# Forecasting Outside the Box: Application-Driven Optimal Pointwise Forecasts for Stochastic Optimization

Tito Homem-de-Mello[1], Juan Valencia[2],  Felipe Lagos[3], and  Guido Lagos[4]

[1]*School of Business, Universidad Adolfo Ibáñez, Chile*
[2]*School of Business, Universidad Adolfo Ibáñez, Chile*
[3]*Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Chile*
[4]*Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Chile*

November 8, 2024

**Abstract**

The exponential growth in data availability in recent years has led to new formulations of data-driven optimization problems. One such formulation is that of stochastic optimization problems with *contextual information*, where the goal is to optimize the expected value of a certain function given some contextual information (also called *features*) that accompany the main data of interest. The contextual information then allows for a better estimation of the quantity of interest via machine learning methods, thereby leading to better solutions. Oftentimes, however, machine learning methods yield just a *pointwise estimate* instead of an entire distribution. In this paper we show that, when the problem to be solved is a class of two-stage stochastic programs (namely, those with fixed recourse matrix and fixed costs), under mild assumptions the problem can be solved with just *one* scenario. While such a scenario—which does not have be unique—is usually unknown, we present an integrated learning and optimization procedure that yields the best approximation of that scenario within the modeler's pre-specified set of parameterized forecast functions. Numerical results conducted with inventory problems from the literature (with synthetic data) as well as a bike-sharing problem with real data demonstrate that the proposed approach performs well when compared to benchmark methods from the literature.

## 1  Introduction

The area of stochastic optimization has evolved considerably in the past decade. Traditionally, stochastic optimization models assumed the existence of a known probability distribution to represent the underlying uncertainty, and formulated the problem in terms of optimizing the expected value (or another risk measure) of a certain function with respect to a decision variable, where the expectation corresponds to that distribution. One can formulate the problem in a generic way as

$$\min_{z \in Z} \; \mathbb{E}_P \left[ G(z, \xi) \right], \tag{1}$$

where $z$ is the decision variable, $\xi$ represents the uncertainty, and $P$ is the distribution of $\xi$. Much of the earlier efforts aimed at developing *scenario generation/reduction techniques* for the case where $P$ has either large or infinite support, in order to approximate the original problem with one in which the distribution

has moderately-sized support and hence decomposition methods can be used to solve the problem. Among the scenario generation methods are clustering (Dupačová, Consigli, and Wallace, 2000), moment-matching techniques (Hoyland and Wallace, 2001; Hoyland, Kaut, and Wallace, 2003; Mehrotra and Papp, 2014), and Monte Carlo/Quasi-Monte Carlo methods (Shapiro, Dentcheva, and Ruszczynski, 2021, Chapter 5), (Homem-de-Mello and Bayraksan, 2014; Leövey and Römisch, 2015). Another class of methods, based on probability metrics, aims at finding a distribution $Q$ with relatively few scenarios in such a way that $Q$ minimizes a distance $d(P, Q)$ between $Q$ and the original distribution $P$. We refer to Dupačová, Gröwe-Kuska, and Römisch (2003); Heitsch and Römisch (2003); Heitsch and Römisch (2009); Pflug (2001); Pflug and Pichler (2011) and references therein for further discussions on this type of methods.

Recent efforts have been directed to developing scenario generation/reduction techniques that use the information from the optimization problem at hand rather using just the distribution of the underlying random variables. Such methods typically focus on two-stage stochastic programs in order to exploit the structure of the problem. For instance, Bertsimas and Mundru (2023) define the distance between two probability distributions in terms of the cost functions of the optimization problem. Henrion and Römisch (2022) use problem information to compute the distribution that leads to the best uniform approximation of $\mathbb{E}_P[G(z, \xi)]$ over all feasible $z$, whereas Zhang, Wang, Jacquillat, and Wang (2023) propose a scenario subset selection model that optimizes the approximation of the recourse function over a pool of first-stage solutions. Keutchayan, Ortmann, and Rei (2023) develop a problem-driven scenario clustering method that produces a partition of the scenario set that enables representative scenarios to be identified. A different approach is used in Arpón, Homem-de-Mello, and Pagnoncelli (2018) and Fairbrother, Turner, and Wallace (2019), who work with a variation of (1) where the objective function is a tail risk measure such as Conditional Value-at-Risk, and develop scenario generation methods that exploit the structure of that objective.

In the past few years, considerable attention has been given to the case where one does not know in advance the probability distribution; rather, only *data* is available. Of course, one possible approach in such cases is to fit a distribution $\hat{P}$ to the data using standard techniques such as Maximum Likelihood Estimation, and then solve the stochastic optimization problem using $\hat{P}$ as the "true" distribution. An alternative approach for the data-driven case is to use *distributionally robust optimization* (DRO); here, the goal is to find the solution that minimizes the worst expectation among a set of distributions, called the ambiguity set. In the data-driven setting the ambiguity set is often given by the set of distributions that are close to the empirical distribution corresponding to the data; such an approach, discussed for instance in Mohajerin Esfahani and Kuhn (2018) and Blanchet, Kang, and Murthy (2019), has the advantage of mitigating "overfitting" to the data, thereby yielding better out-of-sample performance of the solutions.

The enormous growth in the availability of data in recent years—more specifically, the presence of *contextual information* in the data (also called *covariates*, or *features* in the literature)— has led to the development of new models in stochastic optimization. In such models the uncertainty represented by $\xi$ can be predicted to some extent by the available contextual information. Thus, the goal is to optimize the expected value of a certain function conditionally on some given value of the contextual information, henceforth called the *contextual information of interest*. That is, the goal is to find the best decision corresponding to data with some characteristics. An example of such a situation will be discussed in Section 6; the goal is there to determine the best assignment of bicycles on a given day for a bike-sharing service, given some contextual information such as the weather forecast for that day. Formally, we can write the problem as

$$\min_{z \in Z} \mathbb{E}\left[G(z, \xi) \,|\, X = x\right], \tag{2}$$

where again $z \in \mathbb{R}^n$ is the decision variable, $Z \subseteq \mathbb{R}^n$ is the feasibility set, $\xi$ is a random variable with support

$\Xi \subseteq \mathbb{R}^m$ that represents the main uncertainty, $X$ is a random variable with support $\mathcal{X} \subseteq \mathbb{R}^s$ that represents the contextual information and $x$ is the contextual information of interest.

The difficulty of optimizing by taking into account the presence of contextual information as in (2) is how to define a proper *conditional* distribution of $\xi$ given $X = x$ to use in the model. Simple techniques such as "slicing" the data to keep only the data points corresponding to the contextual information of interest are not practical since the resulting dataset may be too small or even empty, which occurs in case the particular contextual information of interest has not been observed in the dataset. An alternative in such cases is to use *machine learning* approaches that can forecast the uncertainty as a function of the contextual information. Modern machine learning methods such as neural networks, regularized regression and classification trees, among others, can be used to learn the dependence of $\xi$ on $x$ (see, e.g., Bottou, Curtis, and Nocedal (2018); Hastie, Tibshirani, and Friedman (2009)). The key then is how to combine these predictions with the optimization model.

The use of machine-learning-based forecasts for optimization can be accomplished in multiple ways, and has become a fruitful topic in stochastic optimization with many papers in the past five years. As discussed in the recent survey paper by Sadana, Chenreddy, Delage, Forel, Frejinger, and Vidal (2024), three main approaches can be found in the literature: (i) *decision-rule optimization*, which aims to approximate directly the optimal solutions of (2) as a function of $x$ by means of techniques such as linear decision rules or reproducing kernel Hilbert spaces; (ii) *sequential learning and optimization*, which uses machine learning techniques to estimate the conditional distribution of the uncertainty given the contextual information of interest, and then applies standard methods to solve the stochastic optimization problem corresponding to that conditional distribution; and (iii) *integrated learning and optimization*, where the forecast and optimization are combined within the same problem. For instance, in the SPO (smart "predict, then optimize") framework of Elmachtoub and Grigas (2021), for a given feature $x$ the problem is written as

$$\min_{z \in Z} \mathbb{E}\left[\xi^T z \,|\, X = x\right] \;=\; \min_{z \in Z} \left(\mathbb{E}\left[\xi|x\right]\right)^T z, \tag{3}$$

where we use $\mathbb{E}\left[\xi|x\right]$ as a short for $\mathbb{E}\left[\xi \,|\, X = x\right]$. Note that the linearity of the above model implies that in order to solve the problem we only need an estimate $\xi_x$ of $\mathbb{E}\left[\xi|x\right]$, i.e., a pointwise forecast. The key idea of the SPO approach is to measure the *decision error* induced by the estimation error of $\mathbb{E}\left[\xi|x\right]$), and to measure the performance of the prediction in terms of its impact in the objective function instead of using a standard error criterion such as least-squares. Such an idea can actually be traced back to Bengio (1997) but has gained traction in recent years—albeit with different names such as *integrated conditional expectation and optimization* (Grigas, Qi, and Shen, 2021), *end-to-end learning* (Donti, Amos, and Kolter, 2017), *application-driven learning* (Dias-Garcia, Street, Homem-de-Mello, and Muñoz, 2024), and *decision-focused learning* (Mandi, Bucarey, Tchomba, and Guns, 2022), in addition to the other terminology mentioned above. The idea has gone even beyond scientific papers; for instance, a recent Harvard Business Review article describes the implementation of a forecasting methodology for supply chains, called *optimal machine learning* by the authors, that "involves using artificial intelligence technology to create a mathematical model that takes data inputs [...] and links them to planning decisions" (Agrawal, Cohen, Deshpande, and Deshpande, 2024).

The goal of this is paper to solve problems of the form (2) when the function $G$ corresponds to a *two-stage stochastic program*, using the problem information to measure the forecast error. Notice that the problem is considerably harder than (3), since in principle we need, as discussed earlier, to forecast the entire conditional distribution of $\xi$ given $X = x$ using a problem-based approach. This is in fact the approach used by Grigas et al. (2021), although the approach has some limitations such as fixing in advance the support of the distribution. We show however that forecasting the entire conditional distribution is not necessary

for an important class of problems. More specifically, we consider two-stage stochastic programs with fixed recourse and fixed costs (henceforth denoted FRFC), i.e.,

$$\min_{z \in Z} \ f(z) + \mathbb{E}\left[Q(z,\xi)\right] \tag{4}$$

where $Z \subseteq \mathbb{R}^n$ is a convex set, $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex function such that $Z \subseteq \text{Dom } f$, $Q$ is the second stage function

$$Q(z,\xi) = \quad \min \ q^\top y \tag{5}$$
$$\text{s.t.} \quad Wy = h - Tz \tag{6}$$
$$y \geq 0 \tag{7}$$

and $\xi$ denotes the random element $(h, T)$. The qualifier "fixed recourse and fixed costs" refers to the fact that in the second-stage problem (5)-(7) neither the matrix $W$ nor the vector $q$ are random. The class of two-stage stochastic programs with FRFC is actually quite large, as it includes problems where the uncertainty corresponds to demand, as is the case in many inventory, energy, capacity planning and logistics problems; in fact, the vast majority of two-stage test problems reported in the literature, or available in public repositories such as `stoprog.org`, are of FRFC type. A notable exception consists of problems where the coefficient $q$ corresponds to random prices or returns, as in the case of portfolio models for example.

A key result of the paper is the proof that, under mild assumptions, *the two-stage stochastic program given in (4)-(7) can be solved with only one scenario,* in the sense that there exists one scenario $\xi^* = (h^*, T^*)$ (possibly outside the support of $\xi$) such that solving the problem $\min_{z \in Z} \ f(z) + Q(z, \xi^*)$ yields an optimal solution that is also optimal for (4)-(7). In other words, it suffices to solve the simpler problem

$$\min_{z \in Z} \ f(z) + q^\top y \tag{8}$$
$$\text{s.t.} \quad T^* z + Wy = h^* \tag{9}$$
$$y \geq 0. \tag{10}$$

instead of (4)-(7). This is a surprising result, which to best of our knowledge has not been shown in the literature. In fact, one of the main arguments for solving stochastic optimization problems such as (1) instead of the simpler one-scenario problem $\min_{u \in U} \ G(u, \bar{\xi})$ for some fixed scenario $\bar{\xi}$ is the fact that the one-scenario problem does not capture the variability of the uncertainty. For instance, Wallace (2000) presents a compelling argument by means of very simple examples of stochastic optimization problems for which solving the problem for one scenario (regardless of the choice of the scenario) can never yield the same solution as the stochastic one.

So, how to reconcile our main result with the proven need to use the full distribution of the uncertainty? As we shall see later in the paper, the key lies in the particular characteristics of two-stage stochastic programs with FRFC. In light of that result, it is not surprising that none of examples presented in Wallace (2000) are two-stage stochastic programs with FRFC, so there is no contradiction between our result and the conclusions in Wallace (2000)[1]. Our result can also be viewed as a generalization of a property that holds for the well known newsvendor problem, which is a particular case of a two-stage stochastic program with FRFC—indeed, if we solve the newsvendor model with one scenario that coincides with the critical value $F^{-1}(\frac{c_u}{c_u+c_o})$, where

---

[1]Actually, Example 1 in Wallace (2000) could be formulated as a very special case of a two-stage stochastic program with FRFC where, in the notation of (5)-(7), the function $Q$ is either 0 or $\infty$. Such a model however does not have relatively complete recourse, a common assumption that we also make in our developments.

$c_u$ and $c_o$ are respectively underage and overage costs and $F$ is the true distribution, then the optimal solution of the one-scenario problem is the same as the optimal solution of the original problem. While such a property appears obvious in the case of the newsvendor model, our results generalizes it for all two-stage stochastic programs with FRFC. We emphasize that the "optimal scenario" $\xi^*$ might actually lie *outside* the support $\Xi$; that is, the model might produce a statistically impossible pointwise forecast, but which ultimately leads to the optimal solution of the original problem. This is what the expression "forecasting outside the box" in the title of the paper alludes to.

The one-scenario result, while appealing from a theoretical perspective, has its caveats. One is that the "optimal scenario" does not respect any dependencies (either functional or statistical ones) among the random variables; we will discuss this issue in more detail later. Another caveat is that it is a result about *existence* of such a scenario; indeed, the result states that the scenario depends on the optimal solution $z^*$ of (4)-(7), so generating that scenario is obviously as hard as solving the original problem. On the other hand, we make no claims about uniqueness of the optimal scenario. For our purposes, it suffices to know that we can search for *some* scenario with the property that solving the deterministic problem with that scenario yields the same solution as the stochastic one.

Those caveats notwithstanding, in the context of (2) where there is contextual information, we shall see that the result becomes quite useful. Indeed, consider a mapping $z^*(x)$ which yields an optimal solution to (2) as a function of $x$. As determining the entire mapping $z^*(\cdot)$ is impractical from a computational perspective, it is natural to think of ways to *approximate* that function, and this is precisely what we do in this paper by approximating the function that maps $x$ to an "optimal scenario" for problem (2). That is, instead of approximating the optimal *solution* mapping as done for instance in Ban and Rudin (2019), we approximate an optimal *scenario* mapping, call it $\xi_x^*$ to emphasize its dependence on $x$. Thus, our approach can be viewed as a bridge between the "integrated learning and optimization" and the "decision rule optimization" methods, in the sense that we aim at producing problem-based biased forecasts but we already know that there exists an optimal forecast which is actually a (linear) function of the optimal solution. Note that in our approach we do not need to deal with distributions, only with parameterized pointwise forecasts $\Psi(\theta, x)$ which constitute the vast majority of forecasts obtained with machine learning techniques. Once an approximation to an optimal scenario mapping (call it $\Psi(\theta^*, x)$) is constructed from training data, we can easily obtain the corresponding solution to (2) for any given $x$ by solving the one-scenario problem in (8)-(10) with $\Psi(\theta^*, x)$ in place of $\xi^*$.

Naturally, the task of approximating an optimal scenario mapping is not simple. This is where the notion of *application-driven forecasts* developed in Dias-Garcia et al. (2024) becomes key. As discussed earlier, the approach in that paper falls into the category of works that measure the quality of the pointwise forecast in terms of its impact on the optimization problem; in the case of Dias-Garcia et al. (2024), this is accomplished by solving a bi-level problem. A distinctive feature of the approach in Dias-Garcia et al. (2024) that is useful here is the fact the method aims at finding the *best possible values* of the parameters of the forecast function; thus, as we show in the present paper, as long as the class of forecast functions is flexible enough, the pointwise forecast yielded by the algorithm will be a good approximation of an optimal scenario. There is however a trade-off between the flexibility allowed by class of forecast functions and the computational effort required to solve the bi-level model; still, such an effort is spent at the *training* stage—as discussed above, once the optimal parameters are found, solving (2) amounts to solving a simple problem. Our numerical results in Section 6, where we study three problems from the literature—two with synthetic data and one with real data—indicate that the one-scenario forecast actually performs very well.

The remainder of the paper structured as follows: in Section 2 we introduce our main result, which is the

existence of an optimal scenario for two-stage stochastic programs with FRFC. Section 3 discusses optimal solution mappings as a generalization of (2) for multiple values of the contextual information $x$. In Section 4 we present our proposed approach and show that, in the limit, it produces the best possible parameters for given forecast function $\Psi$. We also show conditions under which the forecast will indeed yield a good approximation of an optimal scenario warranted by the result in Section 2. Section 5 presents two specific algorithms that implement our strategy—one based on regression, the other based on classification and regression trees (CART)—and a few algorithms from the literature that are used as benchmarks. Finally, in Section 6 we show numerical results for three problems from the literature. Concluding remarks are presented in Section 7.

