Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming

Alexander Shapiro[*]

School of Industrial and Systems Engineering,
Georgia Institute of Technology,
Atlanta, Georgia 30332-0205,
e-mail: ashapiro@isye.gatech.edu

Abstract. In this tutorial we discuss several aspects of modeling and solving multistage stochastic programming problems. In particular we discuss distributionally robust and risk averse approaches to multistage stochastic programming, and the involved concept of time consistency. This tutorial is aimed at presenting a certain point of view on multistage stochastic optimization, rather than a complete survey of the topic.

Keywords: Stochastic programming, distributional robustness, dynamic equations

1 Introduction

There are many practical problems where one has to make decisions sequentially based on data (observations) available at time of the decision. Trying to make such decisions under uncertainty in some optimal way looking forward in time leads to the area of multistage optimization. Traditionally uncertainty was modeled as randomness using tools of probability theory. However uncertainty can come in many different ways where the basic concept of probability distribution could be questionable. Even if there is available data from which a relevant probability distribution could be estimated, there are many modeling questions. A useful model should on one hand represent reality in a reasonable way and on the other hand should be computationally manageable. This raises a nontrivial question of a balance/compromise between these two, often contradictory, requirements. This tutorial is aimed at presenting a certain point of view on multistage stochastic optimization rather than a complete survey of the topic. There are many technical issues, sometimes quite nontrivial, involved in the presented material. We try here to avoid discussion of complicated technical details whenever it is reasonably possible.

Development of theory and practice of sequential decisions is going back to the pioneering work of Abraham Wald on sequential analysis more than 70 years ago. Traditionally there

[*] This research was partly supported by NSF grant 1633196 .

are three modelling approaches dealing with sequential optimization problems, namely the Markov Decision Process (MDP), Stochastic Optimal Control (SOC) and Stochastic Programming. Although they deal with related classes of problems, the employed modeling and solution methods are somewhat different. For the theory of MDP and SOC we can refer to the classical monographs by Puterman [27] and Bertsekas and Shreve [7]. From the beginning, the MDP and SOC methodologies were based on dynamic programming equations. On the other hand, the stochastic programming started in works of Beale [2] and Dantzig [12], originally was concerned with a two stage setting and was based on a finite number of generated scenarios.

Recently an alternative to stochastic approach was suggested by the so-called adjustable robust multistage modeling of an uncertain process. This originated in Ben-Tal, Goryashko, Guslitzer and Nemirovski [5], we can refer to Ben-Tal, El Ghaoui and Nemirovski [4] for a thorough discussion of that approach, and to Yanikoğlu, Gorissen and Hertog [42] for a recent survey. Which of these two approaches - stochastic or robust, should be applied depends on a particular application. Also what became known in recent literature as distributionally robust stochastic programming, is somewhat between these two approaches to dealing with uncertainty. It is closely related to the so-called risk averse stochastic programming. In the next section we demonstrate some aspects of the involved issues on the classical example of the inventory model (we refer to Zipkin [43] for a thorough discussion of that model).

We use the following notation and terminology throughout the paper. We denote $[a]_+ := \max\{0, a\}$, by the notation ':=' we mean 'equal by definition'. By $\mathbb{E}[X]$ we denote the expectation of random variable X, and by $\mathbb{E}_{|Y}[X]$ the conditional expectation of X given random variable Y. By $\mathbf{1}_A$ we denote the indicator function of set A, i.e., $\mathbf{1}_A(\omega) = 1$ if $\omega \in A$ and $\mathbf{1}_A(\omega) = 0$ if $\omega \notin A$. A sample space $(\Omega, \mathcal{F})$ consists of an abstract set Ω and sigma algebra $\mathcal{F}$ of subsets of Ω. In particular if $\Omega = \{\omega_1, ..., \omega_m\}$ is finite, then we assume that the associated sigma algebra $\mathcal{F}$ consists of all subsets of Ω. If P and Q are two probability measures on $(\Omega, \mathcal{F})$, then it is said that Q is absolutely continuous with respect to P if $A \in \mathcal{F}$ and $P(A) = 0$ implies that $Q(A) = 0$. By the Radon–Nikodym Theorem, Q is absolutely continuous with respect to P if and only if there exists density function $f = dQ/dP$ such that $Q(A) = \int_A f dP$ for any $A \in \mathcal{F}$. If $\Omega = \{\omega_1, ..., \omega_m\}$ is finite and P is a probability measure (distribution) on Ω such that probability $p_i = P(\{\omega_i\})$, $i = 1, ..., m$, of every elementary event is positive, then any probability measure Q on Ω is absolutely continuous with respect to P. Recall that random process $\boldsymbol{\xi}_t \in \mathbb{R}^{d_t}$, $t = 1, ..., T$, is Markovian if the conditional distribution of $\boldsymbol{\xi}_{t+1}$ given $\boldsymbol{\xi}_{[t]}$ is the same as the conditional distribution of $\boldsymbol{\xi}_{t+1}$ given $\boldsymbol{\xi}_t$, $t = 1, ..., T-1$, where $\boldsymbol{\xi}_{[t]} := (\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_t)$ denotes history of the data process $\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_T$, up to time t. We use bold face notation $\boldsymbol{\xi}_t$ for random vector to distinguish it from ξ_t considered as a vector variable.

2 Inventory model

Suppose that a company has a planning horizon of T periods of time. At each time period $t = 1, ..., T$, the company has to make a decision of placing an order to replenish its inventory level to x_t. In the sequel we sometimes refer to the time period when a decision is made as *stage*. The cost incurred at stage t is

$$c_t(x_t - y_t) + b_t[d_t - x_t]_+ + h_t[x_t - d_t]_+. \tag{2.1}$$

Here c_t, b_t, h_t are the ordering cost, backorder penalty cost and holding cost per unit, respectively, y_t is the current inventory level and d_t is the demand at time t. Clearly the order quantity $x_t - y_t$ should be nonnegative, i.e., x_t should be not less than y_t. A natural objective is to minimize the total cost over the T periods (stages).

Let us take a close look at modeling of this problem. At the beginning of the planning horizon the manager wants to make a decision about order quantities $x_t - y_t$ for every time period $t = 1, ..., T$. It is assumed that the initial inventory level y_1 and the parameters c_t, b_t, h_t, $t = 1, ..., T$, are known. What is not known to the manager are future values of the demand d_t. This raises questions of how to model uncertainty of the demand process, what is the objective to optimize (minimize) and for how many periods T to look ahead. A classical approach is to model the demand process $D_1, ..., D_T$ as a stochastic process with a specified (joint) probability distribution (we use capital letter D_t for the demand at time t viewed as a random variable, and d_t for its particular realization). At each time period t we know the past, i.e., we know realization $d_{[t-1]} = (d_1, ..., d_{t-1})$ of the demand process. Naturally our decision should be based on that knowledge, i.e., $x_t = x_t(d_{[t-1]})$ should be a *function* of the realization of the demand process until time t. The first stage decision x_1 is deterministic, it is made before observing any realization of the random data becomes available, and hence should not depend on realizations of the demand process. (For uniformity of the notation we can write $x_1 = x_1(d_0)$ and think about $D_0 = d_0$ as deterministic.) At the beginning of the planning horizon, by defining functions $x_t = x_t(d_{[t-1]})$, $t = 1, ..., T$, the manager assigns a policy (also called decision rule) describing what order decision to make for any future realization of the demand process.

It should be clearly understood that the decision variables $x_t = x_t(d_{[t-1]})$, $t = 1, ..., T$, here are *functions* of the demand process. By substituting random variable D_t instead of its particular realization d_t, we view decisions $\boldsymbol{x}_t = x_t(D_{[t-1]})$ as random variables. We use the bold face script $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ when considering these variables as *random*, rather than their respective realizations x_t and y_t.

Next important choice is what objective to optimize. In this modeling the total cost is a function of the demand process and a chosen policy, hence is a random variable. One possible approach is to minimize the expected value of the total cost over the planning horizon. This can be written as the following optimization problem

$$\begin{array}{cl} \min\limits_{\boldsymbol{x}_t \geq \boldsymbol{y}_t} & \mathbb{E}\left[\sum\limits_{t=1}^{T} c_t(\boldsymbol{x}_t - \boldsymbol{y}_t) + b_t[D_t - \boldsymbol{x}_t]_+ + h_t[\boldsymbol{x}_t - D_t]_+\right] \\ \text{s.t.} & \boldsymbol{y}_{t+1} = \boldsymbol{x}_t - D_t,\ t = 1, ..., T. \end{array} \tag{2.2}$$

Note that actually there are two sets of decision variables in this formulation, namely x_t and y_t which are related by the balance equations $y_{t+1} = x_t - d_t$. For a chosen policy $x_t = x_t(d_{[t-1]})$, $t = 1, ..., T$, the inventory levels y_t are completely defined by these balance equations and the initial value y_1. We use notation $\pi = \big(x_1, x_2(d_{[1]}), ..., x_T(d_{[T-1]})\big)$ for a considered policy. The optimization (minimization) in (2.2) is performed over policies satisfying the respective feasibility constraints. The expectation in (2.2) is taken with respect to the specified (joint) probability distribution of the demand process $D_1, ..., D_T$, with $\boldsymbol{x}_t = x_t(D_{[t-1]})$, $t = 2, ..., T$, viewed as *random variables*. The feasibility constraints $\boldsymbol{x}_t \geq \boldsymbol{y}_t$, $t = 1, ..., T$, should be satisfied for almost every (a.e.) realization of the demand process $D_1, ..., D_T$.

Unless stated otherwise we assume that the probability distribution of the demand process *does not* depend on the chosen policy. In some cases it is natural to consider situations where our actions can change the demand distribution, and hence the expectation, denoted $\mathbb{E}^\pi$, depends on the policy. This will make the problem much more difficult from the modeling and computational points of view.

In the above formulation (2.2) the total cost is minimized *on average*. This could be justified by the Law of Large Numbers. That is, if this procedure is repeated many times under more or less the same probabilistic assumptions, then optimizing on average makes sense. It should be noted, however, that for a particular realization of the demand process the total cost could be quite different from its expected value. Also the above problem (2.2) depends on the specified probability distribution of the demand process. This raises the question of controlling the risk, which we will discuss later.

For a considered policy π we need to compute the expected value $\mathbb{E}[Z]$ of the total cost

$$Z := \sum_{t=1}^{T} c_t(\boldsymbol{x}_t - \boldsymbol{y}_t) + b_t[D_t - \boldsymbol{x}_t]_+ + h_t[\boldsymbol{x}_t - D_t]_+. \tag{2.3}$$

Recall that by using the bold face $\boldsymbol{x}_t = x_t(D_{[t-1]})$ and $\boldsymbol{y}_t$ we emphasize that these are random variables associated with the considered policy; consequently the total cost Z is a random variable. The expected value of the total cost depends on the chosen policy and the specified distribution of the demand process. From the modeling point of view typically it is natural to model the corresponding random data process (here the demand process) as having a *continuous* distribution with an infinite number of possible realizations. However, this leads to two major difficulties. One is how to compute the expected value of the total cost. The other is that the minimization should be performed over feasible policies which are functions of the demand process. In case the demand process has an infinite number of possible realizations this leads to an infinite dimensional optimization problem. The standard approach, used in stochastic programming, is to discretize the random data process. Let us have a closer look at that approach.

As it was pointed above our decision at each period time t depends on the realization $d_{[t-1]} = (d_1, ..., d_{t-1})$ of the demand process up to time t. We also assume that we know (specify) the conditional probability distribution of the future demand $(D_t, ..., D_T)$ given $D_{[t-1]} = d_{[t-1]}$. This is an important point, our decisions are conditional and consequently we need to model the demand distribution in a conditional way. This leads to the concept of scenario tree. Starting with a root node we need to construct sample paths of the demand process. One can think here about the root node as known value of the demand at time $t = 0$. Then at the next time period $t = 1$ possible values of realizations of the demand are specified. Conditional on these realizations, possible realizations of the demand at time $t = 2$ are specified, and so on until the end of the planning horizon. Each path of the constructed tree represents a possible realization of the demand process and is called scenario. Each node at time t represents history of the process up to time t. If the number of leaves at each node at time t is the same, say N_t, then the total number of scenarios is $N = \prod_{t=1}^{T-1} N_t$. Such scenario tree represents possible paths of the demand process. Still it needs to be equipped with respective conditional probabilities to make it a stochastic process.

There are several problems with the scenario tree approach. One is the modeling question of how to construct such a tree. In some cases we have at our disposal historical data from which we can try to construct the probability distribution of the demand process. Typically we observe just one realization (sample path) of the data process, we cannot go back in time to make another measurement. On the other hand we need to evaluate *conditional* distributions of the data process. The only way to deal with this problem is to make structural assumptions about the data process. The most natural and often used approach is to assume a Markovian structure of the process. One classical approach is to use time series analysis to fit an autoregressive model to the observed data.

