/1.9781611979084.19.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory, Third Edition*. SIAM, 2021. doi: 10.1137/1.9781611976595.

Yongjia Song and James Luedtke. An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. *SIAM Journal on Optimization*, 25(3):1344–1367, 2015. doi: 10.1137/140967337.

Wim van Ackooij, Welington de Oliveira, and Yongjia Song. Adaptive partition-based level decomposition methods for solving two-stage stochastic programs with fixed recourse. *INFORMS Journal on Computing*, 30(1):57–70, 2018. doi: 10.1287/ijoc.2017.0765.

R. M. Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969. doi: 10.1137/0117061.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2):841–887, 2018. doi: 10.2139/ssrn.3063289.

Wei Zhang, Kai Wang, Alexandre Jacquillat, and Shuaian Wang. Optimized scenario reduction: Solving large-scale stochastic programs with quality guarantees. *INFORMS Journal on Computing*, 35(4):886–908, 2023. doi: 10.1287/ijoc.2023.1295.