## 2   One-scenario optimality

We start by stating a key result of the paper, which will be used in the sequel. It ensures that, under certain conditions, there exists an "optimal scenario" for the two-stage problem given in (4)-(7) such that if the problem is solved only with that scenario, it yields an optimal solution to the original problem.

To proceed, we make the following assumptions:

**Assumption 1.** *The feasibility set $Z$ is non-empty, and the function $f$ defined in (4) is such that the relative interior of its domain (denoted* $\mathrm{ri}(\mathrm{Dom}\, f)$*) is non-empty.*

The assumption on non-emptiness of $Z$ is natural, otherwise the two-stage problem of interest is infeasible. The assumption on $f$ is mild, holding for example if $\mathrm{Dom}\, f = \mathbb{R}^n$.

Consider now the dual problem of (5)-(7):

$$\max\ (h-Tz)^\top u \tag{11}$$

$$\text{s.t.} \qquad W^\top u \leq q \tag{12}$$

$$u \in \mathbb{R}^m. \tag{13}$$

**Assumption 2.** *The feasibility set $U$ defined by (12)-(13) is non-empty and bounded.*

This assumption ensures that $Q(z,\xi)$ is finite for all values of $z$ and $\xi$.

We state now the main result. Note that no assumptions are made about the distribution of the uncertainty.

**Theorem 1.** *Suppose Assumptions 1-2 hold, and let $z^*$ be an optimal solution to (4)-(7). Let $\xi^* = (h^*, T^*)$ be defined such that $T^* := \mathbb{E}[T]$ and $h^* := T^* z^*$. Then, we have that*

$$z^* \in \underset{z \in Z}{\mathrm{argmin}}\ f(z) + Q(z, \xi^*). \tag{14}$$

*Thus, if the optimal solution of the* one-scenario *problem* $\min_{z \in Z} f(z) + Q(z, \xi^*)$ *is unique, then it must coincide with $z^*$.*

*Proof.* Assumption 2 implies that there exist a set $\widetilde{U} := \{u_1, \ldots, u_k\}$ (where each $u_i$ is a vertex of $U$) such that, for any values of $z$ and $\xi$, we have

$$Q(z, \xi)\ =\ \max\{(h - Tz)^\top u\ :\ u \in \widetilde{U}\}. \tag{15}$$

It is well known that $Q(\cdot, \xi)$ is convex for all $\xi$. From (15), define the set

$$\widetilde{U}_z(\xi) := \underset{u \in \widetilde{U}}{\operatorname{argmax}} \, (h - Tz)^\top u. \tag{16}$$

It follows (see, e.g., Rockafellar 1970) that the subdifferential set of $Q$ at $(z, \xi)$ w.r.t $z$ is given by

$$\partial_z Q(z, \xi) = -T^\top \operatorname{conv}(\widetilde{U}_z(\xi)) \subseteq -T^\top \operatorname{conv}(\widetilde{U}), \tag{17}$$

and so by applying the expectation operator to both sides of the above relationship we have that

$$\mathbb{E}\left[\partial_z Q(z, \xi)\right] \subseteq -\mathbb{E}[T]^\top \operatorname{conv}(\widetilde{U}). \tag{18}$$

Now, since $Q$ is a convex function, we have that

$$\partial_z \mathbb{E}\left[Q(z, \xi)\right] = \mathbb{E}\left[\partial_z Q(z, \xi)\right] \tag{19}$$

and thus from (18)-(19) we conclude that

$$\partial_z \mathbb{E}\left[Q(z, \xi)\right] \subseteq -\mathbb{E}[T]^\top \operatorname{conv}(\widetilde{U}). \tag{20}$$

Consider now any optimal solution $z^*$ to (4)-(7). Since such a problem is convex, if follows that the optimality condition for $z^*$ is

$$0 \in \partial_z \big(f(z^*) + \mathbb{E}\left[Q(z^*, \xi)\right]\big) + N_Z(z^*) \tag{21}$$

where $N_Z(z)$ denotes the normal cone of $Z$ at $z \in Z$. By Assumption 1 and 2, we have that

$$\emptyset \subseteq \operatorname{ri}(\operatorname{Dom} f) \cap \operatorname{ri}(\operatorname{Dom} Q(\cdot, \xi)),$$

so the formula $\partial_z \big(f(z) + \mathbb{E}\left[Q(z, \xi)\right]\big) = \partial_z f(z) + \partial_z \mathbb{E}\left[Q(z, \xi)\right]$ applies (Rockafellar, 1970) and then from (20) and (21) we then have that

$$0 \in \partial_z f(z^*) + N_Z(z^*) - \mathbb{E}[T]^\top \operatorname{conv}(\widetilde{U}). \tag{22}$$

Define now $\xi^* = (h^*, T^*)$ such that $T^* := \mathbb{E}[T]$ and $h^* := T^* z^*$. Then, from (16) we have

$$\widetilde{U}_{z^*}(\xi^*) = \widetilde{U} \tag{23}$$

(since any dual solution is optimal in that case) and thus from (17) it follows that

$$\partial_z Q(z^*, \xi^*) = -(T^*)^\top \operatorname{conv}(\widetilde{U}) = -\mathbb{E}[T]^\top \operatorname{conv}(\widetilde{U}). \tag{24}$$

From the above equation, together with (22), we conclude that

$$0 \in \partial_z f(z^*) + N_Z(z^*) + \partial_z Q(z^*, \xi^*), \tag{25}$$

i.e., $z^*$ is an optimal solution to the one-scenario problem $\min_{z \in Z} f(z) + Q(z, \xi^*)$. Clearly, if the latter problem has a unique optimal solution, then it must coincide with $z^*$. $\qquad \square$

Note that the uniqueness of optimal solutions of $\min_{z \in Z} f(z) + Q(z, \xi^*)$ is assured for example when $f$ is *strictly* convex—a property that can be enforced by adding a regularization term if needed.

Theorem 1 shows the existence of an optimal scenario. As discussed earlier, it is clear that such a result is not of immediate use as it depends on optimal solutions of the same problem. In next sections, however, we will see how Theorem 1 plays a fundamental role in the development of approximating policies.

Still, some conclusions can be drawn from Theorem 1. For instance, the special case when $T$ is random with $\mathbb{E}[T] = 0$ leads to a somewhat surprising result, which to the best of our knowledge has not been observed in the literature. Essentially, it says that in that setting the second stage is irrelevant for the first-stage decision. While the result is obvious when $T$ is constant equal to zero—since in that case there is no link between the first and second stages—it is interesting to see that such a property remains valid when $T$ is random with $\mathbb{E}[T] = 0$.

**Corollary 1.** *Suppose Assumptions 1-2 hold. Suppose also that the random matrix $T$ is such that $\mathbb{E}[T] = 0$. Then, if the first-stage problem $\min_{z \in Z} f(z)$ has a unique optimal solution $z^*$, then $z^*$ is an optimal solution of* (4)-(7).

*Proof.* By Theorem 1, the "optimal scenario" $\xi^*$ is given by $T^* = \mathbb{E}[T] = 0$ and $h^* = T^* z^* = 0$. It follows that $Q(z, \xi^*)$ does not depend on $z$ and hence the optimal solutions of the one-scenario problem $\min_{z \in Z} f(z) + Q(z, \xi^*)$ are the same as those of the problem $\min_{z \in Z} f(z)$. Again, Theorem 1 ensures that if the latter problem has a unique optimal solution, then such a solution is optimal for (4)-(7). $\square$

Theorem 1 also provides an interesting support for a practice, observed in the context of energy system operators, of solving operational planning problems with only one scenario of demand instead of modeling the entire distribution. As discussed in Dias-Garcia et al. (2024), there are reasons for such a choice. System operators then compute pointwise forecasts of the demand and add a heuristic bias to it. Theorem 1 shows that, as long as the *optimal* bias is added, the practice is actually correct, in the sense that the final result is the same as though the full distribution were used. Indeed, the numerical results in Dias-Garcia et al. (2024)—using a method that actually motivated the development of the present paper—show that the use of a carefully computed bias in that setting yields very satisfactory results compared to practical benchmarks. Our results in Section 6 also corroborate that idea, now with the theoretical support provided by Theorem 1 as well as the results in Sections 3 and 4.

## 3  Optimal solution mappings

As discussed in Section 1, our goal is to derive approximations to the function that maps a contextual information $x$ to an optimal solution of (2). We start by formalizing the notion of optimal solution mappings discussed in the introduction. Let $z_S^* : \mathbb{R}^s \mapsto \mathbb{R}^n$ be a mapping defined such that $z_S^*(x)$ is an optimal solution of (2) for $x \in \mathbb{R}^s$ (we assume that (2) does indeed have an optimal solution for any $x \in \mathbb{R}^s$). Such a definition conveys the fact that, in practice, the decision maker will likely not be interested in solving the problem for a particular value of $x$; rather, the decision maker would like to have a *policy* that yields (or, more realistically, approximates) the optimal solution of (2) for any given $x$. In that sense, a policy $\pi$ is a mapping $\pi : \mathbb{R}^s \mapsto \mathbb{R}^n$ such that $\pi(x)$ is feasible (i.e., $\pi(x) \in Z$ for any $x \in \mathbb{R}^s$). Such a policy, of course, must be constructed from available data.

A natural question that arises then is, how to evaluate one such policy? One way to do so is by computing the *out-of-sample* performance of that policy. That is given a dataset $(x_1, \xi_1), \ldots, (x_t, \xi_t)$, we use part of the dataset as training data to construct the policy $\pi$, and then use the remaining dataset (the testing data)

to evaluate out-of-sample performance. This is a standard procedure but it illustrates the need for policies rather than seeking for just an optimal solution for a given $x$. In fact, if the contextual information of interest $x$ does not appear in the testing dataset, or if there are only a handful of observations with that $x$, we cannot really evaluate the objective function of (2). Such an issue of course is also present in the training data, but in that case building approximations of the conditional distribution of $\xi$ with respect to $x$ constitutes one way to generate a policy, as seen in Section 1. In the out-of-sample evaluation we cannot resort to such approximations, otherwise we would be distorting the actual value of the objective function of (2).

The above discussion suggests that what we are really interested in is the average performance of a policy $\pi$ over the set of features, i.e.,

$$\mathcal{P}(\pi) := \int_{\mathcal{X}} \mathbb{E}_{\xi}\left[G(\pi(x), \xi) \mid X = x\right] F_X(dx), \tag{26}$$

where $F_X$ is the distribution of the features $X$. Note that we can rewrite (26) as

$$\begin{aligned} \mathcal{P}(\pi) &= \mathbb{E}_X\left[\mathbb{E}_{\xi}\left[G(\pi(X), \xi) \mid X\right]\right] \\ &= \mathbb{E}_{X,\xi}\left[G(\pi(X), \xi)\right]. \end{aligned} \tag{27}$$

The search for the best policy can then be formulated as

$$\min_{\pi \in \Pi} \mathcal{P}(\pi), \tag{28}$$

where $\Pi$ is the set of mappings $\mathcal{X} \mapsto Z$. The notion of a policy in the above context is discussed in Sadana et al. (2024).

An example of a mapping $\pi \in \Pi$ is given by optimal solutions of (1), i.e., the solution of the problem that ignores the contextual information:

$$\pi_{NF}(x) \in \underset{z \in Z}{\operatorname{argmin}} \ \mathbb{E}_{\xi}\left[G(z, \xi)\right] \quad \forall x \in \mathcal{X}, \tag{29}$$

where the subscript $NF$ stands for "no features". As $\pi_{NF}(\cdot) \in \Pi$, we have that

$$\mathbb{E}_{X,\xi}\left[G(\pi_{NF}(X), \xi)\right] \geq \min_{\pi \in \Pi} \mathbb{E}_{X,\xi}\left[G(\pi(X), \xi)\right],$$

which conveys the intuitive notion that ignoring the contextual information may lead to sub-optimal solutions.

Another example, of course, is given by optimal solutions of (2):

$$\pi_S(x) \in \underset{z \in Z}{\operatorname{argmin}} \ \mathbb{E}_{\xi}\left[G(z, \xi) \mid X = x\right] \quad \forall x \in \mathcal{X}. \tag{30}$$

Proposition 1 below shows that $\pi_S$ in fact characterizes the optimal solutions of (28). Although the result is mentioned in Sadana et al. (2024) as a consequence of an interchangeability property from Rockafellar and Wets (1998), we present a proof here to make the paper self-contained and to add some intermediate steps to that argument.

**Proposition 1.** *Suppose Assumptions 1-2 hold. Then, the mapping $\pi_S$ defined in (30) solves (28). Moreover, if $\pi^*$ solves (28), then $\pi^*(x) \in \underset{z \in Z}{\operatorname{argmin}} \ \mathbb{E}_{\xi}\left[G(z, \xi) \mid X = x\right]$ for all $x \in \mathcal{X}$ except perhaps on a set of $F_X$-measure zero.*

*Proof.* Consider the function $f : \mathbb{R}^n \times \mathbb{R}^s \mapsto \mathbb{R}$ defined as

$$f(z,x) := \mathbb{E}_\xi \left[ G(z,\xi) \,|\, X = x \right].$$

Then, from (30) we have that $\pi_S(x) \in \underset{z \in Z}{\mathrm{argmin}} \ f(z,x)$. Moreover, we have that

$$
\begin{aligned}
\min_{\pi \in \Pi} \mathcal{P}(\pi) &= \min_{\pi \in \Pi} \mathbb{E}_{X,\xi} \left[ G(\pi(X),\xi) \right] \\
&= \min_{\pi \in \Pi} \int_{\mathcal{X}} \mathbb{E}_\xi \left[ G(\pi(x),\xi) \,|\, X = x \right] F_X(dx) \\
&= \min_{\pi \in \Pi} \int_{\mathcal{X}} f(\pi(x),x) \, F_X(dx) \quad\quad\quad\quad\quad\quad\quad (31) \\
&= \int_{\mathcal{X}} \min_{z \in Z} f(z,x) \, F_X(dx) \quad\quad\quad\quad\quad\quad\quad (32) \\
&= \int_{\mathcal{X}} f(\pi_S(x),x) \, F_X(dx) \\
&= \int_{\mathcal{X}} \left( \mathbb{E}_\xi \left[ G(\pi_S(x),\xi) \,|\, X = x \right] \right) F_X(dx) \\
&= \mathbb{E}_{X,\xi} \left[ G(\pi_S(X),\xi) \right] = \mathcal{P}(\pi_S),
\end{aligned}
$$

where the equality in (32) follows from Theorem 14.60 in Rockafellar and Wets (1998) . Thus, $\pi_S$ solves (28). The converse statement also follows from the same Theorem 14.60, which ensures—under a finiteness assumption that holds under Assumptions 1-2 —that any $\pi$ that solves (31) must coincide with the mapping defined by an optimal solution of the inner problem in (32) for $F_X$-almost all $x \in \mathcal{X}$ . $\qquad \square$

The result in Proposition 1 is intuitive—the mapping that yields an optimal solution for each $x$ also yields an optimal solution on the average, and vice-versa. Approximating that mapping, however, is a difficult task. By using Theorem 1, however, we can obtain a stronger result in the case of two-stage problems of the form (4)-(7) under an additional assumption stated below.

**Assumption 3.** *The one-scenario problem $\min_{z \in Z} f(z) + Q(z,\xi)$ has a unique optimal solution for each value of $\xi \in \mathbb{R}^m$.*

As discussed earlier, the assumption holds for example when $f$ is strictly convex. When $f$ is linear, the assumption can be enforced by adding a regularization term, or by perturbing the coefficients $c$ and $q$ as discussed in Dias-Garcia et al. (2024).

**Proposition 2.** *Consider the mapping $\pi_S$ defined in (30), and suppose that $G(z,\xi) = f(z) + Q(z,\xi)$ with $Q$ defined in (5)-(7). Also, suppose Assumptions 1-3 hold. For each $x \in \mathcal{X}$, define $\xi_x^*$ as an optimal scenario to (2)—the existence of which is ensured by Theorem 1. Then, by defining $\pi_D(x)$ as the unique optimal solution of one-scenario problem $\min_{z \in Z} f(z) + Q(z,\xi_x^*)$ we have that the mapping $\pi_D$ is the unique solution to (28), except perhaps on a set of $F_X$-measure zero.*

*Proof.* The proof follows the same steps as in the proof of Proposition 1, noting that under the assumptions of the proposition we have that $\pi_D(x) = \pi_S(x)$ by virtue of Theorem 1. $\qquad \square$

Proposition 2 has an important consequence: since $\pi_D(x)$ can be obtained simply by solving (8)-(10) with $\xi_x^*$ in place of $\xi^*$, to approximate the optimal mapping $\pi^*$ that solves (28) it suffices to approximate an optimal scenario $\xi_x^*$ for each $x$. This leads to the notion of "optimal pointwise forecasts", a term that appears in the title of the paper. We discuss this topic in detail in the next section.