Another difficulty is computational. Our decision variables, with respect to which we need to perform the optimization, are *policies* (decision rules) $\pi = \big(x_1, x_2(d_{[1]}), ..., x_T(d_{[T-1]})\big)$. Since the scenario tree is finite, the number of considered policies is also finite and is proportional to the number of scenarios. In principle it is also possible to consider scenario trees with an infinite number of scenarios - although useful as a conceptual visualization of the data process, it is not directly applicable for computational purposes. In order to make a reasonable approximation of the data process one needs to continue branching of scenarios at all time periods. This leads to a quick (exponential) growth of the total number of scenarios. For example if at each stage we use just $N_t = 100$ leaves, the total number of scenarios is $N = 10^{2(T-1)}$, e.g., with $T = 4$ we have million scenarios. Note that employing just one leaf, $N_t = 1$, starting from a certain time period, makes the problem deterministic from that period on, ignoring variability of the data process. This suggests that for time periods $T \geq 4$, say, the scenario tree discretization approach could be impractical from the computational point of view. It could also be noted that solving the problem for the constructed scenario tree defines policies only for the scenarios of that tree. This does not say what decisions to make, except the first stage decision which is made before observing any realization of the data process, at least not in a direct way, for possible realizations of the data process different from the constructed ones (we will discuss this further in Section 6).

An alternative approach is suggested by dynamic programming. The expectation operator has the following decomposition property. If X and Y are two random variables having a joint distribution, then $\mathbb{E}[X] = \mathbb{E}\left[\mathbb{E}_{|Y}[X]\right]$, provided the corresponding expectations are well defined. This formula has a simple intuitive interpretation. In case of discrete distributions, if we partition values of X according to values of Y, for each partition compute the average and then take the average of these averages, it will be the total average of X. Using this property we can write the expected value of the total cost Z, defined in (2.3), in the following decomposable form

$$\mathbb{E}[Z] = \mathbb{E}\Big[\mathbb{E}_{|D_{[1]}}\big[\mathbb{E}_{|D_{[2]}}[\cdots \mathbb{E}_{|D_{[T-1]}}[Z]]\big]\Big]. \tag{2.4}$$

(Recall that the total cost Z is associated with the considered policy and is a function of the random demand process, and hence is random.) Hence problem (2.2) can be written as

$$\min_{(x_1,\dots,x_T(\cdot))\in\Pi} \mathbb{E}\Big[\mathbb{E}_{|D_{[1]}}\big[\mathbb{E}_{|D_{[2]}}[\cdots\mathbb{E}_{|D_{[T-1]}}[Z]]\big]\Big], \tag{2.5}$$

where Π is the set of feasible policies. The notation $(x_1, x_2(\cdot), \dots, x_T(\cdot))$ emphasizes that the minimization in (2.5) is performed over the respective functions. Recall that the first stage decision x_1 is deterministic and made before observing any realization of the random data process, and that a policy is feasible if it satisfies the feasibility constraints for a.e. realization of the data process.

This leads to the dynamic programming equations. Consider minimization with respect to the last decision $x_T = x_T(d_{[T-1]})$. Under mild regularity conditions the minimization and (conditional) expectation operators can be interchanged (see Section 5.1), i.e., the minimization with respect to x_T can be taken inside the expectation $\mathbb{E}_{|D_{[T-1]}}[Z]$. Continuing this process backwards in time, with respect to decision variables $x_{T-1}, \dots, x_1$, it is possible to write the following dynamic programming equations. The value (also called cost-to-go) functions $V_t(y_t, d_{[t-1]})$, $t = T, \dots, 2$, are given as optimal values of the respective problems

$$\min_{x_t \ge y_t} c_t(x_t - y_t) + \mathbb{E}_{|D_{[t-1]}=d_{[t-1]}}\Big[b_t[D_t - x_t]_+ + h_t[x_t - D_t]_+ + V_{t+1}\left(x_t - D_t, D_{[t]}\right)\Big], \tag{2.6}$$

with $V_{T+1}(\cdot,\cdot)$ omitted. Finally, at the first stage we need to solve the problem

$$\min_{x_1 \ge y_1} c_1(x_1 - y_1) + \mathbb{E}\Big[b_1[D_1 - x_1]_+ + h_1[x_1 - D_1]_+ + V_2(x_1 - D_1, D_1)\Big]. \tag{2.7}$$

The value $V_t(y_t, d_{[t-1]})$ represents the minimum expected value of the cost from the time period t upwards until the end T of the horizon, conditional on $D_{[t-1]} = d_{[t-1]}$ and inventory level y_t. A policy $\bar\pi = (\bar x_1, \dots, \bar x_T(d_{[T-1]}))$ is an optimal solution of the problem (2.2) if and only if each $\bar x_t$, $t = 1, \dots, T$, solves the respective minimization problem (2.6). In particular, the first stage optimal decision $\bar x_1$ and the optimal value of problem (2.2) are given by respective an optimal solution and the optimal value of problem (2.7) conditional on the initial inventory level y_1.

By writing the dynamic programming equations we reduce the original problem (2.2) to the sequence of finite dimensional optimization problems (2.6) - (2.7), with the respective *deterministic* decision variables x_t and y_t. A cost of that reduction is that we need to represent the value functions $V_t(y_t, d_{[t-1]})$ in a computationally feasible way in order to keep them in the computer memory going backward in time t. This is simplified dramatically if the process $D_1, \dots, D_T$ is *stagewise independent*, i.e., for each $t = 2, \dots, T$, the random variable D_t is independent of $D_{[t-1]}$. Then the conditional expectations in equations (2.6) become the corresponding unconditional expectations, and consequently value functions $V_t(y_t)$ can be viewed as functions of the respective univariate variables y_t only (this can be shown by induction going backwards in time). In that case at every stage we only need to remember the current inventory level y_t, and do not need to keep track of the respective history of the demand process. In the terminology of optimal control, variables y_t are called *state variables* and x_t called *control variables*. We will discuss this further in Section 6.

3 Risk neutral multistage programming

Consider now the following multistage stochastic programming problem

$$\begin{array}{ll} \min & \mathbb{E}\left[\sum_{t=1}^{T} c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\right] \\ \text{s.t.} & \boldsymbol{B}_t\boldsymbol{x}_{t-1} + \boldsymbol{A}_t\boldsymbol{x}_t = \boldsymbol{b}_t,\ \boldsymbol{x}_t \in \mathcal{X}_t,\ t = 1, ..., T, \end{array} \tag{3.1}$$

with the term B_1x_0 at the first stage omitted (alternatively x_0 can be viewed as the (given) initial state of the system). Here $\boldsymbol{\xi}_t \in \mathbb{R}^{d_t}$, $t = 1, ..., T$, is a random data process, $c_t : \mathbb{R}^{n_t} \times \mathbb{R}^{d_t} \to \mathbb{R}$ are costs functions, $b_t = b_t(\xi_t)$ are right side vectors, $B_t = B_t(\xi_t)$ and $A_t = A_t(\xi_t)$ are matrices of appropriate dimensions and $\mathcal{X}_t \subset \mathbb{R}^{n_t}$ are polyhedral sets, e.g., $\mathcal{X}_t = \{x_t \in \mathbb{R}^{n_t} : x_t \geq 0\}$. It is assumed that the first stage data vector ξ_1 is known (deterministic). As before we use the bold face script to emphasize which considered variables are random. To simplify the presentation we consider the data (b_t, B_t, A_t) as a function of ξ_t rather than the whole history $\xi_{[t]}$.

Optimization in (3.1) is performed over feasible policies (also called decision rules). A policy is a sequence of (measurable) functions $x_t = x_t(\xi_{[t]})$, $t = 1, ..., T$. Each $x_t(\xi_{[t]})$ is a function of the data process $\xi_{[t]} = (\xi_1, ..., \xi_t)$ up to time t, this ensures the *nonanticipative* property of a considered policy. The first stage decision x_1 is deterministic, i.e., does not depend on realizations of random data (here the process starts at time $t = 1$ with x_1 being deterministic, while in the inventory model, discussed in Section 2, the stage at time $t = 0$ is viewed as deterministic). Note again that the decisions $\boldsymbol{x}_t = x_t(\boldsymbol{\xi}_{[t]})$ become random when the data process is viewed as random. This is emphasized by using the respective bold face script in problem (3.1). The constraints should be satisfied for almost every realization of the random data process.

We denote by Π the set of feasible policies $\pi = \big(x_1, x_2(\xi_{[2]}), ..., x_T(\xi_{[T]})\big)$. It is assumed that the probability distribution of the data process $\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_T$ *does not* depend on the considered policies $\pi \in \Pi$. The constraints $B_tx_{t-1} + A_tx_t = b_t$ define the dynamics of the system and often represent balance equations. It is possible to consider more general formulations of stochastic programs with nonlinear dynamic constraints. Nevertheless the above formulation is sufficiently general to cover many practical applications and to demonstrate the main properties of multistage stochastic programs. In particular if the cost functions $c_t(x_t, \xi_t) = c_t(\xi_t)^\top x_t$ are linear, problem (3.1) becomes a multistage stochastic linear program. The inventory model, discussed in Section 2, can be considered in this framework with cost functions defined in (2.1).

An approach to solving problem (3.1), often used in stochastic programming, is to discretize distribution of the data process by constructing a scenario tree and consequently solving the obtained deterministic optimization problem. In the linear case this leads to a large scale linear program with the size proportional to the number of generated scenarios. In the two stage case ($T = 2$) one needs to discretize only the probability distribution of the random vector $\boldsymbol{\xi}_2$. Randomization techniques, such as the Sample Average Approximation (SAA) method, are shown to be quite efficient in that case. On the other hand, in the multistage case the number of scenarios, needed to approximate distribution of the data process with a reasonable accuracy, grows exponentially with increase of the number of stages. Consequently computational complexity, measured in terms of the number of scenarios, grows exponentially making this approach impractical for the number of stages say greater than three (this was already discussed in Section 2). Of course there could be situations where there is no need to look at say more than one step ahead. In that case one can take the approach of solving the respective two stage problems and recomputing the first stage solution every time the new observations become available. We will discuss this further in Section 6.

Next we discuss the alternative approach of dynamic programming which has its own limitations. Suppose for the moment that the multistage problem (3.1) is deterministic, i.e., the cost functions $c_t(x_t)$ and the corresponding constraints do not involve random parameters; of course this can be considered in terms of problem (3.1) with just one scenario. That is, consider the following problem

$$\begin{array}{cl} \min\limits_{x_1,\dots,x_T} & \sum_{t=1}^T c_t(x_t) \\ \text{s.t.} & B_t x_{t-1} + A_t x_t = b_t,\ x_t \in \mathcal{X}_t,\ t = 1,\dots,T. \end{array} \tag{3.2}$$

Here B_t, A_t and b_t are deterministic (not random) and minimization in (3.2) is performed over deterministic vectors $x_1, \dots, x_T$. We can approach problem (3.2) in the following way. First we perform minimization with respect to x_T. Consequently the obtained minimal value should be minimized with respect to $x_1, \dots, x_{T-1}$. This leads to the following minimization problem

$$\begin{array}{cl} \min\limits_{x_1,\dots,x_{T-1}} & \sum_{t=1}^{T-1} c_t(x_t) + V_T(x_{T-1}) \\ \text{s.t.} & B_t x_{t-1} + A_t x_t = b_t,\ x_t \in \mathcal{X}_t,\ t = 1,\dots,T-1, \end{array} \tag{3.3}$$

where $V_T(x_{T-1})$ is the optimal value of problem

$$\begin{array}{cl} \min\limits_{x_T \in \mathcal{X}_T} & c_T(x_T) \\ \text{s.t.} & B_T x_{T-1} + A_T x_T = b_T. \end{array} \tag{3.4}$$

The above problem (3.4) represents minimization with respect to x_T, its optimal value is a function of x_{T-1} only. Next we perform minimization in (3.3) with respect to x_{T-1}, which leads to the minimization problem

$$\begin{array}{cl} \min\limits_{x_{T-1} \in \mathcal{X}_{T-1}} & c_{T-1}(x_{T-1}) + V_T(x_{T-1}) \\ \text{s.t.} & B_{T-1} x_{T-2} + A_{T-1} x_{T-1} = b_{T-1}. \end{array} \tag{3.5}$$

Optimal value, denoted by $V_{T-1}(x_{T-2})$, of the above problem is a function of x_{T-2}. And so on continuing this process, going backwards in time, eventually at the first stage the following problem should be solved

$$\begin{array}{cl} \min\limits_{x_1 \in \mathcal{X}_1} & c_1(x_1) + V_2(x_1) \\ \text{s.t.} & A_1 x_1 = b_1. \end{array} \tag{3.6}$$

The idea of going backwards in time can be extended to the stochastic setting. This requires the following additional observations. Using the decomposability property of the expectation operator, problem (3.1) can be written in the following form (compare with (2.5))

$$\begin{array}{cl} \min & \mathbb{E}_{|\xi_1}\left[\cdots \mathbb{E}_{|\boldsymbol{\xi}_{[T-1]}}\left[\sum_{t=1}^T c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\right]\right] \\ \text{s.t.} & \boldsymbol{B}_t \boldsymbol{x}_{t-1} + \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t,\ \boldsymbol{x}_t \in \mathcal{X}_t,\ t = 1,\dots,T. \end{array} \tag{3.7}$$

Recall that ξ_1 is deterministic and hence $\mathbb{E}_{|\xi_1}$ is the same as the unconditional expectation, we write this in that form for uniformity of notation. Furthermore by interchanging the respective minimization and (conditional) expectation operators, problem (3.1) can be written in the following nested form (see Section 5.1 for a discussion of the interchangeability principle)

$$\begin{array}{r}\min\limits_{\substack{A_1x_1=b_1\\x_1\in\mathcal{X}_1}} c_1(x_1) \;+\; \mathbb{E}_{|\xi_1}\Big[\min\limits_{\substack{\boldsymbol{B}_2x_1+\boldsymbol{A}_2\boldsymbol{x}_2=\boldsymbol{b}_2\\\boldsymbol{x}_2\in\mathcal{X}_2}} c_2(\boldsymbol{x}_2,\boldsymbol{\xi}_2)+\cdots\\ +\mathbb{E}_{|\boldsymbol{\xi}_{[T-1]}}\Big[\min\limits_{\substack{\boldsymbol{B}_T\boldsymbol{x}_{T-1}+\boldsymbol{A}_T\boldsymbol{x}_T=\boldsymbol{b}_T\\\boldsymbol{x}_T\in\mathcal{X}_T}} c_T(\boldsymbol{x}_T,\boldsymbol{\xi}_T)\Big]\Big].\end{array} \tag{3.8}$$

The nested formulation leads to the dynamic programming equations. That is, at stage $t = T$ the value (cost-to-go) function $V_T(x_{T-1}, \xi_T)$ is given by the optimal value of the problem (recall that in the stochastic setting, elements of B_T, A_T and b_T can be also functions of ξ_T)

$$\begin{array}{ll}\min\limits_{x_T\in\mathcal{X}_T} & c_T(x_T,\xi_T)\\ \text{s.t.} & B_Tx_{T-1}+A_Tx_T=b_T.\end{array} \tag{3.9}$$

Note that variable x_T in (3.9) is viewed as a deterministic vector $x_T \in \mathbb{R}^{n_T}$. The optimal value $V_T(x_{T-1}, \xi_T)$ of problem (3.9) is a function of $x_{T-1} \in \mathbb{R}^{n_{T-1}}$ and realization ξ_T of the random vector $\boldsymbol{\xi}_T$. At stages $t = T-1, ..., 2$, the value function $V_t(x_{t-1}, \xi_{[t]})$ is given by the optimal value of the problem

$$\begin{array}{ll}\min\limits_{x_t\in\mathcal{X}_t} & c_t(x_t,\xi_t)+\mathcal{V}(x_t,\xi_{[t]})\\ \text{s.t.} & B_tx_{t-1}+A_tx_t=b_t,\end{array} \tag{3.10}$$

where

$$\mathcal{V}(x_t,\xi_{[t]}) := \mathbb{E}_{|\boldsymbol{\xi}_{[t]}=\xi_{[t]}}[V_{t+1}(x_t,\boldsymbol{\xi}_{[t+1]})]$$

is the conditional expectation of $V_{t+1}(x_t, \boldsymbol{\xi}_{[t+1]})$ given realization $\xi_{[t]}$ of $\boldsymbol{\xi}_{[t]}$. At the first stage the following problem should be solved

$$\begin{array}{ll}\min\limits_{x_1\in\mathcal{X}_1} & c_1(x_1)+\mathbb{E}[V_2(x_1,\boldsymbol{\xi}_2)]\\ \text{s.t.} & A_1x_1=b_1.\end{array} \tag{3.11}$$

As it was pointed out in the previous section, the dynamic programming equations reduce the original optimization problem to the sequence of finite dimensional optimization problems (3.9) - (3.11) with respect to *deterministic* decision variables x_t. The value (cost-to-go) function $V_t(x_{t-1}, \xi_{[t]})$, at time t, depends on $x_{t-1} \in \mathbb{R}^{n_{t-1}}$ because of the balance equations $B_tx_{t-1} + A_tx_t = b_t$. It also depends on the history $\xi_{[t]}$ of the data process because the conditional expectation $\mathbb{E}_{|\boldsymbol{\xi}_{[t]}}$ involves the conditional distribution of $\boldsymbol{\xi}_{[t+1]}$ given realization $\xi_{[t]}$ of $\boldsymbol{\xi}_{[t]}$. Dependence on so many variables makes it inapplicable for computational purposes. The situation simplifies if we assume that the data process has a Markovian structure. From the statistical point of view the simplest case is when the process $\boldsymbol{\xi}_t$, $t = 1, ..., T$, is *stagewise independent*, i.e., random vector $\boldsymbol{\xi}_{t+1}$ is independent of $\boldsymbol{\xi}_{[t]}$, $t = 1, ..., T-1$. In the stagewise independent case, since $\boldsymbol{\xi}_T$ is independent of $\boldsymbol{\xi}_{[T-1]}$, we have that the conditional expectation $\mathbb{E}_{|\boldsymbol{\xi}_{[T-1]}}[V_T(x_{T-1}, \boldsymbol{\xi}_T)]$ is the same as the unconditional expectation $\mathbb{E}[V_T(x_{T-1}, \boldsymbol{\xi}_T)]$. Hence

$V_{T-1}(x_{T-2}, \xi_{T-1})$ is a function of ξ_{T-1} rather than $\xi_{[T-1]}$. Continuing this backwards in time we obtain that for $t = T-1, ..., 2$, the cost-to-go function $V_t(x_{t-1}, \xi_t)$ is given by the optimal value of the problem

$$\begin{array}{ll} \min\limits_{x_t \in \mathcal{X}_t} & c_t(x_t, \xi_t) + \mathcal{V}_{t+1}(x_t) \\ \text{s.t.} & B_t x_{t-1} + A_t x_t = b_t, \end{array} \tag{3.12}$$

where $\mathcal{V}_{t+1}(x_t) := \mathbb{E}[V_{t+1}(x_t, \boldsymbol{\xi}_{t+1})]$ is the expected cost-to-go function (the expectation is taken with respect to the distribution of $\boldsymbol{\xi}_{t+1}$). Formally this can be shown by induction in $t = T-1, ..., 2$. Consequently in the stagewise independent case we only need to keep track of the functions $\mathcal{V}_{t+1}(x_t)$ of x_t alone, when solving problems (3.12). Yet one of the main difficulties of the dynamic programming approach is how to represent functions $\mathcal{V}_{t+1}(x_t)$ in the computer when the dimension of x_t is large, this is the so-called “curse of dimensionality”. We will discuss this further in Section 6.

The relevant and important question is how many variables are needed for representation of the cost-to-go functions. In a somewhat informal way such variables are called *state variables*. That is, at every stage we need to keep track of state variables only. For example in the inventory model at every stage we only need to know the current inventory level y_t, provided the demand process is stagewise independent. If the demand process is not stagewise independent, then the distribution of the current demand D_t depends on realizations at previous stages and hence we need to remember these realizations as well. Of course, in the stagewise independent case, we can refer to all components of decision vectors x_t of the multistage problem (3.1) as state variables. Naturally one can think about a minimal number of state variables to represent the value functions. This question depends on a considered model. In the approach of optimal control (below) there is an explicit separation between state and control variables.

3.1 Optimal control model

Problem (3.1) can be compared with the optimal control (in discrete time) formulation (e.g., Bertsekas and Shreve [7])

$$\begin{array}{ll} \min & \mathbb{E}\left[\sum_{t=1}^{T} c_t(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_t) + c_{T+1}(\boldsymbol{y}_{T+1})\right], \\ \text{s.t.} & \boldsymbol{y}_{t+1} = F_t(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_t),\ t = 1, ..., T, \\ & \boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{y}_t),\ t = 1, ..., T. \end{array} \tag{3.13}$$

Here variables $y_t \in \mathbb{R}^{n_t}$, $t = 1, ..., T+1$, represent state of the system, $u_t \in \mathbb{R}^{m_t}$, $t = 1, ..., T$, are controls, $\boldsymbol{\xi}_t \in \mathbb{R}^{d_t}$, $t = 1, ..., T$, are random vectors (random noise or disturbances), $c_t : \mathbb{R}^{n_t} \times \mathbb{R}^{m_t} \times \mathbb{R}^{d_t} \to \mathbb{R}$, $t = 1, ..., T$, are cost functions, $c_{T+1}(y_{T+1})$ is final cost function, $F_t : \mathbb{R}^{n_t} \times \mathbb{R}^{m_t} \times \mathbb{R}^{d_t} \to \mathbb{R}^{n_{t+1}}$ are measurable mappings, and $\mathcal{U}_t : \mathbb{R}^{n_t} \rightrightarrows \mathbb{R}^{m_t}$ are multifunctions (point-to-set mappings). Values y_1 and ξ_0 are deterministic (initial conditions). The optimization in (3.13) is performed over policies satisfying the corresponding feasibility constraints almost surely (with probability one). The inventory model (discussed in Section 2) is of that form with cost functions (2.1), state variables y_t, control variables x_t, random disturbances D_t, balance equations $y_{t+1} = x_t - d_t$ and control constraints $\mathcal{U}_t(y_t) := \{x_t : x_t \geq y_t\}$.

It is possible to view problem (3.13) as a stochastic programming problem with decision variables $x_t = (y_{t+1}, u_t)$. If moreover mappings $F_t(y_t, u_t, \xi_t)$ are linear (affine), then problem (3.13) can be considered as a particular case of problem (3.1). The corresponding dynamic programming equations here take the form (compare with dynamic equations of the inventory model)

$$V_t(y_t, \xi_{[t-1]}) = \inf_{u_t \in \mathcal{U}_t(y_t)} \mathbb{E}_{|\boldsymbol{\xi}_{[t-1]}=\xi_{[t-1]}} \left[c_t(y_t, u_t, \boldsymbol{\xi}_t) + V_{t+1}\big(F_t(y_t, u_t, \boldsymbol{\xi}_t), \boldsymbol{\xi}_{[t]}\big) \right], \tag{3.14}$$

$t = 2, ..., T$. In optimal control the random process ξ_t, $t = 1, ..., T$, is often viewed as noise and assumed to be stagewise independent. In that case, value (expected cost-to-go) functions $V_t(y_t)$ depend only on the state variables y_t and the conditional expectation in (3.14) becomes the respective unconditional expectation. Also the minimizers $\bar{u}_t = \bar{\pi}_t(y_t)$, $t = 1, ..., T$, on the right hand side of (3.14), provided that such minimizers exist, define an optimal policy for problem (3.13). That is, in the stagewise independent case optimization in (3.13) can be performed over policies of the form $u_t = \pi_t(y_t)$, $t = 1, ..., T$.

If the data process is Markovian, then the value (cost-to-go) function $V_t(y_t, \xi_{t-1})$ also depends on ξ_{t-1} with the conditional expectation in (3.14) taken with respect to ξ_{t-1} rather than $\xi_{[t-1]}$. In that case one can think about (y_t, ξ_{t-1}) as state variables. For example, suppose that the data process can be modeled as the first order autoregressive time series $\xi_{t+1} = \mu + \Phi\xi_t + \varepsilon_t$, where μ is vector and Φ is matrix of estimated parameters and ε_t is the error process assumed to be independent identically distributed (iid). Then in terms of state variables $z_t = (y_t, \xi_t)$, problem (3.13) can be written as

$$\begin{array}{ll} \min & \mathbb{E}\left[\sum_{t=1}^{T} c_t(\boldsymbol{z}_t, \boldsymbol{u}_t) + c_{T+1}(\boldsymbol{y}_{T+1})\right], \\ \text{s.t.} & \boldsymbol{z}_{t+1} = \hat{F}_t(\boldsymbol{z}_t, \boldsymbol{u}_t, \boldsymbol{\varepsilon}_t),\ t = 1, ..., T, \\ & \boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{y}_t),\ t = 1, ..., T, \end{array} \tag{3.15}$$

with $\hat{F}_t(z_t, u_t, \varepsilon_t) := (F_t(y_t, u_t, \xi_t), \mu + \Phi\xi_t + \varepsilon_t)$ and $\boldsymbol{\varepsilon}_t$ viewed as the noise process.

4 Distributionally robust and risk averse stochastic programming

One of the basic modeling assumptions, used in the previous section, is that the probability distribution of the random data is specified exactly and moreover does not depend on our actions. In real applications the "true" distribution is never known exactly and at best could be estimated from available data. This motivates to consider a worst distribution approach. Such min-max approach has a long history, it originated in John von Neumann's game theory. In an elegant paper Scarf [32] analyzed the one stage inventory model (called the Newsvendor problem) when only the mean and variance of the distribution of the demand are known. In stochastic programming discussion of the worst distribution approach goes back at least to Žáčková [40]. In the recent literature it is often referred to as distributionally robust stochastic programming.