# 4 Optimal pointwise forecasts

Proposition 2 demonstrates that, when $G(z, \xi) = f(z) + Q(z, \xi)$ with $Q$ defined in (5)-(7)—which we will assume throughout this section—the search for policies that approximately solve (28) can be reduced to the search of policies that approximate the mapping $\pi_D(\cdot)$. One way to derive approximations to $\pi_D(x)$ is by approximating an optimal scenario $\xi_x^*$ with another function of $x$. To see this, notice that Assumptions 1-3 imply that the function $z_D^*(\cdot)$ defined as

$$z_D^*(\xi) := \underset{z \in Z}{\operatorname{argmin}} \ f(z) + Q(z, \xi) \tag{33}$$

is *continuous*; see, e.g., Corollaries 8.1 and 9.1 in Hogan (1973). Since $\pi_D(x) = z_D^*(\xi_x^*)$, it follows that if $\psi(x)$ is a function such that $\psi(x) \approx \xi_x^*$, then $\widetilde{\pi}_D(x) := z_D^*(\psi(x))$ yields an approximation to $\pi_D(x)$.

In light of the above discussion, we define the following data-driven approximation to $\xi_x^*$, constructed from observations $(x_1, \xi_1), \ldots, (x_N, \xi_N)$:

$$\psi(x) := \Psi(\theta_N^*, x), \tag{34}$$

where $\Psi(\theta, x)$ is a *pointwise forecast function* of $\xi$ as a function of $x$ parameterized by $\theta$, and $\theta_N^*$ solves the bi-level problem

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} \left| G(z_D^*(\hat{\xi}_n), \xi_n) - G(z_D^*(\xi_n), \xi_n) \right| \tag{35}$$

$$\text{s.t.} \quad \hat{\xi}_n = \Psi(\theta, x_n), \quad n = 1, \ldots, N \tag{36}$$

$$z_D^*(\hat{\xi}_n) = \underset{z \in Z}{\operatorname{argmin}} \ G(z, \hat{\xi}_n), \quad n = 1, \ldots, N \tag{37}$$

$$z_D^*(\xi_n) = \underset{z \in Z}{\operatorname{argmin}} \ G(z, \xi_n), \quad n = 1, \ldots, N. \tag{38}$$

A few remarks about Model (35)-(38) are in order. As discussed in Section 1, there is a growing body of literature on forecast models that are tailored to the optimization problem where such forecast is used. Model (35)-(38) falls in that category, as it measures the error between the forecast $\hat{\xi}_n$ and the observed data $\xi_n$ in terms of the cost of using the respective optimal solutions. More specifically, given the scenario realization $\xi_n$, $z_D^*(\xi_n)$ given by (38) would have been the best possible decision for that scenario, thereby realizing a cost of $G(z_D^*(\xi_n), \xi_n)$. On the other hand, since we only have a forecast $\hat{\xi}_n = \Psi(\theta, x_n)$ of $\xi_n$, we compute the decision $z_D^*(\hat{\xi}_n)$ given by (37), which after the realization of $\xi_n$ incurs a cost of $G(z_D^*(\hat{\xi}_n), \xi_n)$. Thus, we want to bias the forecast function $\Psi(\theta, \cdot)$ so that it minimizes the *post-hoc* total forecast regret $\frac{1}{N} \sum_{n=1}^{N} |G(z_D^*(\hat{\xi}_n), \xi_n) - G(z_D^*(\xi_n), \xi_n)|$.

Note that, by definition of $z_D^*$, we must have $G(z_D^*(\hat{\xi}_n), \xi_n) \geq G(z_D^*(\xi_n), \xi_n)$ so we can remove the absolute value in (35). Moreover, it is clear that $G(z_D^*(\xi_n), \xi_n)$ is constant for the optimization problem in $\theta$. It follows that (35)-(38) can be equivalently written as follows, as in Dias-Garcia et al. (2024) Muñoz, Pineda, and Morales (2022):

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} G(z_D^*(\hat{\xi}_n), \xi_n) \tag{39}$$

$$\text{s.t.} \quad \hat{\xi}_n = \Psi(\theta, x_n), \quad n = 1, \ldots, N \tag{40}$$

$$z_D^*(\hat{\xi}_n) = \underset{z \in Z}{\operatorname{argmin}} \ G(z, \hat{\xi}_n), \quad n = 1, \ldots, N. \tag{41}$$

The importance of Model (39)-(41) lies in Theorems 2 and 3 below, which show that the model yields in the limit the *best possible policy* among those based on forecasts parameterized by $\theta$. Theorem 2 is shown in Dias-Garcia et al. (2024) for the case where $G(z, \xi) = c^\top z + Q(z, \xi)$, by using a similar argument to—but with weaker assumptions than—that used by Shapiro and Xu (2008) for more general optimization problems with equilibrium constraints. A closer look at that proof shows that the main requirements are the continuity of the function $z_D^*(\cdot)$—which follows from Assumptions 1-3, as discussed above—and integrability of $G(z_D^*(\hat{\xi}_n), \xi_n)$ as a function of $\theta$, which is ensured by conditions (iv)-(vi) of the theorem. It follows that the result can be easily extended to the case in (8)-(10) where $f$ is a convex function and $Z$ is a convex set; we present the theorem here for completeness and to state the result in our notation.

**Theorem 2.** *Consider Model* (39)-(41)*. Suppose that (i) Assumptions 1-3 hold, (ii) the forecasting function $\Psi(\cdot, \cdot)$ is continuous in both arguments, (iii) the data process $(X_1, \xi_1), \ldots, (X_N, \xi_N)$ is independent and identically distributed (i.i.d.), (iv) the random variable $\xi$ is integrable, (v) the feasibility set $Z$ is bounded, and (vi) the set $\Theta$ is compact and non-empty. Then, with probability 1,*

$$\lim_{N \to \infty} d(\theta_N^*, S^*) = 0, \tag{42}$$

*where $d$ is the Euclidean distance from a point to a set and $S^*$ is defined as*

$$S^* = \operatorname*{argmin}_{\theta \in \Theta} \ \mathcal{P}(\widetilde{\pi}_D^\theta), \tag{43}$$

*with*

$$\widetilde{\pi}_D^\theta(x) \ := \ z_D^*(\Psi(\theta, x)), \tag{44}$$

$\mathcal{P}$ *defined in* (27)*, and $z_D^*$ defined in* (33)*.*

Note that the assumption that the data process $(X_1, \xi_1), \ldots, (X_N, \xi_N)$ is i.i.d. is a common assumption in the literature. Also, in Dias-Garcia et al. (2024) the result is extended to the case where $X_n$ is a (measurable) function of $\xi_1, \ldots, \xi_{n-1}$, and the data process generating $\{\xi_n\}_{n=1}^\infty$ is a *stationary ergodic* time series. Such a situation covers the case where the contextual information actually consists of previous observations, or some function thereof.

Theorem 2 assumes that the forecasting function $\Psi(\cdot, \cdot)$ is continuous in both arguments. Such an assumption covers many cases of interest, such as when the forecast value is an affine function of the contextual information (as in regression), or more generally when $\Psi$ is given by a neural network built upon continuous functions such as ReLu.

For some methods, however, continuity does not hold; this is the case for example of classification trees, or more generally of classification and regression trees (CART). When the continuity assumption does not hold, convergence can still be achieved as long as the parameter $\theta$ takes on only finitely many values. In that case, Theorem 3 below provides a reasonable alternative to Theorem 2.

**Theorem 3.** *Consider Model* (39)-(41)*. Suppose that (i) Assumptions 1-3 hold, (ii) the data process $(X_1, \xi_1), \ldots, (X_N, \xi_N)$ is independent and identically distributed (i.i.d.), (iv) the random variable $\xi$ is integrable, (v) the feasibility set $Z$ is bounded, and (vi) the set $\Theta$ is* finite *and non-empty. Then, the event*

$$\theta_N^* \in S^* \tag{45}$$

*happens w.p.1 for N large enough, where $S^*$ is defined in* (43)*. If in addition, the support $\Xi$ of $\xi$ is bounded, then the convergence occurs* exponentially fast*, in the sense that there exist positive constants $K$ and $\beta$ such*

*that*

$$P\left(\theta_N^* \notin S^*\right) \ \leq \ Ke^{-N\beta}. \tag{46}$$

*Proof.* For the first claim, it suffices to show that the conditions of Proposition 2.1 in Kleywegt, Shapiro, and Homem-de-Mello (2002) hold in this case. As in that paper, Model (39)-(41) is a discrete stochastic optimization problem with finite feasibility set $\Theta$. Moreover, as in the proof of Theorem 2, under the assumptions of the theorem the random variable $G(z_D^*(\Psi(\theta, X_n), \xi_n))$ is integrable for all $\theta$. Proposition 2.1 in Kleywegt et al. (2002) then ensures that (45) holds w.p.1 for $N$ large enough.

The second claim follows from the fact that, under the assumption on boundedness of $\Xi$, the random variable $G(z, \xi)$ is bounded for all $z \in Z$ and hence Proposition 2.2 in Kleywegt et al. (2002) ensures that (46) holds. Note that the assumption on boundedness of $\Xi$ can be relaxed to finiteness of a certain moment generating function in a neighborhood of zero; we refer to Kleywegt et al. (2002) for details. $\square$

The finiteness assumption on $\Theta$ actually holds when standard CART is used as a forecast method, as long as the contextual information $X$ takes on only *finitely many* values. Indeed, in that case the forecast function $\Psi(\theta, x)$ yields the average of the observations $\xi_n$ in each leaf of the tree specified by $\theta$. Such specification consists of the branching order of the components of $x$, together with the threshold associated with each branching. When the contextual information $X$ takes on only finitely many values, it is clear that the set of possible thresholds can be reduced to the set of values taken by $X$. Thus, in that case the set of possible trees —and hence the set of possible values of $\theta$—is finite. The finiteness property can also be seen from the mixed integer programming formulations for CART in Bertsimas and Dunn (2017) and Verwer and Zhang (2019), where it is shown that the branching order is modeled with a finite number of binary variables that depends on the depth of the tree, which is fixed *a priori* as a parameter of the method; the latter paper also shows that the thresholds can be modeled with finitely many binary variables, the number of which depend on the maximum number of distinct values for each feature.

## 4.1 Approximating the optimal forecast

The results in Theorems 2 and 3 show that the policy $\widetilde{\pi}_D^{\theta_N^*}$ defined in (44) converges to the best possible policy obtained with the forecast function $\Psi$. It remains to study how far the policies $\widetilde{\pi}_D^\theta$ are from the policy that solves (28). Since $\widetilde{\pi}_D^\theta(x) = z_D^*(\Psi(\theta, x))$, we see from Proposition 2 that the answer to that question lies in how well $\Psi(\theta_N^*, x)$ approximates an "optimal scenario" $\xi_x^*$. This is summarized in Theorem 4 below, which is proved for the case where the first-stage problem in (4) is linear:

**Theorem 4.** *Consider the case where the first-stage problem in* (4) *is linear, i.e.,* $f(z) = c^\top z$ *and* $Z$ *is polyhedral. Suppose that there exist* $\delta \geq 0$ *and* $\widehat{\theta} \in \Theta$ *such that* $\|\Psi(\widehat{\theta}, x) - \xi_x^*\| \leq \delta$ *for all* $x \in \mathcal{X}$. *Then, under the assumptions of either Theorem 2 or Theorem 3, there exists a constant* $K \geq 0$—*which depends only on the parameters that define the function* $G$—*such that the policy* $\widetilde{\pi}_D^{\theta_N^*}$ *obtained from Model* (39)-(41) *satisfies*

$$\lim_{N \to \infty} \mathcal{P}\left(\widetilde{\pi}_D^{\theta_N^*}\right) - \min_{\pi \in \Pi} \mathcal{P}\left(\pi\right) \ \leq \ K\delta. \tag{47}$$

*Proof.* Under conditions of the theorem, the function $z_D^*$ defined in (33) (which we write here as $z_D^*(h, T)$) is

the optimal solution of the linear program

$$\min_{z \in Z} \ c^\top z + q^\top y$$

$$\text{s.t.} \quad Tz + Wy = h$$

$$y \geq 0.$$

It follows from (Robinson, 1973, Corollary 3.1) that $z_D^*(h, T)$ is a Lipschitz function of $(h, T)$ with constant, say, $M_1$. Thus, by the assumption in the theorem there exist $\delta \geq 0$ and $\widehat{\theta}$ such that $\|\Psi(\widehat{\theta}, x) - \xi_x^*\| \leq \delta$ for all $x \in \mathcal{X}$, which implies that

$$\left\| \widetilde{\pi}_D^{\widehat{\theta}}(x) - \pi_D(x) \right\| = \left\| z_D^*(\Psi(\widehat{\theta}, x)) - z_D^*(\xi_x^*) \right\| \leq M_1 \delta. \tag{48}$$

Next, note that we have, for sufficiently large $N$,

$$\mathcal{P}\big(\widetilde{\pi}_D^{\theta_N^*}\big) - \min_{\pi \in \Pi} \mathcal{P}\big(\pi\big) = \mathcal{P}\big(\widetilde{\pi}_D^{\theta_N^*}\big) - \mathcal{P}\big(\pi_D\big)$$

$$\leq \mathcal{P}\big(\widetilde{\pi}_D^{\widehat{\theta}}\big) - \mathcal{P}\big(\pi_D\big). \tag{49}$$

The inequality in (49) follows from the fact that under the assumptions of either Theorem 2 or Theorem 3 the sequence $\{\theta_N^*\}$ approaches the optimal set $S^*$ and so for $N$ large enough we have that $\mathcal{P}\big(\widetilde{\pi}_D^{\theta_N^*}\big) \leq \mathcal{P}\big(\widetilde{\pi}_D^{\widehat{\theta}}\big)$. Moreover, from (17) we have that the subdifferential set of $G(z, \xi)$ is bounded for all $z \in Z$ and $\xi \in \Xi$ which in turn implies that $G(\cdot, \xi)$ is uniformly Lipschitz, i.e., there exists a constant $M_2 \geq 0$ such that

$$\big|G(z_1, \xi) - G(z_2, \xi)\big| \leq M_2 \|z_1 - z_2\| \quad \text{for all } z_1, z_2 \in Z \text{ and all } \xi \in \Xi. \tag{50}$$

Inequalities (48) and (50), together with (49) and definition (27) of $\mathcal{P}$, then imply that

$$\mathcal{P}\big(\widetilde{\pi}_D^{\theta_N^*}\big) - \min_{\pi \in \Pi} \mathcal{P}\big(\pi\big) \leq K\delta,$$

where $K := M_1 M_2 \delta$. $\hspace{10cm} \square$

Theorem 4 materializes the notion of "optimal pointwise forecasts" alluded to in the title of the paper: as long as an optimal scenario $\xi_x^*$ (viewed as a function of $x$) can be approximated uniformly by some function $\Psi(\theta, \cdot)$, a *pointwise* forecast constructed with the parameters $\theta_N^*$ resulting from Model (39)-(41) will suffice, in the sense that the policy $\widetilde{\pi}_D^{\theta_N^*}(x)$ defined as $z_D^*(\Psi(\theta_N^*, x))$ will yield an approximate solution to (28). As indicated by (47), the quality of the latter approximation depends on how well $\Psi(\theta, \cdot)$ approximates $\xi_{(\cdot)}^*$. Note that, while the form of $\Psi$ must be specified in advance, the actual values of $\theta$ that make $\Psi(\theta, \cdot)$ approximate $\xi_{(\cdot)}^*$ need not be known in advance—in fact, they result from applying Model (39)-(41) as stated in Theorem 4. Thus, the more functions the set $\{\Psi(\theta, \cdot) : \theta \in \Theta\}$ contains, the better the approximation of the policy that solves (28).

Naturally, in order to be able to approximate $\xi_{(\cdot)}^*$ we need this function to have some properties. One such property is described in Proposition 3 below.

**Proposition 3.** *Suppose that $\mathcal{X}$ is a continuous set and that the conditional distribution of $\xi|x$ [2] is close to the distribution of $\xi|x'$ when $x$ is close to $x'$. More precisely, suppose that given $\varepsilon > 0$, there exists $\delta > 0$*

---

[2]Here we abuse the notation—by "the conditional distribution of $\xi|x$" we mean the conditional probability measure defined as $P_x(A) := P(\xi \in A \mid X = x)$ for Borel sets $A$.

*such that*

$$\|x - x'\| < \delta \implies d_W(\xi|x, \xi|x') < \varepsilon, \tag{51}$$

*where $d_W$ denotes the Wasserstein distance between two distributions. Then, if the mapping $\pi_S(\cdot)$ defined in (30) is single-valued for each $x$, then $\pi_S(\cdot)$ is continuous and hence $\xi_{(\cdot)}^*$ constructed in Theorem 1 is continuous.*

*Proof.* The results follows from classical stability results for optimal solutions of stochastic programs. For instance, Corollary 14 in Römisch (2003) shows that, in the case of the two-stage stochastic program (4)-(7), there exists $\overline{\delta} > 0$ such that, if $P$ and $Q$ are two distributions such that $d_W(P, Q) < \overline{\delta}$, then we have that

$$Z^*(Q) \subseteq Z^*(P) + \varphi(L\, d_W(P, Q))\, \mathbb{B}$$

where $Z^*(Q)$ and $Z^*(P)$ are the optimal solution sets of (4)-(7) when the distribution of $\xi$ is respectively $Q$ and $P$, $\varphi(\cdot)$ is a certain increasing function that vanishes at zero, $L$ is a positive constant, and $\mathbb{B}$ is the Euclidean unit ball. Thus, if $Z^*(P)$ and $Z^*(Q)$ are singletons, it follows that given $\eta > 0$ sufficiently small, there exists $0 < \varepsilon < \overline{\delta}$ such that $\|Z^*(P) - Z^*(Q)\| < \eta$ whenever $d_W(P, Q) < \varepsilon$. By putting the conditional distributions of $\xi|x$ and $\xi|x'$ in place of $P$ and $Q$, from (51) we see that the condition $d_W(P, Q) < \varepsilon$ holds whenever $\|x - x'\| < \delta$. We conclude that $\pi_S(\cdot)$ is continuous. Since $\xi_x^* = T^*\pi_S(x)$, we see that in that case $\xi_{(\cdot)}^*$ is continuous as well. $\qquad\square$

A particular case where condition (51) in Proposition 3 is satisfied is when the uncertainty $\xi$ can be written as a linear model of $x$, i.e.,

$$\xi = A + Bx + \epsilon,$$

where $A$ is a constant vector, $B$ is a matrix of coefficients, and $\epsilon$ has a multi-variate Normal distribution with mean 0 and covariance matrix $\Sigma$. That is, for fixed $x$ and $x'$ we have that

$$\xi|x \sim \text{Normal}(A + Bx, \Sigma), \quad \xi|x' \sim \text{Normal}(A + Bx', \Sigma)$$

which are close when $x$ is close to $x'$.

## 4.2   Solving the bi-level problem

We discuss now some methods to solve the bi-level problem (39)-(41) which is constructed from observations $(x_1, \xi_1), \ldots, (x_N, \xi_N)$ of the feature $X$ and the random variable $\xi$. Throughout this section, we will assume that the matrix $T$ in (6) is not random, so the random variable $\xi$ corresponds only to the right-hand side term $h$. Recalling that $G(z, \xi)$ is defined as $c^\top z + Q(z, \xi)$ (with $Q$ defined in (5)-(7)), we can write (39)-(41) as

$$\min_{\theta, (z_1, y_1^u), \ldots, (z_N, y_N^u)} \frac{1}{N} \sum_{n=1}^{N} c^\top z_n + q^\top y_n^u \tag{52}$$

$$\text{s.t.} \quad \theta \in \Theta \tag{53}$$

$$z_n \in Z, \ y_n^u \geq 0, \quad n = 1, \ldots, N \tag{54}$$

$$Tz_n + Wy_n^u = \xi_n, \quad n = 1, \ldots, N \tag{55}$$

$$(z_n, y_n^\ell) = \operatorname*{argmin}_{z \in Z, \ y \geq 0} \left\{ c^\top z + q^\top y \ : \ Tz + Wy = \Psi(\theta, x_n) \right\}, \quad n = 1, \ldots, N. \tag{56}$$

Note that we have different second-stage variables for the upper and lower level problems, denoted respectively $y_n^u$ and $y_n^\ell$, $n = 1, \ldots, N$. Also, since the lower level problems are just linear programs, we can write (52)-(56) as a single level problem using KKT conditions, as customary in the bi-level literature. For instance, assuming for simplicity that the set $Z$ is the positive orthant, we have that (52)-(56) is equivalent to

$$\min_{\theta, \{(z_n, y_n^u, y_n^\ell, \lambda_n)\}_{n=1}^N} \frac{1}{N} \sum_{n=1}^N c^\top z_n + q^\top y_n^u \tag{57}$$

$$\text{s.t.} \quad \theta \in \Theta \tag{58}$$

$$z_n \geq 0, \quad n = 1, \ldots, N \tag{59}$$

$$T z_n + W y_n^u = \xi_n, \ y_n^u \geq 0, \quad n = 1, \ldots, N \tag{60}$$

$$T z_n + W y_n^\ell = \Psi(\theta, x_n), \ y_n^\ell \geq 0, \quad n = 1, \ldots, N \tag{61}$$

$$T^\top \lambda_n \leq c, \quad n = 1, \ldots, N \tag{62}$$

$$W^\top \lambda_n \leq q, \quad n = 1, \ldots, N \tag{63}$$

$$c^\top z_n + q^\top y_n^\ell - \Psi(\theta, x_n)^\top \lambda_n \leq 0, \quad n = 1, \ldots, N. \tag{64}$$

In the above formulation, (58)-(60) are the upper level constraints, (61) represents the lower level primal constraints, (62)-(63) are the lower level dual constraints, and (64) imposes strong duality on the lower level, which is equivalent to writing the complementarity constraints.

Problem (57)-(64) is in general hard to solve, especially when the forecast function $\Psi(\theta, x)$ is a complicated function of $\theta$. A situation where the above model can be solved reasonably efficiently is when (i) the set $\Theta$ is polyhedral, and (ii) the function $\Psi(\theta, x)$ is linear in $\theta$, i.e. $\Psi(\theta, x) = B_x \theta$ for some matrix $B_x$. The latter condition corresponds to an "application-driven regression" whereby, instead of measuring the error with quadratic loss functions as in standard regression, we measure it using the objective function of the problem. This is actually the approach we use in our numerical experiments, as discussed in Section 5. In that case, the problem given by (57)-(64) is a linear program except for (64), which contains the bilinear term $\theta^\top B_{x_n}^\top \lambda_n$. Some alternatives to address that issue include using binary variables to eliminate the bilinear term (or to model the equivalent complementarity constraints), or relaxing (64) by putting it in the objective function. The latter can be solved with a penalty alternating direction method; we refer to Kleinert and Schmidt (2021) for further discussion and comparisons between these approaches. We also remark that some commercial solvers can solve problems with bilinear terms very efficiently.

When $\Theta$ is finite, as in Theorem 3, it may be possible to solve (57)-(64) as a mixed integer program, depending again on the form of $\Psi(\theta, x)$ as a function of $\theta$. Designing efficient algorithms for such a problem is a topic for future research.

## 4.3 The case with functional dependencies

We end this section by noticing that the "optimal scenario" constructed in Theorem 1 is defined according to the constraints of the second-stage problem rather than by the random variables present in those constraints. This is an important distinction, as it implies that dependencies between random variables in different constraints are not necessarily respected by the optimal scenario.

We illustrate this issue with a simple example. Consider the standard newsvendor model, with purchase cost $c_P = 0$, inventory cost $c_I$, and shortage cost $c_S$. We can write the problem as the two-stage problem

$\min_{z \geq 0} \mathbb{E}[Q(z, D)]$, where

$$Q(z, D) = \min_{y^o, y^u} c_I y^o + c_S y^u$$

$$\text{s.t.} \qquad y^o \geq -D + z$$

$$y^u \geq D - z$$

$$y^o, \; y^u \geq 0.$$

In the notation of (6)-(7), we have $y = [y^o, y^u]^\top$, $W = I$, $h = [-D, D]^\top$, and $T = [-1, 1]^\top$. Note that the above model has two constraints with the same random variable ($D$). Note also that in this case the functional dependence between the two components of $h$ (i.e., one is the negative of the other) is maintained, since the optimal scenario is $h^* := Tz^* = [-z^*, z^*]^\top$ and thus $D^* = z^*$, where $z^*$ is the optimal solution of the newsvendor model. Indeed, it is easy to check that the one-scenario problem $\min_{z \geq 0} Q(z, D^*)$ also has optimal solution $z^*$, as stated by Theorem 1.

Suppose now that we add the constraint $z \geq 2D$ to the second stage, so now we have $W = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$, $h = [-D, D, 2D]^\top$, and $T = [-1, 1, 1]^\top$ (to ensure that the complete recourse assumption holds, we can add a slack variable to the constraint $z \geq 2D$ with high cost). Again, let $z^*$ be the optimal solution of the stochastic model. Then, the optimal scenario is $h^* := Tz^* = [-z^*, z^*, z^*]^\top$, so we see that the functional dependence between the second and third components of $h$ (i.e, that the third one is twice the second one) is not respected. However, it is easy to check that solving the one-scenario problem with $h^*$ in place of $h$ will indeed yield the same solution $z^*$ as the stochastic model. Thus, as discussed in Section 4.1, as long as one is able to approximate $h^*$ with the forecast function $\Psi(\cdot, \cdot)$, an approximation of the optimal solution $z^*$ can still be obtained. Later, at the end of Section 6.5—and also in Appendix D—we will comment on the effect of this restriction on one of our case studies.

# 5  Solution Methods

In this section, we present different approaches to find policies $\pi \in \Pi$ to solve the problem (2). We begin by describing application-driven (AD) methods, which consider the structure of the problem and find a function $\Psi(\theta, x)$ that defines the policy $\tilde{\pi}_D^\theta(x) = z_D^*(\Psi(\theta, x))$. Then, we present benchmark methods, where we include standard predict-then-optimize methods and conditional sampling methods.

## 5.1  Application-Driven Forecasts Methods

For AD methods, we consider two $\Psi(\theta, x)$ forecast functions, linear regression and a method based on regression trees known as M5. In both cases, we seek to determine the optimal parameters for each function by solving the bi-level problem (39)-(41). We also propose a heuristic solution method.

As mentioned above, the problem (39)-(41) seeks a parameterization of the function $\Psi(\theta, x)$ such that the $z$ decisions obtained from this prediction function, $\hat{\xi}_n$, $n = 1 \ldots, N$, minimize the total cost obtained by considering observations $\xi_n$. Let $x_n$ be the covariates for observations $\xi_n$. The function $\Psi(\theta, x)$ allows us to obtain a policy $\tilde{\pi}_D^{\hat{\theta}}(x)$, which minimizes the error $G(\tilde{\pi}_D^{\hat{\theta}}(x), \xi_n)$. Let $\hat{\theta}$ be the parameterization found by solving the bi-level problem. The policy is given by,

$$\tilde{\pi}_D^{\hat{\theta}}(x) \in \operatorname*{argmin}_{z \in Z} G(z, \Psi(\hat{\theta}, x)). \tag{65}$$

In general, the bi-level problem (39)-(41) is difficult to solve, including the problems presented in our computational study. For this reason, we consider heuristic methods to solve it, in particular, the implementation of the Meta algorithm and the Nelder-Mead method presented in Dias-Garcia et al. (2024). In this work, this heuristic method yields high-quality solutions in short computational running times. The Meta algorithm is as follows:

---

**Output:** Optimized $\theta$

**1** Initialize $\theta$;

**2 while** *Not converged* **do**

**3**      Update $\theta$;

**4**      **foreach** $n \in \{1, \ldots, N\}$ **do**

**5**          Forecast: $\hat{\xi}_n \leftarrow \Psi(\theta, x_n)$;

**6**          Plan Policy: $z_n^* \leftarrow \underset{z \in Z}{\operatorname{argmin}}\, G(z, \hat{\xi}_n)$;

**7**          Cost Assessment: $cost_n \leftarrow G(z_n^*, \xi_n)$;

**8**      **end**

**9**      Compute cost: $cost(\theta) \leftarrow \sum_{n=1}^{N} cost_n$;

**10 end**

**Algorithm 1:** Meta algorithm.

---

This algorithm starts by initializing the vector $\theta \in \Theta$, which can be simply initialized with zero values. As long as the algorithm does not converge, the variables are updated. The convergence criterion is that the objective function decreases less than $\epsilon = 10^{-7}$ between two consecutive iterations. The update for $\theta$ aims to minimize the objective function, for which we need an optimization method. As in Dias-Garcia et al. (2024), we propose to use a derivative-free method, the Nelder-Mead approach. For each $n = 1, \ldots, N$, the forecast for observation $(x_n, \xi_n)$ is found using the current $\theta$ vector and the covariate vector $x_n$. Then, for the prediction $\hat{\xi}_n$, a plan $z_n$ is obtained. The cost of this plan is calculated using the observation $n$ value, $\xi_n$. Finally, the total cost for the current $\theta$, $cost(\theta)$, is calculated.

### 5.1.1 Linear Regression Prediction

For a linear regression, the parameterization of the $\Psi(\theta, x)$ function is given by,

$$\Psi(\theta, x) = \theta_0 + \sum_{i=1}^{s} \theta_i x_i. \tag{66}$$

The resulting policy from the linear function under the Application-Driven approach is denoted as $\tilde{\pi}_N^{AD}(x)$.

### 5.1.2 M5 Prediction

The M5 model is presented in Quinlan (1992). It is a generalization to classification and regression tree models (CART) that seeks to improve the prediction made in each region, using linear regression instead of an average value.

The CART method considers a partition of the feature space into $R$ regions, where for each partition $\ell \in \{1, \ldots, R\}$ a constant value is determined as a prediction (Hastie et al., 2009; Murphy, 2012). The partition is carried out by branching a decision tree and each region $\ell \in \{1, \ldots, R\}$ is a leaf of the tree. Tree structures allow to fit nonlinear functions, obtaining high-quality predictions. In general, the method for defining these partitions is heuristic; when a node is branched, the splitting variable $x_j$, $j = 1, \ldots, s$, and the

cutoff value $r$ that has the greatest impact on reducing the variance of the training data are identified. This splitting variable and point are defined so that two half-planes are obtained:

$$\mathcal{R}_1(j,r) = \{n = 1, \ldots, N \mid x_n \leq r\} \text{ and } \mathcal{R}_2(j,r) = \{n = 1, \ldots, N \mid x_n > r\}. \tag{67}$$

We then seek the splitting variable $j$ and the split point $r$ that minimize

$$\min_{j,r} \left\{ \min_{\psi_1} \sum_{n \in \mathcal{R}_1(j,r)} (\xi_n - \psi_1)^2 + \min_{\psi_2} \sum_{n \in \mathcal{R}_2(j,r)} (\xi_n - \psi_2)^2 \right\}. \tag{68}$$

For given $j$ and $r$, the inner minimization is solved by

$$\psi_1 = \frac{1}{|\mathcal{R}_1(j,r)|} \sum_{n \in \mathcal{R}_1(j,r)} \xi_n \text{ and } \psi_2 = \frac{1}{|\mathcal{R}_2(j,s)|} \sum_{n \in \mathcal{R}_2(j,s)} \xi_n. \tag{69}$$

In practice, the splitting variable and the split point are determined by reviewing all the given observations and then determining the best pair $(j,r)$. This guarantees that the regions $\mathcal{R}_1(j,r)$ and $\mathcal{R}_2(j,r)$ are always non-empty.

The branching continues until the tree reaches a given maximum height or the number of data points in each leaf is less than or equal to a given value, or both. The prediction for a new vector of covariates $X$, corresponding to a region $\ell \in \{1, \ldots, R\}$, is the average value of the observations in that region, $\bar{\xi}_\ell$.

The M5 model considers an additional step for prediction given a vector $X$: using the observations in a region, a linear regression is estimated. We propose to perform this linear regression considering an AD approach. That is, once a tree structure is defined, where each region has a set of observations, for this set, we solve the bi-level problem (39)-(41) and consider a linear regression function for $\Psi(\theta, x)$, as in equation (66). For each leaf $\ell \in R$, we obtain a vector $\theta_\ell$, which takes into account the structure of the problem. The training of the tree follows the same heuristic used by CART. The resulting policy from using M5 under the Application-Driven approach is denoted as $\tilde{\pi}_N^{M5+AD}(x)$.

If the decision tree has a maximum tree size, with a fixed partition $\{1, \ldots, R\}$, constructed using i.i.d. observations, then the convergence of each parameter $\theta_{\ell N}$, $\ell \in \{1, \ldots, R\}$, is guaranteed.

**Proposition 4.** *Consider a decision tree with a fixed partition $\{1, \ldots, R\}$, constructed using independent and identically distributed (i.i.d.) $N' < \infty$ observations. Also, consider Model (39)-(41). Suppose that (i) Assumptions 1-3 hold, (ii) the data process $(X_1, \xi_1), \ldots, (X_N, \xi_N)$ is i.i.d., (iii) the random variable $\xi$ is integrable, and (iv) the primal and dual feasibility sets $Z$ and $U$ are bounded. Then, with probability 1,*

$$\lim_{N \to \infty} d(\theta_{\ell N}^*, S_\ell^*) = 0, \tag{70}$$

*where $d$ is the Euclidean distance from a point to a set and $S_\ell^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \ \mathbb{E}_{X,\xi} \left[ G\big(\tilde{\pi}_D^\theta(X), \xi\big) \mid (X, \xi) \in (\mathcal{X}_\ell, \Xi_\ell) \right]$, and $(\mathcal{X}_\ell, \Xi_\ell)$ is the support for region $\ell \in \{1, \ldots, R\}$.*

*Proof.* Let $\ell \in \{1, \ldots, R\}$ be a region of the decision tree. Let $h_\ell(N)$ be the number of elements in region $\ell$, with $N$ training data points. We want to show that $h_\ell(N)$ goes to infinity when $N$ goes to infinity. When the data are i.i.d., by the law of large numbers, we have that $\frac{h_\ell(N)}{N} \to p_\ell$ w.p. 1, where $p_\ell$ is the probability that an observation belongs to that region. We then have 2 options:

1. $p_\ell > 0$: in this case, we have that for $N$ large, $h_\ell(N) \sim Np_\ell$ and hence $h_\ell(N) \to \infty$ w.p. 1.

2. $p_\ell = 0$: in this case, the probability that there are observations in the region $\ell$ is zero, *i.e.* $h_\ell(N) = 0$ for all $N$ w.p. 1. Since the region $\ell$ is not empty (at least one observation from $N'$ training data is in region $\ell$), this cannot happen.