4.1 Static setting

In the static setting distributionally robust stochastic problems can be formulated in the following minimax form

$$\min_{x\in\mathcal{X}} \sup_{Q\in\mathfrak{M}} \mathbb{E}_Q[F(x,\omega)]. \tag{4.1}$$

Here $\mathcal{X}$ is a nonempty subset of $\mathbb{R}^n$, $\mathfrak{M}$ is a specified set of probability measures (distributions) on a sample space $(\Omega,\mathcal{F})$, and $F:\mathcal{X}\times\Omega\to\mathbb{R}$ is an objective function. It is assumed that for every $x\in\mathcal{X}$ the random variable $F_x(\omega)=F(x,\omega)$ is $\mathcal{F}$-measurable and the expectation $\mathbb{E}_Q[F_x(\omega)]=\int_\Omega F_x(\omega)dQ(\omega)$, with respect to every $Q\in\mathfrak{M}$, is well defined and finite valued. Recently such distributionally robust stochastic programs became a hot topic of research with various suggestions for the so-called ambiguity set $\mathfrak{M}$ of distributions (see, e.g., Wiesemann, Kuhn, and Sim [41] and references therein).

With the set $\mathfrak{M}$ we can associate the following functional

$$\mathcal{R}(Z) := \sup_{Q\in\mathfrak{M}} \mathbb{E}_Q[Z]. \tag{4.2}$$

In order for the functional $\mathcal{R}$ to be well defined, we need to restrict the set of considered random variables $Z:\Omega\to\mathbb{R}$. We assume that $\mathcal{R}:\mathcal{Z}\to\mathbb{R}$ is defined on a linear space $\mathcal{Z}$ of measurable functions (variables) $Z:\Omega\to\mathbb{R}$. That is, if values $\mathcal{R}(Z)$ and $\mathcal{R}(Z')$ are defined for some $Z,Z':\Omega\to\mathbb{R}$, then values $\mathcal{R}(Z+Z')$ and $\mathcal{R}(\alpha Z)$ are also defined for any $\alpha\in\mathbb{R}$. We also assume that if $Z\in\mathcal{Z}$ and $a\in\mathbb{R}$, then $Z+a\in\mathcal{Z}$. With this notation we can write problem (4.1) as

$$\min_{x\in\mathcal{X}} \mathcal{R}[F_x]. \tag{4.3}$$

The functional $\mathcal{R}$ has the following properties.

(i) *Subadditivity*, if $Z,Z'\in\mathcal{Z}$, then $\mathcal{R}(Z+Z')\le\mathcal{R}(Z)+\mathcal{R}(Z')$.

(ii) *Monotonicity*, if $Z,Z'\in\mathcal{Z}$ and $Z\ge Z'$, then $\mathcal{R}(Z)\ge\mathcal{R}(Z')$.

(iii) *Translation equivariance*, if $Z\in\mathcal{Z}$ and $a\in\mathbb{R}$, then $\mathcal{R}(Z+a)=\mathcal{R}(Z)+a$.

(iv) *Positive homogeneity*, if $Z\in\mathcal{Z}$ and $\alpha\ge 0$, then $\mathcal{R}(\alpha Z)=\alpha\mathcal{R}(Z)$.

Functionals satisfying the above properties (i)-(iv) were called *coherent risk measures* in Artzner et al [1] (for a discussion of coherent risk measures we can refer to Föllmer and Schied [14] and Shapiro, Dentcheva and Ruszczyński [36]).

It turns out that in a certain sense every coherent risk measure, satisfying conditions (i)-(iv), has a dual representation of the form (4.2). Suppose for the moment that the sample space $\Omega=\{\omega_1,...,\omega_m\}$ is finite equipped with sigma algebra of all its subsets. Then every probability distribution Q on $(\Omega,\mathcal{F})$ can be identified with a vector $\mathfrak{q}=(q_1,...,q_m)\in\Delta_m$, where

$$\Delta_m := \left\{\mathfrak{q}\in\mathbb{R}^m : \textstyle\sum_{i=1}^m q_i=1,\ \mathfrak{q}\ge 0\right\}. \tag{4.4}$$

Moreover, with every variable $Z:\Omega\to\mathbb{R}$ we can associate vector $z=(z_1,...,z_m)\in\mathbb{R}^m$ with components $z_i=Z(\omega_i)$, $i=1,...,m$. We can view then a coherent risk measure $\mathcal{R}$ as functional $\mathcal{R} : \mathbb{R}^m \to \mathbb{R}$. By the classical convex analysis the conditions (i) and (iv) imply that $\mathcal{R}$ is the support function of a set $\mathfrak{M} \subset \mathbb{R}^m$, i.e., $\mathcal{R}(z) = \sup_{\mathfrak{q} \in \mathfrak{M}} \mathfrak{q}^\top z$ for some set $\mathfrak{M} \subset \mathbb{R}^m$. Note that the set $\mathfrak{M}$ is not defined uniquely by the respective functional $\mathcal{R}$, and the functional $\mathcal{R}$ is not changed if $\mathfrak{M}$ is replaced by the topological closure of its convex hull. Therefore it is natural to assume that the considered set $\mathfrak{M}$ is convex and closed. It is possible to show that condition (ii) implies that every vector $\mathfrak{q} \in \mathfrak{M}$ has nonnegative components, and condition (iii) implies that $\sum_{i=1}^m q_i = 1$, and hence $\mathfrak{M}$ is a subset of Δ_m. That is, in the finite dimensional setting there is a one-to-one correspondence between coherent risk measures and functionals of the form

$$\mathcal{R}(Z) = \sup_{\mathfrak{q} \in \mathfrak{M}} \sum_{i=1}^m q_i Z(\omega_i), \quad \mathfrak{M} \subset \Delta_m. \tag{4.5}$$

In infinite dimensional settings the situation is more delicate. In order to derive a duality relation between coherent risk measures, satisfying conditions (i)-(iv), and functionals of the form (4.2) one needs to define a proper space $\mathcal{Z}$ of allowable variables $Z : \Omega \to \mathbb{R}$ and a dual space $\mathcal{Z}^*$ equipped with a scalar product $\langle \zeta, Z \rangle$ for any $Z \in \mathcal{Z}$ and $\zeta \in \mathcal{Z}^*$. One possible approach is the following. Suppose that there is a reference probability measure P defined on the sample space $(\Omega, \mathcal{F})$ and consider the corresponding probability space $(\Omega, \mathcal{F}, P)$. Let $\mathcal{Z}$ be the space of variables Z having finite p-th order moments, $p \in [1, \infty)$, i.e., measurable $Z : \Omega \to \mathbb{R}$ belongs to $\mathcal{Z}$ if $\mathbb{E}_P[|Z|^p]$ is finite. This space is denoted $\mathcal{Z} = L_p(\Omega, \mathcal{F}, P)$. The dual of the space $L_p(\Omega, \mathcal{F}, P)$ is the space $\mathcal{Z}^* = L_q(\Omega, \mathcal{F}, P)$, where $q \in (1, \infty]$ is such that $1/p + 1/q = 1$, with the scalar product $\langle \zeta, Z \rangle := \int_\Omega \zeta(\omega) Z(\omega) dP(\omega)$. In particular for $p = 1$ the dual of space $L_1(\Omega, \mathcal{F}, P)$ is the space $L_\infty(\Omega, \mathcal{F}, P)$ of essentially bounded variables. In this construction the set $\mathfrak{M}$ consists of probability measures absolutely continuous with respect to the reference measure P, i.e., each $Q \in \mathfrak{M}$ has density $\zeta = dQ/dP \in \mathcal{Z}^*$. And indeed it is possible to show that if $\mathcal{R} : L_p(\Omega, \mathcal{F}, P) \to \mathbb{R}$ is a (real valued) coherent risk measure satisfying the conditions (i)-(iv), then it has the dual representation

$$\mathcal{R}(Z) = \sup_{\zeta \in \mathfrak{A}} \int_\Omega \zeta(\omega) Z(\omega) dP(\omega), \; Z \in L_p(\Omega, \mathcal{F}, P), \tag{4.6}$$

with $\mathfrak{A} \subset \mathcal{Z}^*$ being a set of density functions, i.e., every $\zeta \in \mathfrak{A}$ is nonnegative valued and $\int_\Omega \zeta(\omega) dP(\omega) = 1$ (cf., Ruszczyński and Shapiro [31]). In that setting the probabilistic statement is made with respect to the reference probability measure P. In particular by writing $Z \geq Z'$ for some $Z, Z' \in \mathcal{Z}$ we mean that this inequality holds almost surely (a.s), i.e., $Z(\omega) \geq Z'(\omega)$ for all ω in the set Ω except on a subset of Ω of P-measure zero.

By the above discussion the distributionally robust and risk averse approaches (4.1) and (4.3) to stochastic optimization can be viewed as dual to each other. However, the difference is how the set $\mathfrak{M}$ of probability distributions and the risk measure $\mathcal{R}$ are defined in the respective settings. In the risk averse setting it is assumed existence of a reference probability measure P. It is natural then to consider *law invariant* risk functionals when $\mathcal{R}(Z)$ depends only on the distribution of Z. That is, the functional $\mathcal{R}$ is law invariant if $\mathcal{R}(Z) = \mathcal{R}(Z')$ whenever $Z, Z' \in \mathcal{Z}$ are distributionally equivalent with respect to P, i.e., $P(Z \leq z) = P(Z' \leq z)$ for all $z \in \mathbb{R}$. Law invariance necessarily implies that every measure $Q \in \mathfrak{M}$ in the dual representation (4.2) is absolutely continuous with respect to the reference measure P. Indeed, it follows that if $A \in \mathcal{F}$ is such that $P(A) = 0$, then the indicator function $\mathbf{1}_A$ is distributionally equivalent to the identically zero variable, and hence $\mathcal{R}(\mathbf{1}_A) = \mathcal{R}(0) = 0$. This implies that $\mathbb{E}_Q[\mathbf{1}_A] = 0$ for every $Q \in \mathfrak{M}$. Since $\mathbb{E}_Q[\mathbf{1}_A] = Q(A)$, we obtain that if $P(A) = 0$, then $Q(A) = 0$, and hence Q is absolutely continuous with respect to P.

There are basically two popular ways how the set $\mathfrak{M}$ of probability distributions is defined. In one approach a reference probability distribution P is specified and $\mathfrak{M}$ is defined as a set of probability measures in some sense close to P. A very general way of doing this is the following. Let $\mathfrak{H}$ be a nonempty set of measurable functions $h : \Omega \to \mathbb{R}$. For a probability measure Q on $(\Omega, \mathcal{F})$ consider

$$d(Q, P) := \sup_{h \in \mathfrak{H}} \left| \int_\Omega h(\omega) dQ(\omega) - \int_\Omega h(\omega) dP(\omega) \right|. \tag{4.7}$$

Formula (4.7) defines a semi-distance between probability measures Q and P. This could be semi-distance since it could happen that $d(Q, P) = 0$ for two different probability measures Q and P. In order to avoid such cases, and hence to have $d(\cdot, \cdot)$ as a distance, the set $\mathfrak{H}$ should be sufficiently large to separate different probability measures. Define then $\mathfrak{M}$ as a set of probability measures Q such that $d(Q, P) \le \varepsilon$ for some $\varepsilon > 0$. In this formulation probability measure $Q \in \mathfrak{M}$ does not need to be absolutely continuous with respect to P, in which case the corresponding functional $\mathcal{R}$ cannot be considered in the dual form (4.6). If moreover only absolutely continuous with respect to P probability measures $Q \in \mathfrak{M}$ are considered, then in a sense this construction allows to generate any set $\mathfrak{A}$ of density functions (we refer to Section 3, and specifically to Proposition 3.3, of [34] where such construction is discussed in detail).

An alternative approach is to define the set $\mathfrak{M}$ by moment constraints. That is, to consider probability measures Q on the sample space $(\Omega, \mathcal{F})$ such that $\int_\Omega \phi_i(\omega) dQ(\omega) \le b_i$, $i = 1, ..., \ell$, for some measurable functions $\phi_i : \Omega \to \mathbb{R}$. For example, it can be assumed that only say first and second order moments of the considered distributions are known (estimated). Also qualitative type properties of considered distributions can be added to the construction of $\mathfrak{M}$. In such approach there is no reference probability measure and it does not make sense to talk about law invariance and dual representations of the form (4.6). Nevertheless the respective functional $\mathcal{R}(Z)$ can be well defined by formula (4.2) and satisfy the conditions (i)-(iv). There is a delicate technical issue here. In the monotonicity condition (ii), two variables $Z, Z' : \Omega \to \mathbb{R}$ are considered such that $Z \ge Z'$. In the setting with the reference probability measure P this inequality is understood as holding almost surely with respect to P. In the setting of moment constraints it is natural to define it as that $Z(\omega) \ge Z'(\omega)$ for all $\omega \in \Omega$ (see Remark 4.1 below).

4.2 Multistage setting

Consider formulation (3.1) of multistage stochastic programming, referred to as *risk neutral* formulation. It is tempting, similar to (4.1), to specify a family $\mathfrak{M}$ of probability distributions of the data process $(\xi_1, ..., \xi_T)$, and to define the following distributionally robust analogue of (3.1)

$$\min_{\pi \in \Pi} \sup_{Q \in \mathfrak{M}} \mathbb{E}_Q \left[\sum_{t=1}^T c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t) \right] \tag{4.8}$$

with the optimization performed over a set Π of policies satisfying the feasibility constraints. However, formulation (4.8) does not explicitly specify dynamics of the considered problem in a sense which we elaborate below.