We conclude that the number of elements in the region $\ell$ goes to infinity. Then, we apply Theorem 2, which guarantees convergence of the parameter $\theta_{\ell N}^*$. □

## 5.2 Benchmark methods

In this section, we present benchmark methods, which we will use as a solution reference for our AD methods in our computational study. We start by describing predict-then-optimize methods (PO), commonly used in practice, and then conditional distribution methods (CD).

### 5.2.1 Predict then Optimize Framework

Predict-then-optimize (PO) methods consider two stages to find a policy. In the first stage, the $\theta \in \Theta$ parameters are found for the $\Psi(\theta, x)$ function, $x \in \mathcal{X}$, so that an error metric takes its smallest value. This metric aims to measure the difference between the observed value and the prediction of the function $\Psi(\theta, x)$. In general, for observations $(x_1, \xi_1), \ldots, (x_N, \xi_N)$, the quadratic error, $\sum_{n=1}^{N} \|\xi_n - \Psi(\theta, x_n)\|^2$, is used as the error metric. Once the parameter vector $\hat{\theta}$ is determined, the policy $\pi \in \Pi$ is defined according to Equation (65).

For these PO methods we consider a linear function (Equation (66)) and a decision tree method. The tree method we use is CART, which, unlike M5, partitions the feature space and then makes predictions as averages of the observations. CART manages to achieve a better approximation of the data function, but it is not an AD method. In fact, to define the CART parameters (structure of the tree), the objective is to minimize the quadratic error. Similarly to M5, CART is usually constructed by means of a greedy heuristic (Hastie et al., 2009).

The resulting policy from the linear function is denoted as $\tilde{\pi}_N^{LS}(x)$ and the resulting policy from CART, $\tilde{\pi}_N^{CART}(x)$, $x \in \mathcal{X}$.

### 5.2.2 Conditional Distribution Methods

Conditional distribution methods (CD) seek to approximate the conditional expectation $\mathbb{E}_\xi[G(z, \xi)|X = x]$, $x \in \mathcal{X}$, in Equation (30), using a set of scenarios for each $x$. These methods also require two stages; in a first stage the scenarios for each $x$ must be determined, a set $\mathcal{N}(x)$, and then, in the second stage, the following problem is solved to find the conditional distribution policy $\tilde{\pi}(x)$:

$$\tilde{\pi}(x) \in \underset{z \in Z}{\operatorname{argmin}} \ \frac{1}{|\mathcal{N}(x)|} \sum_{n \in \mathcal{N}(x)} G(z, \xi_n), \qquad \forall x \in \mathcal{X}. \tag{71}$$

Note that here we are assuming that each scenario has the same weight (probability) in the objective function.

The simplest of these conditional methods ignores the context $x \in \mathcal{X}$, and the problem is simply solved by considering the observations $(x_1, \xi_1), \ldots, (x_N, \xi_N)$ given. For this, the policy is determined by solving these scenarios as done using the sample average approximation approach (SAA). We denote this approach as the SAA policy. Note that this method can also be seen as an approximation of the $\pi^{NF}$ policy of Equation (29).

Given a set of observations, the policy obtained $\tilde{\pi}_N^{SAA}(x)$, $x \in \mathcal{X}$, is given by:

$$\tilde{\pi}_N^{SAA}(x) \in \underset{z \in Z}{\text{argmin}} \; \frac{1}{N} \sum_{n=1}^{N} G(z, \xi_n), \qquad \forall x \in \mathcal{X}. \tag{72}$$

Other approaches to generate policies based on conditional scenarios are based on machine learning techniques. Using contextual information $x \in \mathcal{X}$ and predictive methods, scenarios for $x$ are generated. Then, a solution is found by solving an approximation problem as in (71). In Bertsimas and Kallus (2020), several prediction methods are proposed to determine these scenarios, such as $k$-nearest-neighbors (KNN), local linear regression (LOESS), CART and random forests (RF). As a benchmark for our computational experiments, we consider the KNN method, since in other computational studies it has also been a benchmark approach, showing good performance (Kannan, Bayraksan, and Luedtke, 2022). The KNN method determines the $k$ nearest neighbors for an $x \in \mathcal{X}$, $\mathcal{N}(x) = \left\{ n = 1, \dots, N \mid \sum_{j=1}^{N} \mathbb{I}[\|x - x_i\| \geq \|x - x_j\|] \leq k \right\}$, where $\|\cdot\|$ is a distance metric (*e.g.*, Euclidean distance). The resulting policy $\tilde{\pi}_N^{KNN}(x)$, is given by:

$$\tilde{\pi}_N^{KNN}(x) \in \underset{z \in Z}{\text{argmin}} \; \frac{1}{|\mathcal{N}(x)|} \sum_{n \in \mathcal{N}(x)} G(z, \xi_n). \tag{73}$$

The last CD method that we present is the one proposed in Kannan et al. (2022). In this method, we start by estimating $\theta$ by minimizing the squared error of the $\Psi(\theta, x)$ function and the observations (as for the linear regression), $x \in \mathcal{X}$. Then, a point prediction is obtained for each $x \in \mathcal{X}$. Using $\Psi(\hat{\theta}, x)$, an error $\varepsilon_n := \xi_n - \Psi(\hat{\theta}, x_n)$ is calculated, $n = 1, \dots, N$. The conditional distribution (30) can be approximated, obtaining a $\tilde{\pi}_N^{ER-SAA}(x)$ policy given by:

$$\tilde{\pi}_N^{ER-SAA}(x) \in \underset{z \in Z}{\text{argmin}} \; \frac{1}{N} \sum_{n=1}^{N} G\left(z, \Psi(\hat{\theta}, x) + \varepsilon_n\right), \qquad x \in \mathcal{X}. \tag{74}$$

# 6 Computational experiments

In this section, we present some numerical experiments to support our theoretical results. The goal of these experiments is to provide a "proof of concept" for the idea introduced in this paper that solving stochastic optimization problems with fixed recourse and fixed costs can be accomplished by solving a one-scenario problem, where this single scenario is generated as an "optimal" pointwise forecast in the sense of Theorems 2 and 3. Thus, we do not attempt to demonstrate that our method is superior to other methods in the literature; rather, we want to show that this is a new alternative approach to solve stochastic optimization problems with contextual information that deserves further investigation.

We apply our methodology to three problems from the literature: a resource allocation one, a shipment planning problem, and also a bike-sharing reallocation problem with real-world data. We test the methods described in Section 5 and show their performance on each of the three problems.

## 6.1 Problem 1: two-stage resource allocation

We first consider the two-stage resource allocation problem used in Kannan et al. (2022). There is a set $I$ of resources and a set $J$ of clients. The first-stage decision is the quantity $z_i \geq 0$ of resource $i$ ordered, for each $i$ in $I$. The uncertain parameter is the demand $\xi_j$ of each client $j$ in $J$, and once it is realized, we can observe it and take two second-stage decisions: the amount $y_{i,j}^s \geq 0$ of resource $i$ assigned to client $j$, and the

amount $y_j^c \geq 0$ of unmet demand of client $j$. Other parameters, assumed to be known, are the unitary cost $c_i$ of resource $i$, the unitary cost of unmet demand $q_j$ by client $j$, the yield $\rho_i$ of resource $i$, and the service rate $\mu_{i,j}$ of the allocation of resource $i$ to client $j$. In this way, the problem can be formulated as follows:

$$\min_{z \in \mathbb{R}_+^I} \sum_{i \in I} c_i z_i + \mathbb{E}\left[Q(z, \xi)\right] \tag{75}$$

where the expectation is taken with respect to the random vector $\xi$ of demands of the clients, and where the second-stage cost function $Q(z, \xi)$ is

$$Q(z, \ \xi) := \min_{y^s, \, y^c} \quad \sum_{j \in J} q_j y_j^c \tag{76}$$

$$\text{s.t.} \quad \sum_{j \in J} y_{i,j}^s \leq \rho_i z_i \qquad \qquad \forall \text{ resource } i \in I \tag{77}$$

$$\sum_{i \in I} \mu_{i,j}\, y_{i,j}^s + y_j^c \geq \xi_j \qquad \qquad \forall \text{ client } j \in J \tag{78}$$

$$y^s \in \mathbb{R}_+^{I \times J}, \ y^c \in \mathbb{R}_+^I \tag{79}$$

In Section 6.4 we consider instances with $|I| = 20$ resources and $|J| = 30$ clients.

## 6.2  Problem 2: two-stage shipment planning

We also consider a two-stage shipment planning problem that appears in Bertsimas and Kallus (2020). It consists of a set $I$ of warehouses and a set $J$ of locations. In the first stage, we decide the amount $z_i \geq 0$ to be produced and stored at each warehouse $i$ in $I$, at a cost of $c > 0$ per unit. Then, the demand $\xi_j$ at location $j$ in $J$ is realized, and based on its values we make two second-stage decisions: the last-minute production of $y_i^w$ units at warehouse $i$, incurring a higher unit production cost of $r > c$, and the amount $y_{i,j}^s \geq 0$ of product to be shipped from warehouse $i$ to location $j$, at a unit shipment cost of $s_{i,j}$. The problem is formulated as follows:

$$\min_{z \in \mathbb{R}_+^I} \quad c \sum_{i \in I} z_i + \mathbb{E}\left[Q(z, \xi)\right] \tag{80}$$

where, again, the expectation is taken with respect to the random vector $\xi$ of demands of the clients, and where the second-stage cost function $Q(z, \xi)$ is now

$$Q(z, \xi) := \min_{y^w, \, y^s} \quad r \sum_{i \in I} y_i^w + \sum_{i \in I} \sum_{j \in J} s_{i,j} y_{i,j}^s \tag{81}$$

$$\text{s.t.} \quad \sum_{i \in I} y_{i,j}^s \geq \xi_j \qquad \qquad \forall \text{ location } j \in J \tag{82}$$

$$\sum_{j \in J} y_{i,j}^s \leq z_i + y_i^w \qquad \qquad \forall \text{ warehouse } i \in I \tag{83}$$

$$y^w \in \mathbb{R}_+^I, \ y^s \in \mathbb{R}_+^{I \times J} \tag{84}$$

In Section 6.4 we consider instances with $|I| = 5$ warehouses and $|J| = 12$ locations.

## 6.3 Data generation

To generate the data for both problems described in Sections 6.1 and 6.2, we follow the method in Section 4 of Kannan et al. (2022) to model the dependency between the clients' or locations' demands $\xi = (\xi_j, j \in J)$ in $\mathbb{R}^J$, as a function of a vector of covariates $x = (x_l, l \in L)$, for some finite set $L$. Specifically, we assume that for all $j$ in $J$, the demand $\xi_j$ is determined by the equation

$$\xi_j = a_j + \sum_{l \in L} b_{j,l} \cdot (x_l)^p + \epsilon_j, \tag{85}$$

where $p > 0$ is a fixed *degree*, that we will later take in $\{0.5, 1, 2\}$; and $\epsilon_j \sim N(0, \sigma_j^2)$ for all $j$ in $J$ and are independent across $J$. Finally, $(a_j, j \in J)$, $(b_{j,l}, (j,l) \in J \times L)$ and $(\sigma_j, j \in J)$ are additional model parameters used in Section 4 of Kannan et al. (2022); see Appendix B for further details.

In Section 6.4 we consider instances with $|L| = 3$ covariates, and the covariate vectors $x$ in $\mathbb{R}^L$ are modeled as i.i.d. draws from a multivariate folded/half-normal distributions, as in Section 4 of Kannan et al. (2022). That is, we take $x_l = |\tilde{x}_l|$ for all $l$ in $L$, where the vector $\tilde{x}$ in $\mathbb{R}^L$ has a multivariate normal distribution with zero mean and with covariance matrix generated randomly from a beta(2, 2) distribution and then rescaled to $[-1, 1]$, see Annex G of Kannan et al. (2022).

## 6.4 Results for Problems 1 and 2

In this section we show the results of applying the methods discussed in Section 5 to the problems in Sections 6.1 and 6.2 using the data generation procedure in Section 6.3.

We perform the experiments for the values $p \in \{0.5, 1, 2\}$ for the degree $p$ of the data generation procedure, and train the methods in Section 5 with $N$ samples of the pair $(x, \xi)$, for $N \in \{10^2, 10^3, 10^4\}$. Once the training is done, we test the performance of each method by approximating its corresponding optimality gap using the estimation procedure in Mak, Morton, and Wood (1999) (see Algorithm 1 in Kannan et al. (2022) ), which can be summarized as follows.

1. Generate a sample of the covariate $x$.

2. Generate 1,000 samples of the demand $\xi$ conditional on the covariate value $x$, i.e., from the conditional distribution of $\xi|x$.

3. Compute the solution $z(x)$ of the method in Section 5 being tested, and compute the average cost of the solution $z(x)$ on the latter 1,000 samples.

4. Solve the problem $\min_{z \in Z} \mathbb{E}\left[G(z, \xi) \mid X = x\right]$ where the expected value is approximated using the 1,000 samples of the demand $\xi$.

5. Compute the gap between the value obtained in step 3. minus the one from step 4.

6. Repeat 30 times steps 2. to 5., compute a confidence interval for the gap, and express it as a percentage of the value in step 3.

By using the above algorithm we obtain a normalized estimate of 99%-confidence upper bound for the covariate $x$, say $\widehat{B}_{99}(x)$. Since the data-driven solutions depend on the realization of the covariate sample $x$, we repeat 30 times this procedure —generating 30 covariates $x^1, \ldots, x^{30}$ and their corresponding upper bounds $\widehat{B}_{99}(x^1), \ldots, \widehat{B}_{99}(x^{30})$— and report the results using box plots of the latter 30 upper bounds.

**Resource allocation problem.** In Figure 1 we show the results for the resource allocation problem in Section 6.1 with $|I| = 20$ resources, $|J| = 30$ clients and $|L| = 3$ covariates.

Our findings across the model degrees $p \in \{0.5, 1, 2\}$ are as follows. For all values of $p$, we see that the pure SAA approach, which ignores the contextual information, has poor performance. This is expected when there is some correlation between $\xi$ and $x$, as the knowledge of $x$ improves the estimation of $\xi$.

When the model degree is $p = 0.5$ pr $p = 1$, methods AD, M5+AD, and ER-SAA show very good performance, even when the training data size is small. As more information becomes available, the kNN method also improves its performance. We can see that the predict-then-optimize policies CART and LS perform much worse than the other policies (except SAA), although the difference decreases as $N$ increases.

When the model degree is $p = 2$, we see a somewhat different behavior. The optimality gaps' variability increases due to the nonlinear form of the data. We also see that policies AD and ER-SAA end up performing worse (for larger $N$) than CART and similar to LS. The best performing methods are kNN and M5+AD, which take advantage of partitioning the data space.

Figure 5 in Appendix C shows a zoomed-in version of Figure 1 without CART, SAA and LS so as to emphasize the differences among the remaining four policies.
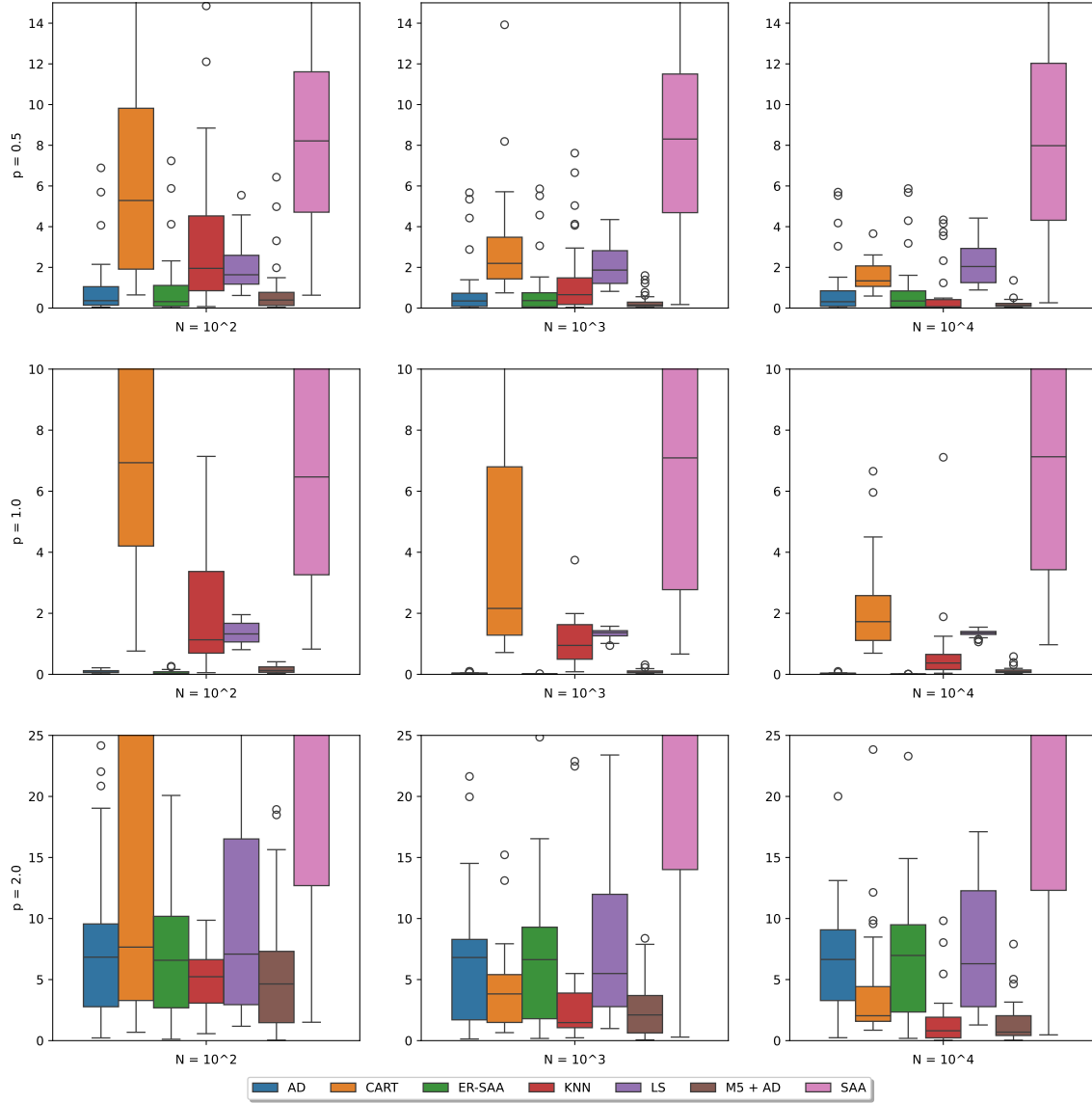
Figure 1: Comparison of methods proposed in Section 5 for the Problem 1 of resource allocation in Section 6.1. $p$ is the degree of the data generation procedure in (85), and $N$ is the number of samples with which each method is trained.

**Shipment planning problem.** In Figure 2 we show the results for the shipment planning problem in Section 6.2 with $|I| = 5$ warehouses, $|J| = 12$ locations and $|L| = 3$ covariates. The conclusions are very similar to the previous problem, except that in this case the predict-then-optimize policies and the SAA policy perform much worse than AD policies and the conditional expectation ones even when $p = 2$. As before, Figure 6 in Appendix C shows a zoomed-in version of Figure 2 without CART, SAA and LS so as to emphasize the differences among the remaining four policies. We also see that, as in the previous problem, the methods that use an "optimal" pointwise forecast demonstrate in most cases similar or superior performance, compared to methods that consider multiple scenarios. In particular, the M5+AD method almost never loses to any of its competitors, regardless of the size of the training dataset, and has the best performance of all when $p = 2$. This supports our theoretical results, showing that a two-stage stochastic program can actually be solved with only one scenario.
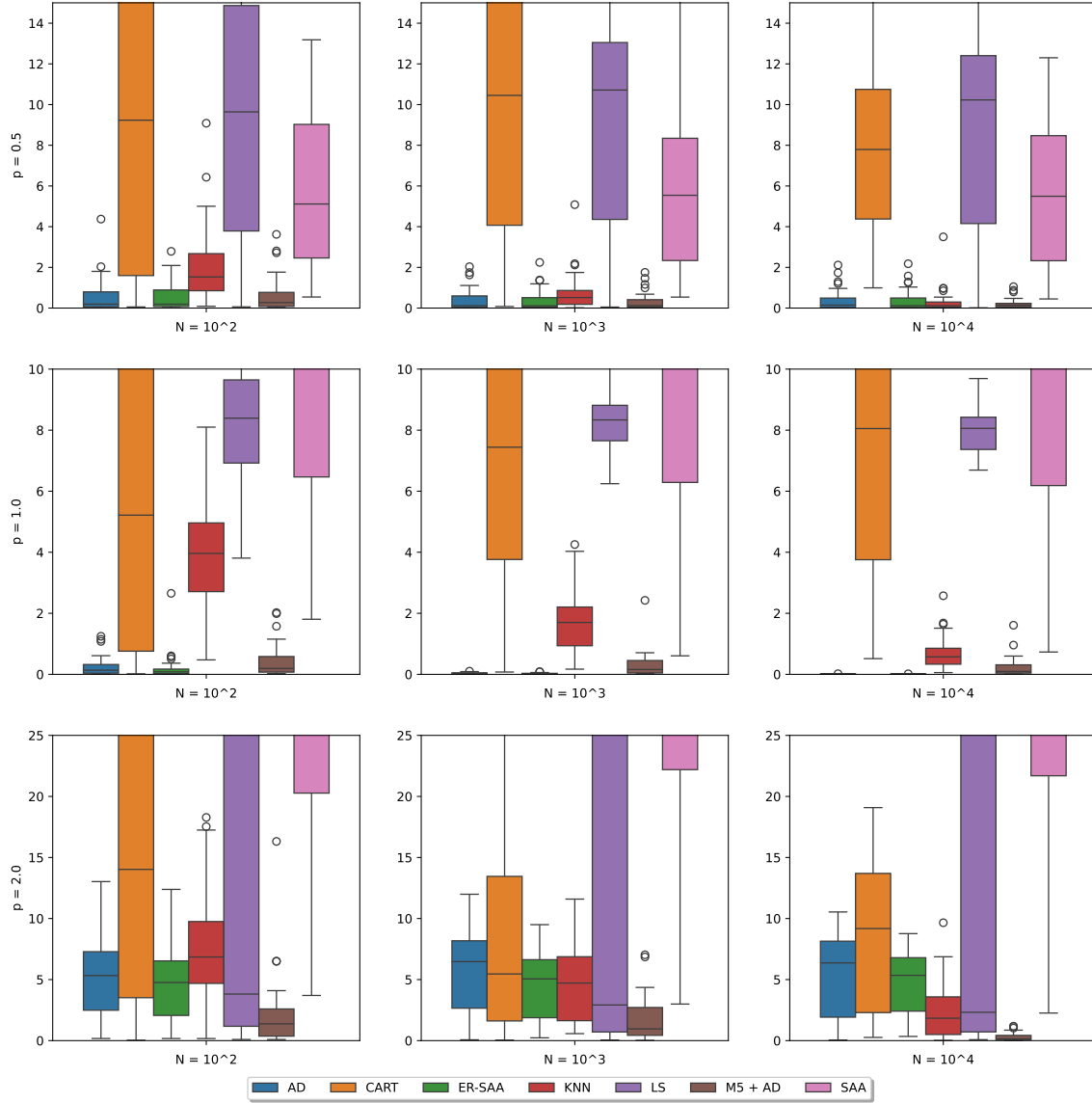
Figure 2: Comparison of methods proposed in Section 5 for Problem 2 of shipment planning in Section 6.2. $p$ is the degree of the data generation procedure in (85), and $N$ is the number of samples with which each method is trained.

## 6.5   Problem 3: bike sharing reallocation with real-world data

To test our methodology in a realistic problem with real-world data, we study the bike reallocation problem in Cavagnini (2019) that uses real-world data from the San Francisco, CA, bike-sharing system. The problem uses open-source data available in the website `https://www.kaggle.com/datasets/benhamner/sf-bay-area-bike-share/data` that describes a system with 350 bikes and 34 stations; see the distribution of the stations in Figure 3. It comprises four datasets containing several data fields, including station status, weather conditions (including temperature, wind speed, and humidity) and trip information. It covers the period from August 2013 to August 2015.

Cavagnini (2019) proposes a two-stage stochastic model to address this bike-sharing problem in San Francisco. Even though their model did not consider features or covariate information, their formulation is
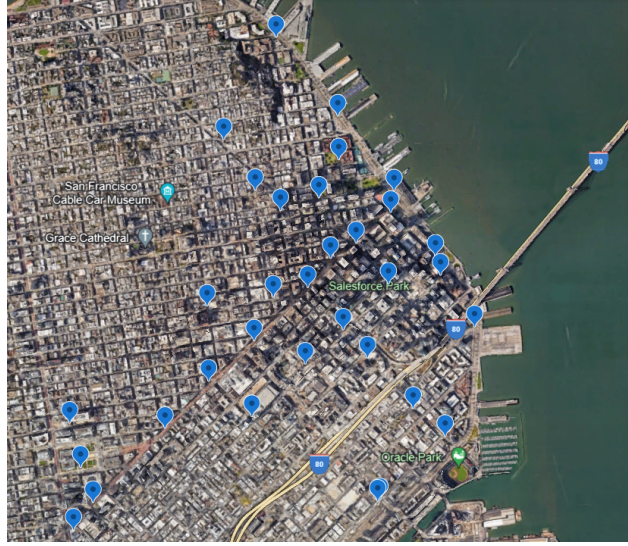
Figure 3: San Francisco's stations of bike sharing system

suitable for the framework of this paper of data-driven stochastic programming with covariates' information. Importantly, though, their formulation considers integer variables, however, to be able to apply our methods, we relax them and consider only continuous variables.

| *Sets* | |
|---|---|
| $\mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$ | set of stations, where the depot is station $|\mathcal{I}|$ |
| *Parameters* | |
| $\underline{x}_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Minimum number of bikes to allocate in station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ |
| $I_{|\mathcal{I}|0}$ | Depot capacity |
| $Q_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Maximum capacity of station $i$ |
| $I_{i0}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Initial availability of bikes at station $i$ |
| $C$ | Maximum capacity of the relocation truck |
| $p_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | bikes' stock-out penalty at station $i$ |
| $c_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Excess penalty at station $i$ |
| $\frac{c_i}{Q_i}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Penalty associated to extra bikes placed at station $i$ after the re-balancing period |
| $f_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Allocation cost at station $i$ |
| $t_{i,i+1}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Re-balancing cost to allocate bikes from station $i$ to $i+1$ |
| $\xi_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Net demand of station $i$ |
| *Variables* | |
| $x_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Amount of bikes to allocate at station $i$ in the first stage |
| $y_{i,i+1}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Amount of bikes to distribute from station $i$ to $i+1$ in the second stage |
| $I_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Inventory or balance of bikes at station $i$ |
| $I_i^+, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Surplus of bikes at station $i$ |
| $I_i^-, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Number of stock-out bikes at station $i$ |
| $B_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Extra bikes balance at station $i$ during the re-balancing period |
| $B_i^+, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Balance of of extra bikes at station $i$ during the re-balancing period |
| $E_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Excess inventory balance of bikes at station $i$ during the re-balancing period |
| $E_i^+, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Excess of bikes at station $i$ during the re-balancing period |

Table 1: Variables, parameters and sets for Problem 3 in Section 6.5.

**Problem formulation.** With a slight abuse of notation, there is a set $\mathcal{I} = \{1, 2, \ldots, |\mathcal{I}|\}$ of stations, where station $|\mathcal{I}|$ is the bike depot. The problem starts at the beginning of the day, where, before knowing

the demand $\xi_i$ of bikes at each station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$, we have to decide the number $x_i$ of bikes to deliver from the depot to the station, incurring a unitary delivery cost of $f_i$. Nonetheless, later we will assume that the decision-maker can predict the demand of each station by using covariate information. Each station $i$ has a capacity of $Q_i$ bikes, and to ensure that a station can satisfy the demand, it requires a minimum number $\underline{x}_i$ of bikes to be allocated at the beginning of the day. Later, during the day, the demand of each station $\xi_i$ is realized and at the end of the day the service provider re-balances the bikes having to re-distribute $y_{i,i+1}$ bikes from station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ to the next station on the fixed route, that without loss of generality, we assume to be station $i+1$, incurring a moving cost of $t_{i,i+1}$. Also, we assume that the route begins and ends at the bike depot. The bikes are reallocated using a truck with a total capacity of $C$ bikes. In this problem, the shared bike provider seeks to prevent situations where a user intends to return a bike to a station that is already full or wants to rent a bike from an empty station. The decision maker also aims to minimize the number of bikes reallocated to prevent bike damage. To archive this, Cavagnini proposes using a *starvation* and a *congestion* term. The starvation is measured by the variable $I_i^-$ and a *stock-out penalty* $p_i$, for each station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$. The congestion is measured with two extra terms at each station $i$ in $\mathcal{I} \setminus \{|\mathcal{I}|\}$: the *extra inventory* term $B_i^+ \geq 0$ measures the number of bikes beyond the number initially allocated, and the *excess inventory* term $E_i^+ \geq 0$ measures the number of bikes over the station's capacity $Q_i$. Any non-negative value of these is penalized with a unit cost of $c_i$ and $c_i/Q_i$, respectively.

With this, the problem formulation is as follows:

$$\min_x \quad \sum_{i \in \mathcal{I}} f_i x_i + \mathbb{E}\left[Q(x, \xi)\right] \tag{86}$$

$$\text{s.t.} \quad x_i \geq \underline{x}_i \qquad\qquad\qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{87}$$

$$\bar{I}_{i0} + x_i \leq Q_i \qquad\qquad\qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{88}$$

$$\sum_{i \in \mathcal{I} \setminus \{|\mathcal{I}|\}} x_i \leq \bar{I}_{|\mathcal{I}|0} \tag{89}$$

$$x_i \geq 0 \qquad\qquad\qquad\qquad \forall i \in \mathcal{I} \tag{90}$$

where the second-stage cost function $Q(x, \xi)$ is

$$Q(x, \xi) = \min \sum_{i \in \mathcal{I} \setminus \{|\mathcal{I}|\}} \left( t_{i,i+1} y_{i,i+1} + \frac{c_i}{Q_i} B_i^+ + c_i E_i^+ + p_i(-I_i^-) \right) \tag{91}$$

$$\text{s.t.} \quad y_{i,i+1} \leq C \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{92}$$

$$I_{|\mathcal{I}|} = \bar{I}_{|\mathcal{I}|0} - \sum_{i \in \mathcal{I} \setminus \{|\mathcal{I}|\}} x_i + y_{|\mathcal{I}|-1,|\mathcal{I}|} \tag{93}$$

$$I_{|\mathcal{I}|} \leq \bar{I}_{|\mathcal{I}|0} \tag{94}$$

$$I_1 = \bar{I}_{|\mathcal{I}|0} + x_1 - \xi_1 - y_{1,2} \tag{95}$$

$$I_i = \bar{I}_{i0} + x_i - \xi_i + y_{i-1,i} - y_{i,i+1} \qquad \forall i \in \mathcal{I} \setminus \{1, |\mathcal{I}|\} \tag{96}$$

$$I_i^+ = \max\{0, I_i\} \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{97}$$

$$I_i^- = \min\{0, I_i\} \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{98}$$

$$E_i = I_i^+ - Q_i \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{99}$$

$$E_i^+ = \max\{0, E_i\} \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{100}$$

$$B_i = I_i^+ - x_i - \bar{I}_{i0} - E_i^+ \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{101}$$

$$B_i^+ = \max\{0, B_i\} \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{102}$$

$$I_i, B_i, E_i \in \mathbb{R} \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{103}$$

$$y_{i,i+1}, I_i^+, B_i^+, E_i^+ \geq 0 \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{104}$$

$$I_i^- \leq 0 \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{105}$$

where the constraints (97), (98), (100) and (102) are easily linearized using additional decision variables that bound the *max* and *min* terms.

We describe now the components of the above model, following closely the discussion in Cavagnini (2019). The objective functions (86) and (91) minimize the total expected penalty, obtained by summing over all the penalties for delivery, re-balancing, extra and excess inventory, and stock-out. The constraints (87) require that the delivered quantity to each station has to be at least the initial requirement. Additionally, constraints (88) guarantee that the sum of the quantity allocated and the initial availability at each station does not exceed the station's capacity. Constraints (89) imply that the total number of allocated bikes within stations is less than the available quantity at the depot. Constraints (92) ensure that the number of bikes transported by the vehicle during re-balancing never exceeds its capacity. Constraints (93) guarantee that the quantity of bikes at the depot at the end of each scenario equals the initial bike availability plus the amount received from the last visited station, minus the quantities delivered to stations. Additionally, constraints (94) ensure that the number of bikes at the depot at the end of the re-balancing period does not exceed its capacity. Furthermore, the "flow balance" constraints for bikes at the first station on the route differ from the remaining stations. Specifically, constraints (95) ensure that for the first visited station, the amount of bikes at the end of the re-balancing period equals the sum of the initially available quantity and the quantity received from the depot, minus the amounts used to satisfy the demand and those bikes that are redistributed to subsequent stations on the route. Similarly, constraints (96) determine the inventory position at a station other than the first, as a function of the initial inventory level, the number allocated, the number withdrawn/returned, and the number redistributed to another station. Constraints (97) and (98) determine the surplus and stock-out quantities for each station. Considering the presence of a valet service, wherein bikes can be returned to full stations, constraints (99) and (100) calculate the number of bikes at each station that exceeds station capacity. Constraints (101) and (102) determine when more bikes are positioned at a

station after re-balancing than were initially allocated, but not more than station capacity.

## 6.6 Results

We now show the results of applying the methods in Section 5 for the previous two-stage problem. The uncertain parameter is the daily demand for bikes at each station, and our methodology uses additional covariate information to obtain a forecast of the demand. In our experiments, we use six features: one indicator (values 0 or 1) showing if it is a business day or not, an integer variable that indicates the number of the week, the month of the year, the year, the precipitation inches, and the wind direction. Also, the dataset contains approximately 730 data points when grouping the demands and features daily for the prediction and optimization procedure.