By interchanging the 'min' and 'max' operators in (4.8) we can consider problem

$$\max_{Q \in \mathfrak{M}} \inf_{\pi \in \Pi} \mathbb{E}_Q \left[\sum_{t=1}^T c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t) \right]. \tag{4.9}$$

Optimal value of problem (4.9) is always less than or equal to the optimal value of problem (4.8). Assuming that the cost functions $c_t(\cdot, \xi_t)$ of problem (4.8) are convex, e.g. linear, we have that the above minimax problem is convex-concave, and hence under mild regularity conditions optimal values of problems (4.8) and (4.9) are equal to each other. In particular, if the number of possible realizations of the data process is finite, then the set $\mathfrak{M}$ can be identified with a (closed convex) subset of Δ_m of the appropriate dimension m, and the equality of the optimal values follows provided the optimal value of (4.8) is finite (this is since the set Δ_m is bounded and hence $\mathfrak{M}$ is a compact subset of $\mathbb{R}^m$).

Suppose further that problem (4.9) has an optimal solution $\bar{Q}$. Then problem (4.8) (as well as problem (4.9)) is equivalent to the risk neutral multistage problem

$$\min_{\pi \in \Pi} \mathbb{E}_{\bar{Q}} \left[\sum_{t=1}^T c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t) \right]. \tag{4.10}$$

In turn problem (4.10) can be represented in the nested form

$$\begin{aligned} \min_{\substack{A_1 x_1 = b_1 \\ x_1 \in \mathcal{X}_1}} c_1(x_1) \quad + \quad & \mathbb{E}_{\bar{Q}|\xi_1} \Big[\min_{\substack{\boldsymbol{B}_2 \boldsymbol{x}_1 + \boldsymbol{A}_2 \boldsymbol{x}_2 = \boldsymbol{b}_2 \\ \boldsymbol{x}_2 \in \mathcal{X}_2}} c_2(\boldsymbol{x}_2, \boldsymbol{\xi}_2) + \cdots \\ & + \mathbb{E}_{\bar{Q}|\boldsymbol{\xi}_{[T-1]}} \Big[\min_{\substack{\boldsymbol{B}_T \boldsymbol{x}_{T-1} + \boldsymbol{A}_T \boldsymbol{x}_T = \boldsymbol{b}_T \\ \boldsymbol{x}_T \in \mathcal{X}_T}} c_T(\boldsymbol{x}_T, \boldsymbol{\xi}_T) \Big] \Big], \end{aligned} \tag{4.11}$$

and the corresponding dynamic programming equations can be written, with respect to the probability distribution $\bar{Q}$ of $(\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_T)$. Of course, in order to identify the 'worst' probability distribution $\bar{Q}$ one would need to solve problem (4.9) which could be even more difficult then solving the original problem. Note also that this distribution $\bar{Q}$ does not depend on realizations of the data process, and as a consequence is not adjusted to the dynamics of the problem. This motivates to consider the following approach.

For a policy $\pi = (x_1, ..., x_T(\xi_{[T]}))$ consider the respective cost $Z^\pi := \sum_{t=1}^T c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)$ with $\boldsymbol{x}_t = x_t(\boldsymbol{\xi}_{[t]})$. For $Q \in \mathfrak{M}$ we can write

$$\mathbb{E}_Q[Z^\pi] = \mathbb{E}_Q \Big[\mathbb{E}_{Q|\xi_{[1]}} \big[\cdots \mathbb{E}_{Q|\boldsymbol{\xi}_{[T-1]}} [Z^\pi] \big] \Big]. \tag{4.12}$$

It follows that

$$\sup_{Q \in \mathfrak{M}} \mathbb{E}_Q[Z^\pi] \leq \sup_{Q \in \mathfrak{M}} \mathbb{E}_Q \Big[\sup_{Q \in \mathfrak{M}} \mathbb{E}_{Q|\xi_{[1]}} \big[\cdots \sup_{Q \in \mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[T-1]}} [Z^\pi] \big] \Big]. \tag{4.13}$$

Reason for possibly strict inequality in (4.13) is that the maxima, with respect to $Q \in \mathfrak{M}$, on the right hand side of (4.13) can depend on realization of the data process (cf., [33]). The right hand side of (4.13) leads to the following *nested* formulation of the distributionally robust problem

$$\begin{array}{r}\min\limits_{\substack{A_1 x_1 = b_1 \\ x_1 \in \mathcal{X}_1}} c_1(x_1) \;\; + \;\; \sup\limits_{Q \in \mathfrak{M}} \mathbb{E}_{Q|\xi_1}\Big[\min\limits_{\substack{\boldsymbol{B}_2 \boldsymbol{x}_1 + \boldsymbol{A}_2 \boldsymbol{x}_2 = \boldsymbol{b}_2 \\ \boldsymbol{x}_2 \in \mathcal{X}_2}} c_2(\boldsymbol{x}_2, \boldsymbol{\xi}_2) + \cdots \\ + \sup\limits_{Q \in \mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[T-1]}}\Big[\min\limits_{\substack{\boldsymbol{B}_T \boldsymbol{x}_{T-1} + \boldsymbol{A}_T \boldsymbol{x}_T = \boldsymbol{b}_T \\ \boldsymbol{x}_T \in \mathcal{X}_T}} c_T(\boldsymbol{x}_T, \boldsymbol{\xi}_T)\Big]\Big].\end{array} \tag{4.14}$$

Note again that the inequality in (4.13) can be strict and problem (4.14) is not equivalent to problem (4.8).

Remark 4.1 There are delicate technical issues involved in the above distributionally robust formulations. In the risk neutral formulation (3.1) the feasibility constraints should be satisfied almost surely (with probability one) with respect to the considered probability distribution of the data process. That is, we can disregard sets of scenarios having measure zero. In case every distribution $Q \in \mathfrak{M}$ is absolutely continuous with respect to a specified reference distribution, the almost sure feasibility with respect to the reference distribution automatically implies the respective almost sure feasibility property for every $Q \in \mathfrak{M}$. When there is no reference distribution, as for example in the setting of moment constraints, it could be required that the feasibility constraints should hold almost surely with respect to every $Q \in \mathfrak{M}$.

As another issue, when the set $\mathfrak{M}$ is uncountable the 'sup' in (4.13) - (4.14) should be replaced by the conditional essential supremum. For a mathematically rigorous introduction of the essential supremum we refer to Karatzas and Shreve [17, Appendix A].

One can start with the nested formulation (4.14) for some specified family $\mathfrak{M}$ of probability distributions. It is possible to show then that there exists a family $\widehat{\mathfrak{M}}$ of probability distributions of $(\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_T)$, generally different from $\mathfrak{M}$, such that problem (4.14) can be formulated in the form (4.8) with $\mathfrak{M}$ replaced by $\widehat{\mathfrak{M}}$ (cf., [33]). For the *nested* formulation (4.14) it is possible to write dynamic programming equations with the respective cost-to-go (value) functions $V_t(x_{t-1}, \xi_{[t]})$ given by the optimal value of the problem (compare with (3.10)–(3.11))

$$\begin{array}{cl}\min\limits_{x_t \in \mathcal{X}_t} & c_t(x_t, \xi_t) + \sup\limits_{Q \in \mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[t]} = \xi_{[t]}}[V_{t+1}(x_t, \boldsymbol{\xi}_{[t+1]})] \\ \text{s.t.} & B_t x_{t-1} + A_t x_t = b_t,\end{array} \tag{4.15}$$

at stages $t = T, T-1, ..., 2$, and $V_{T+1}(\cdot, \cdot)$ omitted. At the first stage the following problem is supposed to be solved

$$\begin{array}{cl}\min\limits_{x_1 \in \mathcal{X}_1} & c_1(x_1) + \sup\limits_{Q \in \mathfrak{M}} \mathbb{E}_Q[V_2(x_1, \boldsymbol{\xi}_2)] \\ \text{s.t.} & A_1 x_1 = b_1.\end{array} \tag{4.16}$$

In fact it suffices to consider the maximum (supremum) in (4.15) with respect to marginal distributions of $\boldsymbol{\xi}_{[t+1]}$ of the probability distributions of the set $\mathfrak{M}$, and similarly in (4.16).

In particular suppose that the family $\mathfrak{M}$ is of the form

$$\mathfrak{M} := \{Q = Q_1 \times \cdots \times Q_T : Q_t \in \mathfrak{M}_t,\ t = 1, ..., T\}, \tag{4.17}$$

where $\mathfrak{M}_t$ is a family of (marginal) distributions of $\boldsymbol{\xi}_t$, $t = 1, ..., T$ (recall that ξ_1 is deterministic, therefore $\mathfrak{M}_1$ is the singleton consisting of measure δ_{ξ_1} of mass one at the point ξ_1). That is, each $Q \in \mathfrak{M}$ is the direct product of the respective marginal distributions of random vectors $\boldsymbol{\xi}_t$. One can view this as a distributionally robust analogue of the stagewise independence assumption, often it is referred to as the *rectangularity* condition. In such rectangular case the dynamic programming equations are simplified, with each cost-to-go function $V_t(x_{t-1}, \xi_t)$ depending on ξ_t, rather than the whole history $\xi_{[t]}$, and is given by the optimal value of the problem

$$\begin{array}{ll} \min\limits_{x_t \in \mathcal{X}_t} & c_t(x_t, \xi_t) + \sup\limits_{Q_{t+1} \in \mathfrak{M}_{t+1}} \mathbb{E}_{Q_{t+1}}[V_{t+1}(x_t, \boldsymbol{\xi}_{t+1})] \\ \text{s.t.} & B_t x_{t-1} + A_t x_t = b_t. \end{array} \tag{4.18}$$

This is analogous to the rectangular robust MDPs considered by Iyengar [16] and Nilim and El Ghaoui [25] (see also the recent tutorial by Mannor and Xu [23] and references therein). Yet even when the set $\mathfrak{M}$, in the nested formulation (4.14) is of the rectangular form (4.17), the corresponding set $\widehat{\mathfrak{M}}$ of formulation (4.8) generally is different from $\mathfrak{M}$ and can be quite complicated (we can refer to [33] for a further discussion of this topic).

As it was already pointed out, the maximization with respect to $Q \in \mathfrak{M}$ on the right hand side of (4.13) depends on realization of the data process. It is possible to write the nested formulation in a minimax form by considering the involved probability distributions as functions of the data process. That is, assuming that on the right hand side of (4.12), at every stage t, the probability measure Q is a function of $\xi_{[t-1]}$ and maximizing over $Q \in \mathfrak{M}$ results in the nested expression given in the right hand side of (4.13). For example, suppose that the number of scenarios is finite and hence the data process can be represented by the corresponding (finite) scenarios tree. A node of that tree at stage t represents history $\xi_{[t]}$ of the data process. The corresponding distribution in the nested formulation can be considered as a function of node $\xi_{[t-1]}$ at stage $t-1$ (cf., [36, Remark 43, page 356]).

5 Time consistency

Discussion of time consistency goes back at least to Koopmans [20] (see Bielecki, Cialenco and Piterat [9] for a recent survey and references therein). Consider a multistage stochastic programming problem of the risk neutral form (3.1). As it was pointed out before, optimization in (3.1) is performed over feasible policies. A policy is a sequence of decisions which, at every time period (stage) $t = 1, ..., T$, depend on a realization $\xi_1, ..., \xi_T$, of the data process, up to time t (this is the so-called nonaticipativity property). Such policies are implementable since at every time period the decision is made based on a historical data available to the decision maker and does not depend on future values of the data process which are unknown at the time of the decision. By solving problem (3.1) we are supposedly find an implementable policy (decision rule) which is optimal in the sense of producing the minimal expected value of the total cost $\sum_{t=1}^{T} c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)$. This optimality is related to the *total* cost and is decided *before* observing any realization of the data process. Suppose now that we are at a period (stage) $t \geq 2$ and already observed a realization of the data process up to that time. An implementable policy still gives a rule for making decisions in the following stages depending on future realizations of the data process.

However, *optimality* of the designed policy was decided from the beginning by solving problem (3.1). So a natural question is whether the designed policy is still optimal at every stage $t = 2, ..., T$ of the decision process conditional on an observed realization of the data process up to time t. In order to answer this question we need to give a precise definition of optimality at every stage $t = 2, ..., T$, given (conditional on) a realization of the data process up to the considered time period. In the risk neutral case this comes naturally by using the conditional expectations and hence writing (3.1) in the form (3.7), and consequently formulating it in the nested form (3.8). In that sense every optimal solution of (3.1) is *time consistent* in the sense that it is still optimal at every stage of the decision process with respect to the conditional expectation criterion. The respective dynamic programming equations (3.10)–(3.11) provide necessary and sufficient conditions for these optimal policies.

The situation is more involved in the distributionally robust and risk averse formulations of stochastic programs. That is, consider the following multistage problem

$$\min_{\pi \in \Pi} \mathcal{R}\left[\sum_{t=1}^{T} c_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\right], \tag{5.1}$$

where Π is the set of policies satisfying the corresponding feasibility constraints and $\mathcal{R}$ is a chosen risk functional.

- We assume that the functional $\mathcal{R} : \mathcal{Z} \to \mathbb{R}$ is *monotone*, i.e., if $Z, Z' \in \mathcal{Z}$ and $Z \geq Z'$, then $\mathcal{R}(Z) \geq \mathcal{R}(Z')$.

Naturally such monotonicity is a minimal requirement which a reasonable risk functional should satisfy. In particular if $\mathcal{R}(Z) := \sup_{Q \in \mathfrak{M}} \mathbb{E}_Q[Z]$, then (5.1) becomes the distributionally robust problem (4.8) and this functional satisfies the monotonicity condition.

5.1 Interchangeability principle and dynamic equations

Derivation of the dynamic programming equations is based on the following interchangeability principle (for the expectation operator this is discussed in Rockafellar and Wets [30, Chapter 14(F)], and for general risk functionals in [35]). Consider a function $f : \mathcal{X} \times \Omega \to \mathbb{R} \cup \{+\infty\}$ and $F(\omega) := \inf_{x \in \mathcal{X}} f(x, \omega)$. Suppose that this random variable F belongs to the considered (specified) space $\mathcal{Z}$ (in particular this implies that $F : \Omega \to \mathbb{R}$ is finite valued), and hence value

$$\mathcal{R}(F) = \mathcal{R}\big[\inf_{x \in \mathcal{X}} f(x, \omega)\big] \tag{5.2}$$

is well defined.

We address now the question whether the functional $\mathcal{R}$ and the 'inf' operator in (5.2) can be interchanged. Let $\mathfrak{X}$ be a set of mappings $\chi : \Omega \to \mathcal{X}$ such that $f_\chi \in \mathcal{Z}$, where

$f_\chi(\cdot) := f(\chi(\cdot),\cdot)$. We have that $f_\chi \geq F$ for every $\chi \in \mathfrak{X}$, and hence by monotonicity of $\mathcal{R}$ it follows that

$$\inf_{\chi\in\mathfrak{X}} \mathcal{R}(f_\chi) \geq \mathcal{R}(F). \tag{5.3}$$

Suppose for the moment that for every $\omega \in \Omega$ the minimum of $f(x,\omega)$ over $x \in \mathcal{X}$ is attained at some point $\bar\chi(\omega) \in \mathcal{X}$, i.e., $F(\cdot) = f(\bar\chi(\cdot),\cdot)$. Since $F \in \mathcal{Z}$ it follows that $\bar\chi(\cdot) \in \mathfrak{X}$. Consequently

$$\mathcal{R}(F) = \mathcal{R}(f_{\bar\chi}) \geq \inf_{\chi\in\mathfrak{X}} \mathcal{R}(f_\chi). \tag{5.4}$$

It follows from (5.3) and (5.4) that

$$\mathcal{R}\big[\inf_{x\in\mathcal{X}} f(x,\omega)\big] = \inf_{\chi\in\mathfrak{X}} \mathcal{R}\big(f(\chi(\omega),\omega)\big). \tag{5.5}$$

It also follows that:

$$\text{if } \bar\chi(\cdot) \in \arg\min_{x\in\mathcal{X}} f(x,\cdot), \text{ then } \bar\chi \in \arg\min_{\chi\in\mathfrak{X}} \mathcal{R}(f_\chi). \tag{5.6}$$

In the above derivation of the interchangeability formula (5.5) we used the following assumptions: (i) the functional $\mathcal{R}$ is monotone, (ii) the min-function $F(\cdot)$ belongs to the considered space $\mathcal{Z}$ of allowable functions, and (iii) the minimum $\min_{x\in\mathcal{X}} f(x,\omega)$ is attained for all $\omega \in \Omega$. The assumption of monotonicity is basic for derivation of the interchangeability formula. The assumption (ii) requires, in particular, verification that the min-function $F(\omega)$ is measurable. The last assumption of existence of minimizers is not essential; the proof can be pushed through by considering ε-optimal solutions.

Consider the converse of implication (5.6), which is directly related to the issue of time consistency,

$$\text{if } \bar\chi \in \arg\min_{\chi\in\mathfrak{X}} \mathcal{R}(f_\chi), \text{ then } \bar\chi(\cdot) \in \arg\min_{x\in\mathcal{X}} f(x,\cdot). \tag{5.7}$$

This implication holds if the monotonicity assumption is strengthened to the following condition of strict monotonicity. A monotone functional $\mathcal{R} : \mathcal{Z} \to \mathbb{R}$ is said to be *strictly monotone* if the following implication holds

$$Z, Z' \in \mathcal{Z},\ Z \geq Z' \text{ and } Z \neq Z',\ \text{implies that } \mathcal{R}(Z) > \mathcal{R}(Z')$$

(note that $Z \geq Z'$ and $Z \neq Z'$ means that $Z \geq Z'$ a.s. and Z is strictly greater than Z' with positive probability). The expectation operator $\mathcal{R} := \mathbb{E}$ is strictly monotone. On the other hand the Average Value-at-Risk functional $\mathcal{R} := \mathsf{AV@R}_\alpha$ is not strictly monotone for $\alpha \in (0,1)$. The $\mathsf{AV@R}_\alpha$ can be defined as (cf., Föllmer and Schied [14], Rockafellar and Uryasev [29])

$$\mathsf{AV@R}_\alpha(Z) := \inf_{t\in\mathbb{R}} \left\{ t + \alpha^{-1}\mathbb{E}_P[Z-t]_+ \right\}, \tag{5.8}$$

with $\mathcal{Z} := L_1(\Omega,\mathcal{F},P)$. In the dual form, it has representation (4.6) with

$$\mathfrak{A} = \left\{ \zeta : 0 \leq \zeta \leq 1/\alpha,\ \textstyle\int_\Omega \zeta dP = 1 \right\}.$$

Without strict monotonicity the implication (5.7) may not hold. In order to see this suppose that there exists a minimizer $\bar x(\omega)$ of $f(x,\omega)$ over $x \in \mathcal{X}$ for every $\omega \in \Omega$, i.e.,

$\bar{x}(\cdot) \in \arg\min_{x\in\mathcal{X}} f(x,\cdot)$. Consider a mapping $\hat{x} : \Omega \to \mathcal{X}$ and let $\bar{Z}(\cdot) := f(\bar{x}(\cdot),\cdot)$ and $\hat{Z}(\cdot) := f(\hat{x}(\cdot),\cdot)$. We have that $\bar{Z} \le \hat{Z}$ and hence $\mathcal{R}(\bar{Z}) \le \mathcal{R}(\hat{Z})$. That is, monotonicity of $\mathcal{R}$ ensures the implication (5.6). On the other hand, without the strict monotonicity it could happen that $\mathcal{R}(\bar{Z}) = \mathcal{R}(\hat{Z})$ although $\bar{Z} \ne \hat{Z}$. It is possible to give various examples of that type where the implication (5.7) does not hold (cf., [35]).

This can be applied already to a two stage setting. That is, consider the following problem

$$\min_{x\in\mathcal{X}} \mathcal{R}(F_x), \tag{5.9}$$

where $F_x(\omega)$ is the optimal value of the second stage problem

$$\min_{y} g(x,y,\omega) \text{ s.t. } (x,y) \in \mathcal{Y}, \tag{5.10}$$

with $\mathcal{Y} \subset \mathbb{R}^n \times \mathbb{R}^k$ and $g : \mathbb{R}^n \times \mathbb{R}^k \times \Omega \to \mathbb{R}$. It is possible to absorb the feasibility constraints $(x,y) \in \mathcal{Y}$ in problem (5.10) into the objective function by giving the infinite, $+\infty$, penalty for violating these constraints. Then in (5.11) we should define $\mathcal{R}(Z)$ for variables $Z(\omega)$ which could take the $+\infty$ value. This can be done by defining $\mathbb{E}_Q[Z] = +\infty$ when Q-probability of $Z = +\infty$ is positive.

Assuming that the functional $\mathcal{R}$ is monotone, the interchangeability principle ensures that the optimal value of problem (5.9) is equal to the optimal value of the problem

$$\min_{x,\boldsymbol{y}(\cdot)} \mathcal{R}\big(g(x,\boldsymbol{y}(\cdot),\cdot)\big) \text{ s.t. } x \in \mathcal{X},\ (x,\boldsymbol{y}(\cdot)) \in \mathcal{Y}, \tag{5.11}$$

where the notation $\boldsymbol{y}(\cdot)$ emphasizes that the minimization in (5.11) is performed over mappings $\boldsymbol{y} : \Omega \to \mathbb{R}^k$. Also if $\bar{x}$ is an optimal solution of the first stage problem (5.9) and

$$\bar{\boldsymbol{y}}(\omega) \in \arg\min_{y}\{g(\bar{x},y,\omega) : (\bar{x},y) \in \mathcal{Y}\},\ \omega \in \Omega, \tag{5.12}$$

then $(\bar{x}, \bar{\boldsymbol{y}}(\cdot))$ is an optimal solution of problem (5.11). However, unless $\mathcal{R}$ is *strictly* monotone, it could happen that the converse implication does not hold. That is, we have the following.

- If the functional $\mathcal{R}$ is not *strictly* monotone, then it could happen that the first stage problem (5.11) possesses optimal solutions which do not satisfy the second stage optimality condition (5.12), and in that sense are not time consistent.

If $\mathcal{R}(Z) := \sup_{Q\in\mathfrak{M}} \mathbb{E}_Q(Z)$ for some set $\mathfrak{M}$ of probability measures, then problem (5.11) can be written as

$$\min_{x,\boldsymbol{y}(\cdot)} \sup_{Q\in\mathfrak{M}} \mathbb{E}_Q[g(x,\boldsymbol{y}(\cdot),\cdot)] \text{ s.t. } x \in \mathcal{X},\ (x,\boldsymbol{y}(\cdot)) \in \mathcal{Y}. \tag{5.13}$$

It turns out that if the set $\mathfrak{M}$ is defined by a finite number ℓ of moment constraints and cardinality of the set Ω is greater than ℓ (in particular if the set Ω is infinite), then the respective functional $\mathcal{R}$ cannot be strictly monotone (cf., [35]).

5.1.1 Finite sample space

Suppose that the set Ω is finite, $\Omega = \{\omega_1, ..., \omega_m\}$, and let $\mathfrak{M}$ be a nonempty closed convex subset of Δ_m. As it was pointed above before here every variable $Z : \Omega \to \mathbb{R}$ can be identified with vector $z = (z_1, ..., z_m)$ in $\mathbb{R}^m$ by setting $z_i := Z(\omega_i)$, $i = 1, ..., m$. Consider the corresponding functional $\mathcal{R}(Z)$ defined in (4.2). That is (compare with equation (4.5))

$$\mathcal{R}(Z) = \sup_{\mathfrak{q}\in\mathfrak{M}} \textstyle\sum_{i=1}^m q_i z_i.$$

This functional is *strictly monotone* if and only if $\mathfrak{M} \subset \mathbb{R}^m_{++}$, i.e., all components of every vector $\mathfrak{q} \in \mathfrak{M}$ are strictly positive (e.g., [35, Proposition 2.4]).

In order to see what happens when $\mathcal{R}$ is not strictly monotone, suppose that the set $\mathfrak{M}$ has vectors with some components equal zero. Consider a point $\tilde{z} \in \mathbb{R}^m$ and let $\tilde{\mathfrak{q}} \in \mathfrak{M}$ be such that $\mathcal{R}(\tilde{z}) = \tilde{\mathfrak{q}}^\top \tilde{z}$. Note that such maximizer

$$\tilde{\mathfrak{q}} \in \arg\max\nolimits_{\mathfrak{q}\in\mathfrak{M}} \mathfrak{q}^\top \tilde{z}$$

always exists, although may be not unique, since the set $\mathfrak{M}$ is closed and bounded. Suppose further that $\tilde{q}_i = 0$ for all $i \in \mathfrak{I}$, where $\mathfrak{I} \subset \{1, ..., m\}$ is a nonempty index set. Now let $\bar{z} \in \mathbb{R}^m$ be such that $\bar{z}_i = \tilde{z}_i$ for $i \in \{1, ..., m\} \setminus \mathfrak{I}$, and $\bar{z}_i \le \tilde{z}_i$ for $i \in \mathfrak{I}$. Since $\bar{z} \le \tilde{z}$ we have that $\mathcal{R}(\bar{Z}) \le \mathcal{R}(\tilde{Z})$. On the other hand

$$\mathcal{R}(\bar{Z}) = \sup_{\mathfrak{q}\in\mathfrak{M}} \sum_{i=1}^m q_i \bar{z}_i \ge \sum_{i=1}^m \tilde{q}_i \bar{z}_i = \sum_{i \notin \mathfrak{I}}^m \tilde{q}_i \bar{z}_i = \sum_{i \notin \mathfrak{I}}^m \tilde{q}_i \tilde{z}_i = \mathcal{R}(\tilde{Z}).$$

Hence for any such $\bar{z}$ we have that $\mathcal{R}(\tilde{Z}) = \mathcal{R}(\bar{Z})$. Of course if $\bar{z}_i$ is strictly less than $\tilde{z}_i$ for some $i \in \mathfrak{I}$, then $\tilde{Z} \ne \bar{Z}$. Note that such vector $\tilde{z}$ always exists here; for example for $\tilde{z} = 0$ the set of the corresponding maximizers coincides with the set $\mathfrak{M}$ and hence there is a corresponding maximizer $\tilde{\mathfrak{q}} \in \mathfrak{M}$ having a zero component by the assumption. That is, $\mathcal{R}(\tilde{Z}) = \mathcal{R}(\bar{Z})$ when there are scenarios having probability zero in the optimal measure $\tilde{Q}$. This corresponds to existence of the so called ineffective scenarios in the sense of Rahimian, Bayraksan and Homem de Mello [28].