To estimate the optimality gap of each method we compute its out-of-sample performance by randomly splitting the dataset in a training and test set, keeping a standard proportion of 80:20 of training and test size, and we take the *out-of-sample cost* as the average of the bike-sharing problem cost over the testing data. Since the solutions' are conditioned to the particular realization of the training and test sets, we replicate this procedure by obtaining 10 training and test sets keeping the proportions mentioned.
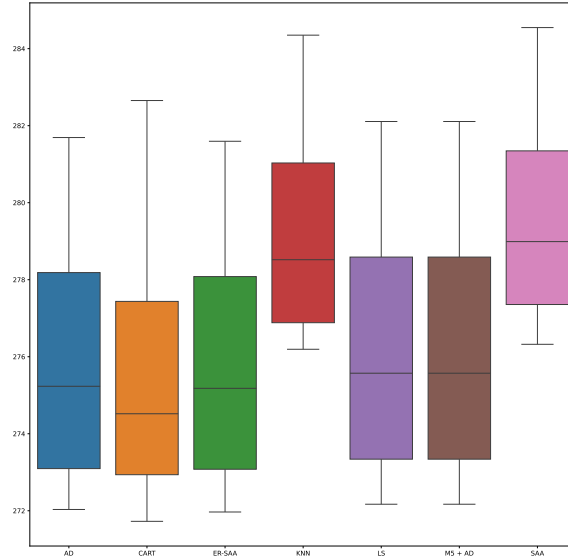


Figure 4: Average out-of-sample costs for 10 different train and test sets for methods proposed in Section 5.

Figure 4 shows the out-of-sample costs for all the methods and the proposed multiple replications procedure. We see that all the methods except KNN and SAA perform similarly, while KNN and SAA are clearly worse in this case. The similarity among the five best methods could be explained by the fact that, as seen in the previous experiments, some of the methods require more data to perform well, but in this case the data set is rather small (730 records). As before, the inferior performance of SAA can be attributed to the fact that such an approach does not take advantage of the contextual information. Similarly to our previous experiments, this example demonstrates that methods that use a properly chosen pointwise forecast such as AD and M5+AD show similar or superior performance compared to methods that consider multiple scenarios.

**Remark:** Cavagnini (2019) proposes the two-stage bike relocation model presented in (86)-(105) that assumes static demand, but in the published version of that work (Cavagnini, Maggioni, Bertazzi, and Hewitt, 2024) the authors attempt to address that limitation by proposing a small model variation that requires estimating

additional random quantities such as the maximum number of consecutive bikes that are returned to a station before being withdrawn. Unfortunately, the limited size of the dataset does not allow for an accurate estimation of those quantities (68, i.e. twice the number of stations); in fact, Cavagnini et al. (2024) generate random samples from the empirical distribution. As our goal in this example is to truly estimate the random quantities directly from the data—by using contextual information—one possible way to circumvent the problem is to define such quantities as fractions of the demand in each node. While such an approach bypasses the need for more data—since it only requires estimating the demand, as in the original model—it leads to functional dependencies among the right-hand side of different constraints, which as discussed in Section 4.3 is not respected by the one-scenario approach. In Appendix D we present the results, where we see that indeed the AD and M5+AD approaches perform worse than other methods. In fact, in that case SAA is among the best performers, thereby suggesting that the contextual information is of little use in such a model. In summary, to properly solve the model in Cavagnini et al. (2024) using contextual information it would be necessary to have a larger dataset that would allow us to estimate all required quantities directly from the data.

# 7   Conclusions

Stochastic optimization problems are defined in terms of the (possibly unknown) distributions of the underlying random variables. Accordingly, methods to solve such problems typically estimate the input distributions and then apply some some scenario generation/reduction technique, perhaps combined with an approach that allows for the decomposition of the problem across scenarios. In the setting of optimization problems with contextual information, the estimation task becomes more difficult as it requires the estimation of conditional distributions for any given value of the contextual information. On the other hand, *end-to-end learning* techniques proposed in the literature have proven very valuable in the contextual information setting by combining the estimation and optimization steps. Still, those methods typically do not estimate entire distributions—or do so in a limited way—so their use for classical two-stage stochastic programs may be restricted.

In this paper we have considered an alternative application-driven approach whereby only *pointwise estimates* are required, when the problem to be solved belongs to the class of two-stage stochastic programs with fixed recourse and fixed costs. The basis for the proposed approach is a novel result that shows that, for that class of problems, it suffices to use *one* scenario, in the sense that solving the problem with that single scenario yields the same solution as the original problem. Our integrated learning and optimization method uses problem information to determine the best parametric approximation of that (unknown) scenario.

The main goal of this work is to show that our novel pointwise approach provides a practical alternative way to solve two-stage problems with contextual information, which by-passes the need for estimating distributions. Our numerical results corroborate that idea. There is of course much room for improvements, especially regarding the development of specialized techniques to solve the bilevel models that are part of the method, and the use of other machine learning methods such as neural networks within our setting. We hope our work will stimulate further research on these topics. We also hope that our one-scenario result can spur new research on alternative methods for stochastic optimization problems that searches over the space of scenarios rather than over the space of solutions.

# Appendices

## A   Estimation of the stations' daily demand

To calculate the net demand for each station $d_i$, we need to forecast the daily demand for the entire system. This is based on the features $X$ that provide information on the daily weather in San Francisco. Subsequently, an adjacency matrix $P_{ij}$ is computed to estimate the proportion of daily trips between stations. The demand for each station is determined by subtracting the number of bikes returned $d_i^r$ from the number of bikes withdrawn $d_i^w$ at each station, following the procedure proposed by Cavaginini. To estimate the daily demand for stations using different methods, we first calculate the overall demand using the methods outlined in Section 5. Then, we calculate the estimated number of bikes withdrawn and returned to each station by multiplying the adjacency matrix by the forecast obtained from the methods proposed. Finally, the net demand is obtained by subtracting the estimated number of withdrawn and returned bikes (i.e. $d_i = d_i^w - d_i^r$). The data that supports the experiments conducted within problem in Section 6 and Appendix D is available at: `https://www.kaggle.com/datasets/benhamner/sf-bay-area-bike-share`.

**Data treatment and forecasting.**   The dataset provides four datasets where information on the trips, theater, and stations' status and information are given. The trips dataset contains information on all the trips within stations, indicating the station and time of beginning and end of each trip. The trip data was grouped to present daily trips for the whole system. The weather dataset contains daily information about weather conditions such as temperature, wind speed, and others. The dataset comprises continuous and categorical variables, however, a vectorization of categorical variables is proposed to address this issue. A binary variable is created to indicate whether a particular day is a holiday. Using station data, a variable has been created to indicate the available docks within stations for each day. Lastly, the station data is used to estimate optimization model parameters such as the capacity of each station. In regards to forecasting, after conducting feature engineering and a feature importance procedure, we selected 6 features to predict the demand for the methods outlined in Section 5. Based on these features, the forecast aims to predict the number of bikes in the system.

**Adjacent matrix calculation and stations' estimated net demand.**   We used the trip data to compute the adjacency matrix $P_{ij}$. This involved counting the number of bikes that departed from each station $i$ and ended at station $j$, which is represented in the $i, j$ position of the matrix obtaining a matrix $C_{ij}$. It is important to mention that the diagonal of the matrix represents the trips that start and end in the same station. To obtain the proportion of the trips we divide the matrix $C_{ij}$ with the total amount of trips of the system in the whole dataset ($P_{ij} := C_{ij} / \sum_{k,l} C_{kl}$). Finally, given the proportion of trips and the daily system forecast for each method, $\hat{y}_{sys}^t$, we obtain the estimated trips within a day of operation by simply multiplying the estimated daily demand by the adjacency matrix ($\hat{y}_{sys}^t P_{ij}$). Finally, given the estimate of the withdrawn and returned bikes for each day and each method, we simply compute the estimated net demand $d_i$ by subtracting the estimated number of withdrawn and returned (i.e. $d_i^w - d_i^r$ ), which are computed by summing the rows and columns of each station respectively, obtaining the net demand for each station $d_i$.

# B   Data generation details

The parameters $(a_j,\, j \in J)$, $(b_{j,l},\, (j,l) \in J \times L)$ and $(\sigma_j,\, j \in J)$ in the data generation procedure (85) are taken in Kannan et al. (2022) as follows:

$$a_j = 50 + 5\delta_{j,0} \tag{106}$$

$$b_{j,1} = 10 + \delta_{j,1} \tag{107}$$

$$bj, 2 = 5 + \delta_{j,2} \tag{108}$$

$$bj, 3 = 2 + \delta_{j,2} \tag{109}$$

where $\{\delta_{j,0}\}_{j \in J}$ are i.i.d. samples from the standard normal distribution, and $\{\delta_{j,1}\}_{j \in J}$, $\{\delta_{j,2}\}_{j \in J}$, $\{\delta_{j,3}\}_{j \in J}$ are i.i.d. samples from a uniform distribution $U(-4, 4)$. Also, $\sigma_j = \sigma = 5$ for all $j$ in $J$.

# C   Zoomed-in versions of Figures 1 and 2

We present below zoomed-in versions of Figures1 and 2 that display only the best four contenders, i.e. we remove the SAA, LS and CART policies.
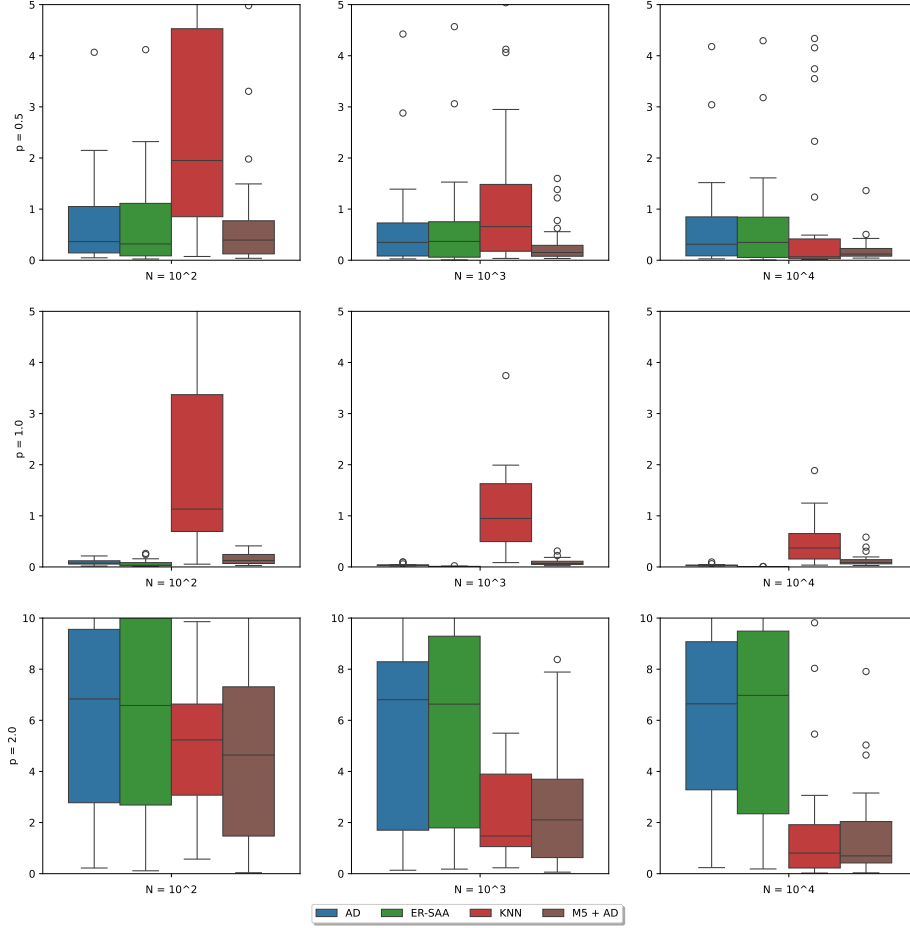
Figure 5: Detailied comparison of methods proposed in Section 5 for the Problem 1 of resource allocation in Section 6.1. $p$ is the degree of the data generation procedure in (85), and $N$ is the number of samples with which each method is trained.

# D   Problem 4: bike sharing rellocation with real data

Cavagnini (2019) proposed the two-stage bike relocation model presented in (86)-(105) that assumed static demand, but Cavagnini et al. (2024) attempted to address this limitation by proposing a small model variation. The proposed model ensures that there are always at least as many bikes available as the maximum number of consecutive bikes that have been withdrawn from a station before being returned. It also ensures that there are always at least as many free docks available as the maximum number of consecutive bikes that have been returned to a station before being withdrawn. This approach helps to prevent congestion and shortages, especially during times of peak consecutive bike withdrawals and returns. To archive that, Cavagnini et al. (2024) introduced extra stochastic parameters $g_i$ and $h_i$ representing the maximum number of consecutive bikes withdrawn from station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ before a return occurs, and the maximum number of consecutive bikes returned station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ before a withdrawal occurs. These parameters are obtained by using Montecarlo sampling using the historical demand, however, in our setting, we considered them as
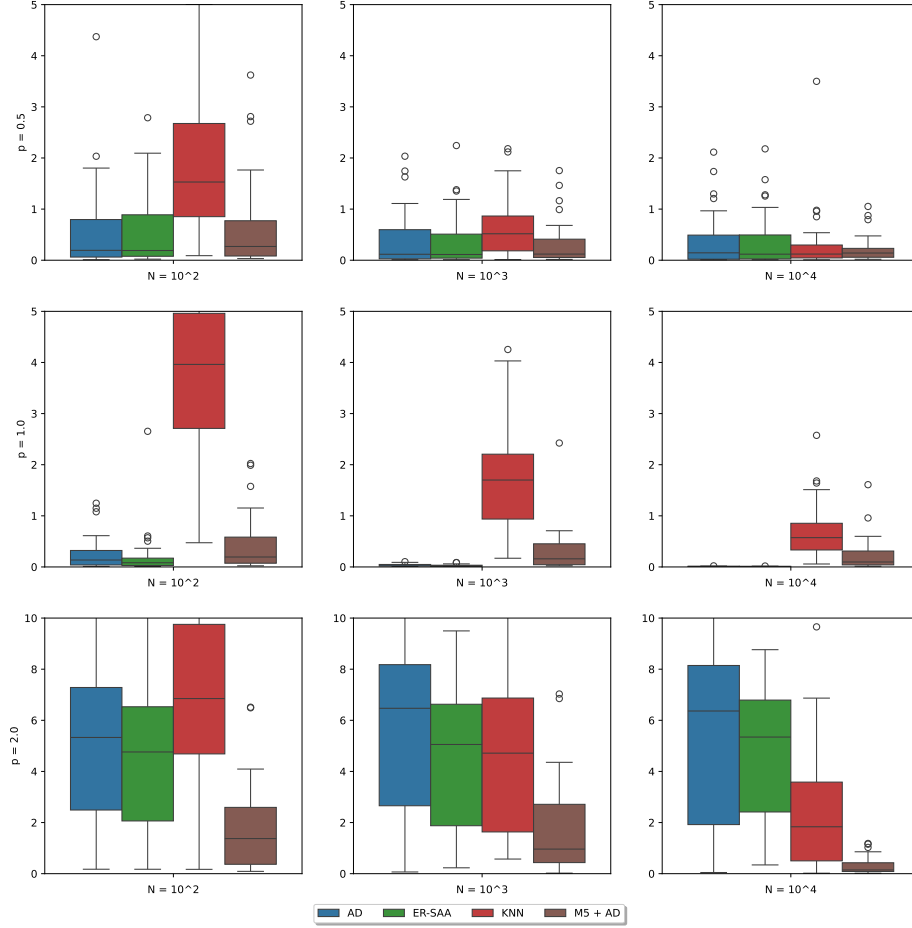
Figure 6: Detailed comparison of methods proposed in Section 5 for Problem 2 of shipment planning in Section 6.2. $p$ is the degree of the data generation procedure in (85), and $N$ is the number of samples with which each method is trained.

fixed parameters due to feasibility, which depends on the estimate of the total number of bikes returned $d_i^r$ and the estimate of the total of bikes withdrawn $d_i^w$ at each station.

**Estimating the new parameters.** To determine the parameters, we calculated the total number of bikes withdrawn, $W_{i,t}$, and returned, $R_{i,t}$, as well as the maximum number of consecutive bikes withdrawn, $w_{i,t}$, and returned, $r_{i,t}$, at each station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ on each day $t \in T$ within the trip dataset. With this data, we calculated $\alpha_{i,t} := w_{i,t}/W_{i,t}$ and $\beta_{i,t} := r_{i,t}/R_{i,t} \quad \forall_{i \in \mathcal{I} \setminus \{|\mathcal{I}|\}, t \in T}$, representing the proportion of the maximum number of consecutive bikes withdrawn and returned over the total withdrawn and returned bikes at each station. Then, we calculate the vectors $\bar{\alpha}_i := \frac{1}{T} \sum_{t \in T} \alpha_{i,t}$ and $\bar{\beta}_i := \frac{1}{T} \sum_{t \in T} \beta i, t$ that represent the expected proportion of maximum number of consecutive bikes withdrawn and returned over the total number of bikes withdrawn and returned. Finally, we estimate $h_i$ and $g_i$ as a proportion of the estimated withdrawn and returned demand, that is, $h_i := \bar{\alpha}_i d_i^w$ and $g_i := \bar{\beta}_i d_i^r$ where $d_i^r$ and $d_i^w$ are obtained by using the forecast

methods of Section 5 and the methodology detailed in Appendix A.

**Problem formulation.** The model proposed by Cavagnini et al. (2024) is similar to the proposed in Cavagnini (2019), however, they introduce some parameters and constraints to address the limitation of the static demand. The problem considers a set $\mathcal{I} = \{1, 2, \ldots, |\mathcal{I}|\}$ of stations with capacity of $Q_i$ bikes, where station $|\mathcal{I}|$ is the bike depot. The bike planning starts at the beginning of the day, where, before knowing the demand $\xi_i$ of bikes at each station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$, we have to allocate a number $x_i$ of bikes to deliver from the depot to the station having a fixed cost of $f_i$ per bike.