Now problem (5.13) can be written as

$$\min_{x,y_1,\dots,y_m} \sup_{\mathfrak{q}\in\mathfrak{M}} \sum_{i=1}^m q_i\, g(x, y_i, \omega_i) \text{ s.t. } x \in \mathcal{X},\ (x, y_i) \in \mathcal{Y},\ i = 1, ..., m. \tag{5.14}$$

Let $\bar{x}$ be an optimal solution of the first stage problem and

$$\bar{y}_i \in \arg\min_y \{g(\bar{x}, y, \omega_i) : (\bar{x}, y) \in \mathcal{Y}\},\ i = 1, ..., m, \tag{5.15}$$

be solutions of the second stage problem. Then $(\bar{x}, \bar{y}_1, ..., \bar{y}_m)$ is an optimal solution of problem (5.14). Moreover, if $\mathfrak{M} \subset \mathbb{R}^m_{++}$, i.e., the functional $\mathcal{R}$ is strictly monotone, then all optimal solutions of problem (5.14) are of that form.

On the other hand suppose that $\mathcal{R}$ is not strictly monotone. Denote $\bar{z}_i := g(\bar{x}, \bar{y}_i, \omega_i)$, where $\bar{x}$ and $\bar{y}_i$, $i = 1, ..., m$, are optimal solutions of the first and second stage problems, respectively. Consider some feasible points $\tilde{y}_i$, i.e. $(\bar{x}, \tilde{y}_i) \in \mathcal{Y}$, $i = 1, ..., m$, and denote $\tilde{z}_i := g(\bar{x}, \tilde{y}_i, \omega_i)$. Suppose that for some $\tilde{\mathfrak{q}} \in \arg\max_{\mathfrak{q} \in \mathfrak{M}} \mathfrak{q}^\top \tilde{z}$ and a nonempty index set $\mathfrak{I}$, it follows that $\tilde{q}_i = 0$ for all $i \in \mathfrak{I}$. Suppose further that $\bar{z}_i = \tilde{z}_i$ for $i \in \{1, ..., m\} \setminus \mathfrak{I}$, and $\bar{z}_i \le \tilde{z}_i$ for $i \in \mathfrak{I}$. Then by the above discussion we have that $\mathcal{R}(\bar{Z}) = \mathcal{R}(\tilde{Z})$, and hence $(\bar{x}, \tilde{y}_1, ..., \tilde{y}_m)$ is also an optimal solution of problem (5.14). That is, if it will be possible to find such $\tilde{y}_i$, not all of them being minimizers of $g(\bar{x}, \cdot, \omega_i)$, then we will have a time inconsistent optimal solution. For example if the set $\mathfrak{M}$ is defined by ℓ moment constraints, then for any $z \in \mathbb{R}^m$ there exists a corresponding maximizer $\tilde{\mathfrak{q}} \in \mathfrak{M}$ with at most $\ell + 1$ nonzero components. Hence when m is larger than $\ell + 1$, we have that there always exists maximizer $\tilde{\mathfrak{q}} \in \mathfrak{M}$ with zero components. In such cases quite often there exist optimal solutions of problem (5.14) which are not time consistent.

5.2 Time consistency of risk averse multistage problems

As it was discussed in the previous section in the risk averse setting (5.1) time inconsistency can happen already for two stage problems, and finite set Ω, if the risk functional $\mathcal{R}$ is not *strictly* monotone. Policy $(\bar{x}, \bar{\boldsymbol{y}}(\cdot))$, with $\bar{x}$ being an optimal solution of the first stage problem (5.9) and $\bar{\boldsymbol{y}}(\cdot)$ satisfying (5.12), is time consistent in the sense that $\bar{\boldsymbol{y}}(\cdot)$ is an optimal solution of the second stage problem. However, if $\mathcal{R}$ is not strictly monotone, then there may exist a mapping $\tilde{\boldsymbol{y}}(\cdot)$, satisfying the feasibility constraint $(\bar{x}, \tilde{\boldsymbol{y}}(\cdot)) \in \mathcal{Y}$, such that for random variables $\bar{Z}(\cdot) := g(\bar{x}, \bar{\boldsymbol{y}}(\cdot), \cdot)$ and $\tilde{Z}(\cdot) := g(\bar{x}, \tilde{\boldsymbol{y}}(\cdot), \cdot)$ we will have that $\mathcal{R}(\bar{Z}) = \mathcal{R}(\tilde{Z})$ while $\tilde{Z}(\omega)$ is strictly bigger than $\bar{Z}(\omega)$ for some set of $\omega \in \Omega$. Although such policy $(\bar{x}, \tilde{\boldsymbol{y}}(\cdot))$ solves the optimization problem (5.11), this inconsistent policy is inferior to the policy $(\bar{x}, \bar{\boldsymbol{y}}(\cdot))$ in the sense that for some realizations of $\omega \in \Omega$ the corresponding value $g(\bar{x}, \tilde{\boldsymbol{y}}(\omega), \omega)$ of the second stage problem is strictly bigger than $g(\bar{x}, \bar{\boldsymbol{y}}(\omega), \omega)$. Of course if problem (5.11) has a unique optimal solution $(\bar{x}, \bar{\boldsymbol{y}}(\cdot))$, then (5.12) should hold and hence this policy is time consistent. That is, inconsistent optimal solutions do not happen when problem (5.11) has *unique* optimal solution.

In multistage cases, when $T \ge 3$, there are additional issues which we are going to discuss now. The equivalence between problem (3.1), its nested formulation (3.8) and the respective dynamic equations is based on the interchageability principle and the decomposability property of the expectation operator. As it was already mentioned, in order to give a precise meaning to time consistency of a considered policy one has to specify an appropriate conditional optimality criterion at every stage of the decision process. In the risk neutral case such optimality criterion is formulated in a natural way as the corresponding conditional expectation. In the risk averse and distributionally robust settings the situation is considerably more delicate. Consider the distributionally robust problem (4.8). It can be written as risk neutral problem (4.10), provided that the corresponding worst case distribution $\bar{Q}$ does exist. Then the required conditional criteria can be thought of as the corresponding conditional expectations with respect to the distribution $\bar{Q}$. However, $\bar{Q}$ is not known and may not exist. Moreover, even if such worst case distribution does exist, it is adjusted to a particular choice of the considered optimization problem and in a sense to its optimal solution. In other words the conditional optimality is adjusted to considered optimal solutions of problem (4.8). In that way any policy could be viewed as time consistent. This does not make much sense.

For the nested formulation it is natural to use the respective *nested* conditional expectation criterion

$$\sup_{Q\in\mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[t]}}\Big[\sup_{Q\in\mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[t+1]}}\big[\cdots \sup_{Q\in\mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[T-1]}}[\,\cdot\,]\big]\Big]. \tag{5.16}$$

Note that criterion (5.16) is not the same as taking the $\sup_{Q\in\mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[t]}}[\,\cdot\,]$. The reason is the same as the one discussed in Section 4.2 for showing that formulations (4.8) and (4.14) are not equivalent. For the nested formulation it is possible to write dynamic programming equations.

- It is said that a policy $\bar\pi = \big(\bar x_1, \bar x_2(\xi_{[2]}), ..., \bar x_T(\xi_{[T]})\big)$ satisfies the dynamic programming equations if for every time period t, vector $\bar x_t = \bar x_t(\xi_{[t]})$ is an optimal solution of the corresponding optimization problem (4.15) for almost every realization $\xi_{[t]}$ of the random vector $\boldsymbol{\xi}_{[t]}$. (By "almost every" we mean that this holds with respect to every $Q \in \mathfrak{M}$ (see Remark 4.1).)

Any policy satisfying the dynamic programming equations is time consistent in the sense of the (nested) conditional criterion (5.16). And conversely any time consistent policy, in the sense of the (nested) conditional criterion, satisfies the dynamic programming equations. However, without *strict* monotonicity it could happen that there exist optimal policies which do not satisfy the dynamic programming equations and are not time consistent in the above sense (as it was already discussed, this could happen even for two stage problems and finite Ω). In the robust setting the corresponding risk functional is of max-type form which is not strictly monotone. It was shown in Bertsimas, Iancu and Parrilo [8] that in the robust setting, in certain cases, affine policies (decision rules) are optimal, but do not satisfy the dynamic programming equations and are not time consistent. For the robust inventory problem time inconsistent optimal policies were explicitly constructed in Delage and Iancu [13]. It is shown in [39] that in the distributionally robust with moment constraints inventory problems, typically there exist an infinite number of time inconsistent optimal policies.

Now consider problem (5.1). For the risk functional $\mathcal{R}$ we may consider its conditional analogues similar to the conditional expectations. In general, precise definition of such conditional risk functionals could be quite technical and is beyond the scope of this paper. Nevertheless in many interesting cases this has a natural intuitive meaning. As it was discussed in Section 4.1, risk functionals $\mathcal{R}(Z)$ which are functions of the distribution of Z (with respect to the reference distribution P), are called law invariant. For example the Average Value-at-Risk functional is law invariant. For law invariant risk functionals their conditional analogues are obtained by using the respective conditional distributions. In the distributionally robust framework, when $\mathcal{R}(Z) = \sup_{Q\in\mathfrak{M}} \mathbb{E}_Q[Z]$, the respective conditional analogue is

$$\mathcal{R}_{|\boldsymbol{\xi}_{[t]}}(Z) := \sup_{Q\in\mathfrak{M}} \mathbb{E}_{Q|\boldsymbol{\xi}_{[t]}}[Z].$$

It looks natural to use the respective conditional risk functional $\mathcal{R}_{|\boldsymbol{\xi}_{[t]}}$ in the conditional optimization criterion. Similar to the distributionally robust setting this will lead to the nested formulation (compare with (4.14))

$$\begin{aligned}\min_{\substack{A_1x_1=b_1\\ x_1\in\mathcal{X}_1}} c_1(x_1) \quad + \quad & \mathcal{R}_{|\xi_1}\Big[\min_{\substack{\boldsymbol{B}_2\boldsymbol{x}_1+\boldsymbol{A}_2\boldsymbol{x}_2=\boldsymbol{b}_2\\ \boldsymbol{x}_2\in\mathcal{X}_2}} c_2(\boldsymbol{x}_2,\boldsymbol{\xi}_2)+\cdots \\ & +\mathcal{R}_{|\boldsymbol{\xi}_{[T-1]}}\Big[\min_{\substack{\boldsymbol{B}_T\boldsymbol{x}_{T-1}+\boldsymbol{A}_T\boldsymbol{x}_T=\boldsymbol{b}_T\\ \boldsymbol{x}_T\in\mathcal{X}_T}} c_T(\boldsymbol{x}_T,\boldsymbol{\xi}_T)\Big]\Big].\end{aligned} \tag{5.17}$$

In order to verify equivalence of the formulations (5.1) and (5.17) we need the decomposability property

$$\mathcal{R}(\cdot)=\mathcal{R}_{|\xi_1}\big[\cdots\mathcal{R}_{|\boldsymbol{\xi}_{[T-1]}}(\cdot)\big], \tag{5.18}$$

similar to the expectation operator. Unfortunately for a class of the law invariant coherent risk measures, such decomposition property holds only for the expectation and max-type functionals. For example for $\mathcal{R} := \mathsf{AV@R}_\alpha$, with $\alpha\in(0,1)$, the decomposability property (5.18) does not hold and formulations (5.1) and (5.17) are not equivalent.