Later in the day, the demand for each station $\xi_i$ is realized. At the end of the day, the service provider rebalances the bikes, having to re-distribute $y_{i,i+1}$ bikes from station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ to the next station $i + 1$ on the fixed route, incurring a moving cost of $t_{i,i+1}$. The bikes are redistributed using a truck with a total capacity of $C$ bikes. The bike provider aims to avoid situations where a user wants to return a bike to a full station or needs to rent a bike from an empty station. The decision maker also wants to minimize the number of bikes redistributed. To achieve this, Cavagnini suggests using starvation and a congestion term. The starvation is parametized by the variable $I_i^-$ and a *stock-out penalty* $p_i$, for each station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$. The congestion is measured with two terms at each station $i$ in $\mathcal{I} \setminus \{|\mathcal{I}|\}$: the *extra inventory* term $B_i^+ \geq 0$ measures the number of bikes beyond the number initially allocated, and the *excess inventory* term $E_i^+ \geq 0$ measures the number of bikes over the station's capacity $Q_i$. Any non-negative value of these is penalized with a unit cost of $c_i$ and $c_i/Q_i$, respectively. Finally, the model introduces the stochastic parameters $g_i$ and $h_i$ are used to prompt the model to determine a target inventory level that is higher than the maximum number of consecutive withdrawn bikes, $g_i$, and to ensure that there are at least $h_i$ free docks. To ensure that we can find a feasible solution even if the total number of bikes withdrawn and returned consecutively exceeds the station capacity, we introduce the variables $a_i$ to represent the difference between the number of allocated bikes $x_i$ and $g_i$ at station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$. Similarly, the variables $b_i$ represent the difference between the number of free racks $Q_i - x_i$ and $h_i$ at station $i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$.

The problem formulation is as follows:

$$\min_{x} \quad \sum_{i \in \mathcal{I}} f_i x_i + \mathbb{E}\left[Q(x, \xi)\right] \tag{110}$$

$$\text{s.t.} \quad x_i \geq g_i - a_i \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{111}$$

$$Q_i - x_i \geq h_i - b_i \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{112}$$

$$\bar{I}_{i0} + x_i \leq Q_i \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \tag{113}$$

$$\sum_{i \in \mathcal{I} \setminus \{|\mathcal{I}|\}} x_i \leq \bar{I}_{|\mathcal{I}|0} \tag{114}$$

$$x_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{115}$$

| Sets | |
|---|---|
| $\mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$ | set of stations, where the depot is station $|\mathcal{I}|$ |
| **Parameters** | |
| $\bar{I}_{|\mathcal{I}|0}$ | Depot capacity |
| $Q_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Maximum capacity of station $i$ |
| $\bar{I}_{i0}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Initial availability of bikes at station $i$ |
| $C$ | Maximum capacity of the relocation truck |
| $p_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | bikes' stock-out penalty at station $i$ |
| $c_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Excess penalty at station $i$ |
| $\frac{c_i}{Q_i}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Penalty associated to extra bikes placed at station $i$ after the re-balancing period |
| $f_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Allocation cost at station $i$ |
| $t_{i,i+1}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Re-balancing cost to allocate bikes from station $i$ to $i+1$ |
| $\xi_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Net demand of station $i$ |
| $g_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Maximum number of consecutive bikes withdrawn from station $i$ |
| $h_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | maximum number of consecutive bikes returned from station $i$ |
| **Variables** | |
| $x_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Amount of bikes to allocate at station $i$ in the first stage |
| $a_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Slack units between the number of allocated bikes in the first stage and the maximum number of consecutive bikes withdrawn at station $i$ |
| $b_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Slack units between the number of available racks and the maximum consecutive bikes returned at station $i$. |
| $y_{i,i+1}, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Amount of bikes to distribute from station $i$ to $i+1$ in the second stage |
| $I_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Inventory or balance of bikes at station $i$ |
| $I_i^+, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Surplus of bikes at station $i$ |
| $I_i^-, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Number of stock-out bikes at station $i$ |
| $B_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Extra bikes balance at station $i$ during the re-balancing period |
| $B_i^+, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Balance of of extra bikes at station $i$ during the re-balancing period |
| $E_i, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Excess inventory balance of bikes at station $i$ during the re-balancing period |
| $E_i^+, \quad i \in \mathcal{I} \setminus \{|\mathcal{I}|\}$ | Excess of bikes at station $i$ during the re-balancing period |

Table 2: Variables, parameters, and sets for Problem 4 in AAAAA.

where the second-stage cost function $Q(x, \xi)$ is

$$Q(x,\xi) = \min \sum_{i \in \mathcal{I} \setminus \{|\mathcal{I}|\}} \left( t_{i,i+1} y_{i,i+1} + \frac{c_i}{Q_i} B_i^+ + c_i E_i^+ + p_i(-I_i^-) + p_i a_i + c_i b_i \right) \qquad (116)$$

$$\text{s.t.} \quad y_{i,i+1} \leq C \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (117)$$

$$I_{|\mathcal{I}|} = \bar{I}_{|\mathcal{I}|0} - \sum_{i \in \mathcal{I} \setminus \{|\mathcal{I}|\}} x_i + y_{|\mathcal{I}|-1, |\mathcal{I}|} \qquad \qquad (118)$$

$$I_{|\mathcal{I}|} \leq \bar{I}_{|\mathcal{I}|0} \qquad \qquad (119)$$

$$I_1 = \bar{I}_{|\mathcal{I}|0} + x_1 - \xi_1 - y_{1,2} \qquad \qquad (120)$$

$$I_i = \bar{I}_{i0} + x_i - \xi_i + y_{i-1,i} - y_{i,i+1} \qquad \qquad \forall i \in \mathcal{I} \setminus \{1, |\mathcal{I}|\} \quad (121)$$

$$I_i^+ = \max\{0, I_i\} \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (122)$$

$$I_i^- = \min\{0, I_i\} \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (123)$$

$$E_i = I_i^+ - Q_i \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (124)$$

$$E_i^+ = \max\{0, E_i\} \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (125)$$

$$B_i = I_i^+ - x_i - \bar{I}_{i0} - E_i^+ \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (126)$$

$$B_i^+ = \max\{0, B_i\} \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (127)$$

$$I_i, B_i, E_i \in \mathbb{R} \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (128)$$

$$y_{i,i+1}, I_i^+, B_i^+, E_i^+ \geq 0 \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (129)$$

$$I_i^- \leq 0 \qquad \qquad \forall i \in \mathcal{I} \setminus \{|\mathcal{I}|\} \quad (130)$$

The formulated problem is similar to the model presented in Section 6.5 but with slight differences. The objective functions (110) and (116) aim to minimize the total expected costs, obtained by summing over all the penalties for delivery, re-balancing, extra and excess inventory, and stock-out in addition to the penalties of the slack variables. Constraints (111) encourage the target inventory quantity to be greater than or equal to the maximum number of consecutive withdrawals, while constraints (112) encourage the number of free racks to be greater than or equal to the maximum number of consecutive returns. To guarantee that a feasible solution can be found if the sum of the maximum number of consecutive withdrawn and returned bikes is greater than the station capacity the slack variables $a_i$ and $b_i$ allow for deviations from these quantities. Constraints (113) ensure that the sum of the allocated quantity and the initial availability at each station does not exceed the station's capacity. Constraints (114) indicate that the total number of allocated bikes within stations is less than the available quantity at the depot. Constraints (117) ensure that the number of bikes transported by the vehicle during re-balancing never exceeds its capacity. Constraints (118) guarantee that the quantity of bikes at the depot at the end of each scenario equals the initial bike availability plus the amount received from the last visited station, minus the quantities delivered to stations. Additionally, constraints (119) ensure that the number of bikes at the depot at the end of the re-balancing period does not exceed its capacity. The flow balance constraints for bikes at the first station on the route differ from the remaining stations. Specifically, constraints (120) ensure that for the first visited station, the amount of bikes at the end of the re-balancing period equals the sum of the initially available quantity and the quantity received from the depot, minus the amounts used to satisfy the demand and those bikes that are redistributed to subsequent stations on the route. Similarly, constraints (121) determine the inventory position at a station other than the first, as a function of the initial inventory level, the number allocated, the number withdrawn/returned, and the number redistributed to another station. Constraints (122) and (123) determine the surplus and stock-out quantities for each station. Constraints (124) and (125) calculate the number of bikes at each station that exceeds station capacity. Constraints (126) and (127) determine when more bikes are positioned at a station after re-balancing than were initially allocated, but not more than station capacity.

## D.1   Results

We now show the results of applying the methods in Section 5 for the previous two-stage problem. The uncertain parameter is the daily demand for bikes at each station, and our methodology uses additional covariate information to obtain a forecast of the demand. As discussed in Section 6, we estimate the parameters $g_i$ and $h_i$ in (111)-(112) as fractions of the demand in node $i$. In our experiments, we use the same features, data, and methodology used in Section 6.5 . To evaluate each method's optimality gap, we randomly split the dataset into training and test sets with an 80:20 ratio. We then calculate the out-of-sample cost by averaging the bike-sharing problem cost over the test data. This process is repeated with 10 sets of training and test data.

Figure 7 displays the out-of-sample costs for the proposed multiple replications procedure for the model discussed in Cavagnini et al. (2024). Despite similar variance in out-of-sample results across all methods, the approaches that consider multi-scenarios such as ER-SAA, KNN, and SAA outperform other methods that use point-wise forecast approximation, such as AD, CART, M5+AD, and LS, showing lower out-of-sample costs. Additionally, application-driven forecast methods within the point-wise forecast methods, such as AD and M5+AD, perform considerably better than methods such as CART and LS. Even though both problems (86)-(105) and (110)-(130) are relatively similar, the results differ due to the functional dependencies among the right-hand side presented because of constraints (111)-(112), as discussed in Section 2, indicating the
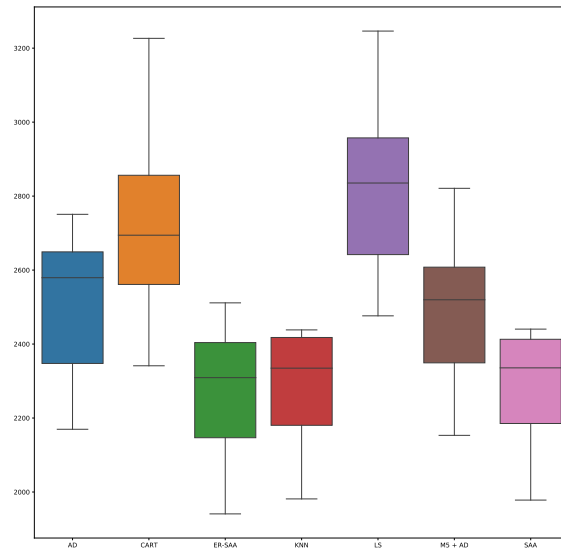
Figure 7: Average out-of-sample costs for 10 different train and test sets for methods proposed in Section 5.

need for a larger data set to solve this problem.

# References

N. Agrawal, M. A. Cohen, R. Deshpande, and V. Deshpande. How machine learning will transform supply chain management. *HARVARD BUSINESS REVIEW*, 103(3-4):128–137, 2024.

S. Arpón, T. Homem-de-Mello, and B. Pagnoncelli. Scenario reduction for stochastic programs with Conditional Value-at-Risk. *Mathematical Programming*, 170:327–356, 2018.

G.-Y. Ban and C. Rudin. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108, 2019.

Y. Bengio. Using a financial training criterion rather than a prediction criterion. *International Journal of Neural Systems*, 8(04):433–443, 1997.

D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106:1039–1082, 2017.

D. Bertsimas and N. Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.

D. Bertsimas and N. Mundru. Optimization-based scenario reduction for data-driven two-stage stochastic optimization. *Operations Research*, 71(4):1343–1361, 2023.

J. Blanchet, Y. Kang, and K. Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019. 10.1017/jpr.2019.49.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

R. Cavagnini. *Stochastic programming models for distribution logistics, bikesharing and production management.* PhD thesis, Università degli studi di Bergamo, 2019.

R. Cavagnini, F. Maggioni, L. Bertazzi, and M. Hewitt. A two-stage stochastic programming model for bike-sharing systems with rebalancing. *EURO Journal on Transportation and Logistics*, page 100140, 2024. ISSN 2192-4376. https://doi.org/10.1016/j.ejtl.2024.100140. URL `https://www.sciencedirect.com/science/article/pii/S2192437624000153`.

J. Dias-Garcia, A. Street, T. Homem-de-Mello, and F. D. Muñoz. Application-driven learning via joint prediction and optimization of demand and reserves requirement. *Operations Research*, 2024. Published online, DOI `https://doi.org/10.1287/opre.2023.0565`.

P. Donti, B. Amos, and J. Z. Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 5484–5494, 2017.

J. Dupačová, G. Consigli, and S. W. Wallace. Scenarios for multistage stochastic programs. *Ann Oper Res*, 100:25–53, 2000.

J. Dupačová, N. Gröwe-Kuska, and W. Römisch. Scenario reduction in stochastic programming: An approach using probability metrics. *Math Program*, 95:493–511, 2003.

A. N. Elmachtoub and P. Grigas. Smart "predict, then optimize". *Management Science*, 2021. URL `https://doi.org/10.1287/mnsc.2020.3922`.

J. Fairbrother, A. Turner, and S. W. Wallace. Problem-driven scenario generation: an analytical approach for stochastic programs with tail risk measure. *Mathematical Programming*, pages 1–42, 2019.

P. Grigas, M. Qi, and Z.-J. M. Shen. Integrated conditional estimation-optimization. arXiv preprint-arXiv:2110.12351, 2021.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Comput Optim Appl*, 24:187–206, 2003.

H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Math Program*, 118:371–406, 2009.

R. Henrion and W. Römisch. Problem-based optimal scenario generation and reduction in stochastic programming. *Mathematical Programming*, 191(1):183–205, 2022.

W. W. Hogan. Point-to-set maps in mathematical programming. *SIAM review*, 15(3):591–603, 1973.

T. Homem-de-Mello and G. Bayraksan. Monte Carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19:56–85, 2014.

K. Hoyland and S. W. Wallace. Generating scenario trees for multistage decision problems. *Manage Sci*, 47 (2):295–307, 2001.

K. Hoyland, M. Kaut, and S. W. Wallace. A heuristic for moment-matching scenario generation. *Comput Optim Appl*, 24:169–185, 2003.

R. Kannan, G. Bayraksan, and J. R. Luedtke. Data-driven sample average approximation with covariate information. *arXiv preprint arXiv:2207.13554*, 2022.

J. Keutchayan, J. Ortmann, and W. Rei. Problem-driven scenario clustering in stochastic optimization. *Computational Management Science*, 20(1):13, 2023.

T. Kleinert and M. Schmidt. Computing feasible points of bilevel problems with a penalty alternating direction method. *INFORMS Journal on Computing*, 33(1):198–215, 2021.

A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

H. Leövey and W. Römisch. Quasi-monte carlo methods for linear two-stage stochastic programming problems. *Mathematical Programming*, 151:315–345, 2015.

W.-K. Mak, D. P. Morton, and R. K. Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1-2):47–56, 1999.

J. Mandi, V. Bucarey, M. M. K. Tchomba, and T. Guns. Decision-focused learning: Through the lens of learning to rank. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14935–14947. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mandi22a.html.

S. Mehrotra and D. Papp. A cutting surface algorithm for semi-infinite convex programming with an application to moment robust optimization. *SIAM Journal on Optimization*, 24(4):1670–1697, 2014.

P. Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2): 115–166, 2018.

M. Muñoz, S. Pineda, and J. Morales. A bilevel framework for decision-making under uncertainty with contextual information. *Omega*, 108:102575, 2022. ISSN 0305-0483.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.

G. C. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming, Series B*, 89(2):251–271, 2001.

G. C. Pflug and A. Pichler. Approximations for probability distributions and stochastic optimization problems. In M. Bertocchi, G. Consigli, and M. A. H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy*, pages 343–387. Springer, 2011.

J. R. Quinlan. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, pages 343–348. World Scientific, 1992.

S. M. Robinson. Bounds for error in the solution set of a perturbed linear program. *Linear Algebra and its applications*, 6:69–81, 1973.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, Berlin, 1998.

W. Römisch. Stability of stochastic programming problems. In A. Ruszczyński and A. Shapiro, editors, *Handbook of Stochastic Optimization*. Elsevier Science Publishers B.V., Amsterdam, Netherlands, 2003.

U. Sadana, A. Chenreddy, E. Delage, A. Forel, E. Frejinger, and T. Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 2024.

A. Shapiro and H. Xu. Stochastic mathematical programs with equilibrium constraints, modelling and sample average approximation. *Optimization*, 57(3):395–418, 2008. 10.1080/02331930801954177.

A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on stochastic programming: modeling and theory.* SIAM, 3rd edition, 2021.

S. Verwer and Y. Zhang. Learning optimal classification trees using a binary linear program formulation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1625–1632, 2019.

S. W. Wallace. Decision making under uncertainty: Is sensitivity analysis of any use? *Operations Research*, 48(1):20–25, 2000.

W. Zhang, K. Wang, A. Jacquillat, and S. Wang. Optimized scenario reduction: Solving large-scale stochastic programs with quality guarantees. *INFORMS Journal on Computing*, 35(4):886–908, 2023.