As it was already mentioned, in the distributionally robust setting the respective problem (4.9) and its nested analogue (4.14) are not equivalent except in some rather special cases. This motivates *to define* the functional in the nested form

$$\mathfrak{R}(\cdot):=\mathcal{R}_{|\xi_1}\big[\cdots\mathcal{R}_{|\boldsymbol{\xi}_{[T-1]}}(\cdot)\big]. \tag{5.19}$$

By replacing functional $\mathcal{R}$ in (5.1) with the nested functional $\mathfrak{R}$ we obtain optimization problem for which we can write the dynamic programming equations (compare with (3.10) and (4.15)) with value function $V_t(x_{t-1},\boldsymbol{\xi}_{[t]})$ given by the optimal value of the following problem

$$\begin{array}{ll}\min\limits_{x_t\in\mathcal{X}_t} & c_t(x_t,\xi_t)+\mathcal{R}_{|\boldsymbol{\xi}_{[t]}}[V_{t+1}(x_t,\boldsymbol{\xi}_{[t+1]})]\\ \text{s.t.} & B_tx_{t-1}+A_tx_t=b_t.\end{array} \tag{5.20}$$

Policy which is given by a solution of these dynamic programming equations is time consistent in the sense of the conditional criteria given by the respective nested functionals

$$\mathfrak{R}_{|\boldsymbol{\xi}_{[t]}}(\cdot):=\mathcal{R}_{|\boldsymbol{\xi}_{[t]}}\big[\cdots\mathcal{R}_{|\boldsymbol{\xi}_{[T-1]}}(\cdot)\big]. \tag{5.21}$$

It could be emphasized again that, unless $\mathcal{R}$ is the expectation or max-type functional, the nested functional $\mathfrak{R}_{|\boldsymbol{\xi}_{[t]}}$ is not the same as the conditional functional $\mathcal{R}_{|\boldsymbol{\xi}_{[t]}}$, and unless $\mathcal{R}$ is *strictly* monotone, problem (5.1) with $\mathcal{R}$ *replaced* by $\mathfrak{R}$ could have optimal solution policies which do not satisfy the respective dynamic programming equations and are not time consistent in the sense of the conditional criterion given by the nested functional (5.21).

6 Policy constructions

In this section we discuss various ways of how implementable policies could be constructed. Let us consider the inventory model example, discussed in Section 2, with the corresponding (risk neutral) optimization problem (2.2). It is well known that if the data process D_t is stagewise independent, then a basestock policy is optimal (e.g., Zipkin [43]). That is, there exist $x_t^*\in\mathbb{R}$, $t=1,...,T$, such that an optimal solution (optimal policy) of problem (2.2) has the following form

$$\bar{x}_t=\max\{y_t,x_t^*\},\ t=1,...,T. \tag{6.1}$$

Let us make the following observations. Here the optimal policy is given in terms of the control variables x_t and is a function of the state variables y_t only. That is, all relevant information for making decision at time t is summarised in the state variable y_t and we do not need to remember the whole history $D_{[t]}$ of the demand process. This is the general situation with the optimal control problem (3.13), provided the respective noise process is stagewise independent, an optimal policy is of the form $u_t = \pi_t(y_t)$ (this was already mentioned in Section 3.1).

In order to find the critical values x_t^* one still needs to solve the corresponding dynamic programming equations. The crucial assumption here is that the demand process is stagewise independent. In that case each value function $V_t(y_t)$ of the dynamic programming equations is a function of one variable (state variable y_t). However, this assumption of stagewise independence could be unrealistic in many applications. Often it makes sense to assume that the demand process has a Markovian structure and can be modeled as, say, an autoregressive time series of order $p \geq 1$. Then the situation changes dramatically. In that case the value function $V_t(y_t, d_{t-1}, ..., d_{t-p})$ is also a function of last p values of the demand process, i.e., it becomes a function of $p + 1$ variables. Then the corresponding optimal policies are not of the simple basestock form (6.1) and could be difficult to compute. This shows that even a relatively simple inventory model could lead to an optimization problem which would be very difficult to solve.

A popular approach in the stochastic programming literature is to approximate the underlying data process by scenario trees. Limitations of this approach were already discussed in Section 2. An argument in favor of the scenario tree approach is that typically only the first stage solution, which is deterministic, is used. Therefore at each time period the problem is recalculated after observing last available realization of the data process. Hence at every time period the decision is conditional on an observed realization and in that way an implementable policy is generated. This is the approach of rolling horizon. As it was already pointed, because of the exponential growth of the number of scenarios, one cannot branch the employed scenario tree for more than a few time periods. In some problems the optimal policies are *myopic* (a policy is said to be myopic if at every time period it suffices to solve the problem looking just one stage ahead). In such cases indeed the rolling horizon approach makes sense, since it suffices to look at just one stage ahead and hence to branch the scenario tree at the first stage only. This raises the question of a value of *multistage* formulation of stochastic programming problems. This question, of course, is problem dependent.

The above discussion illuminates limitations of the scenario trees approach. The alternative approach of dynamic programming was discussed in Section 3. However, there are several problems with the dynamic programming approach as well. While going backwards in time, the value (cost-to-go) functions $V_{t+1}(x_t, \xi_{[t+1]})$ should be kept in the computer memory in a computationally feasible way. When dimension of $(x_t, \xi_{[t+1]})$ is large this becomes practically impossible. The situation is simplified dramatically if the data process is stagewise independent. Then the expected value (cost-to-go) functions $\mathcal{V}_{t+1}(x_t) = \mathbb{E}[V_{t+1}(x_t, \boldsymbol{\xi}_{t+1})]$ do not depend on realizations of the data process. For the optimal control model (3.13) and stagewise independent noise, the respective (expected value) cost-to-go functions $V_t(y_t)$ are functions of the state variables only. A way of representing the cost-to-go functions is by using discretization of state variables. However, the number of discretization points needed for an accurate representation of the cost-to-go functions grows exponentially with increase of the dimension of state variables and this approach becomes impractical for the respective dimensions say greater than 3. This is the so-called "curse of dimensionality", which was recognized from the beginning of the dynamic programming approach (cf., Bellman [3]). Although this is not a rigorous mathematical theorem, there are compelling reasons indicating that generically, even linear, multistage stochastic problems are computationally intractable (cf., [37]). Yet the class of multistage problems is too important to be dismissed that easily. So it makes sense to talk about approximate solutions.

To some extend this can be dealt with by using approximations of the cost-to-go functions. There is a large literature on approximate dynamic programming (e.g., Bertsekas [6] and references therein). When the cost-to-go functions are convex, it could be natural to use their approximations by piecewise linear functions given by maximum of a set of cutting planes. This is an idea of the Stochastic Dual Dynamic Programming (SDDP) method, introduced in Pereira and Pinto [26]. Its origins can be traced to the nested decomposition algorithm of Birge [10]. Assuming that the data process is stagewise independent, consider the expected value functions $\mathcal{V}_{t+1}(x_t)$. Suppose that we have at hand an approximation $\hat{\mathcal{V}}_{t+1}(\cdot)$ of $\mathcal{V}_{t+1}(\cdot)$ for $t = 2, ..., T$. Then an implementable policy $\hat{\pi} = (\hat{x}_1, ..., \hat{x}_T)$ can be computed by going forward with (since B_t, A_t, b_t are functions of the data process, the solutions $\hat{x}_t$ depend on a considered realization of the data process, this is suppressed in the notation)

$$\hat{x}_t \in \arg \min_{x_t \in \mathcal{X}_t} \big\{ c_t(x_t, \xi_t) + \hat{\mathcal{V}}_{t+1}(x_t) : B_t x_{t-1} + A_t x_t = b_t \big\}, \tag{6.2}$$

$t = 1, ..., T$. Of course, this requires to solve optimization problems, given by the right hand side of (6.2), for a considered realization of the data process $\xi_1, ..., \xi_T$. If the costs $c_t(x_t, \xi_t) = c_t^\top x_t$ are linear and the approximate functions $\hat{\mathcal{V}}_{t+1}(\cdot)$ are given as maximum of linear (affine) functions, then such optimization problems can be formulated as linear programs and this approach becomes computationally feasible. Such policy $\hat{\pi}$ satisfies the feasibility constraints and is implementable. Its quality depends on how well functions $\hat{\mathcal{V}}_{t+1}(\cdot)$ approximate the respective true cost-to-go functions. In some cases this allows to solve convex multistage stochastic programs, with a moderate number of state variables, to a reasonable accuracy. The SDDP could be also adapted to deal with risk averse formulations (cf., [38]).

Another possible approach is to consider a family of policies

$$x_t(\xi_{[t]}) := \chi_t(\xi_{[t]}, \theta), \ t = 1, ..., T,$$

defined by a finite number of parameters. Here $\chi_t : \mathbb{R}^{d_2+...+d_t} \times \mathbb{R}^m \to \mathbb{R}^{n_t}$ are explicitly defined functions of the data $\xi_{[t]}$ and parameter vector θ lying in a set $\Theta \subset \mathbb{R}^m$. The multistage problem (3.1) is then reduced to the problem of finding optimal value of the parameter vector $\theta \in \Theta$. That is, problem (3.1) is reduced to the static problem

$$\begin{array}{ll} \min\limits_{\theta \in \Theta} & \mathbb{E}\left[\sum_{t=1}^T c_t(\chi_t(\boldsymbol{\xi}_{[t]}, \theta), \boldsymbol{\xi}_t) \right] \\ \text{s.t.} & \boldsymbol{B}_t \chi_{t-1}(\boldsymbol{\xi}_{[t-1]}, \theta) + \boldsymbol{A}_t \chi_t(\boldsymbol{\xi}_{[t]}, \theta) = \boldsymbol{b}_t, \ \chi_t(\boldsymbol{\xi}_{[t]}, \theta) \in \mathcal{X}_t, \ t = 1, ..., T. \end{array} \tag{6.3}$$

The expectation in (6.3) is taken with respect to the probability distribution of $(\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_T)$. Such parametric approach is often used by practitioners since it is much easier to solve the static problem (6.3) than to solve the corresponding multistage problem.

Success of the parametric approach depends on an appropriate choice of the parameterized family of policies. In the approximate dynamic programming literature this is often approached by using projected equations which are the basis for the approach, called Galerkin method in Bertsekas [6]. Note that the constraints in (6.3) should be satisfied for all (almost all) realizations of the data process $\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_T$. This indicates a serious requirement for the chosen parametric family, since the obtained problem (6.3) could be even infeasible. One possible approach is to consider policies which are linear (affine) functions of the data process. This is the approach of linear decision rules. It was shown in Kuhn, Wiesemann, and Georghiou [21] that for a certain class of linear multistage stochastic programs it is possible formulate the corresponding reduced problem as a tractable linear or semidefinite program of moderate size. Recently this was extended to a two-stage linear decision rules setting in Bodur and Luedtke [11].

The following approach was suggested in Koivu and Pennanen [19]. Suppose that we have a finite number of implementable policies $\pi^i = \big(x_1^i, x_2^i(\xi_{[2]}), ..., x_T^i(\xi_{[T]})\big)$, $i = 1, ..., m$, satisfying the feasibility constraints of the multistage problem (3.1). Consider now policies given by convex combinations of policies π^i. That is

$$\chi_t(\xi_{[t]}, \theta) := \sum_{i=1}^m \theta_i \, x_t^i(\xi_{[t]}), \; t = 1, ..., T, \tag{6.4}$$

with $\theta \in \Delta_m$. Assuming that the sets $\mathcal{X}_t$ are convex, it follows from feasibility of every policy π^i that the convex combination policy is also feasible. Consequently the multistage problem (3.1) is reduced to the static problem

$$\min_{\theta \in \Delta_m} \mathbb{E}\left[\sum_{t=1}^T c_t(\chi_t(\boldsymbol{\xi}_{[t]}, \theta), \boldsymbol{\xi}_t)\right]. \tag{6.5}$$

Note that if the cost functions $c_t(x_t, \xi_t) = c_t^\top x_t$ are linear, then the objective function in (6.5) is simply

$$\sum_{t=1}^T \sum_{i=1}^m \theta_i \, \mathbb{E}\left[c_t^\top x_t^i(\boldsymbol{\xi}_{[t]})\right] = \sum_{i=1}^m \theta_i \mathrm{val}(\pi^i),$$

where $\mathrm{val}(\pi^i) = \mathbb{E}\left[\sum_{t=1}^T c_t^\top x_t^i(\boldsymbol{\xi}_{[t]})\right]$ is the value of policy π^i. In that case the minimum in (6.5) is attained at one of the considered policies with the minimal objective value. That is, in such linear case we do not gain an improvement by considering convex combinations of the given policies.

7 Future directions, open questions

Although a considerable progress was made in recent years, we still do not have a clear understanding about computational complexity of multistage stochastic programming. From the deterministic point of view, already two-stage problems are computationally intractable - it is shown by Hanasusanto, Kuhn and Wiesemann [15] that even the approximate solution of linear two-stage stochastic programs with fixed recourse are #P-hard for a sufficiently high accuracy when the random problem data is governed by independent uniform distributions. However, it was demonstrated that randomization algorithms based on Monte Carlo sampling techniques can solve with a reasonable accuracy a large class of static (two-stage) stochastic programs. We refer to Nemirovski et al [24] for a discussion and comparison of such two popular approaches, namely the Sample Average Approximation (SAA) and Stochastic Approximation (SA) methods.

On the other hand there are compelling reasons to think that *generically* even linear multistage stochastic programming problems are computationally intractable even if a moderate accuracy is sought (cf., Shapiro and Nemirovski [37]). This of course does not mean that certain classes of multistage programs cannot be solved with a reasonable accuracy which could be sufficient for applications. This raises the question of what really is a value of trying to solve *multistage* formulation of a considered problem. Clearly this question is problem dependent. There is a need for a library of realistic examples of multistage programs which can be used for testing various approaches in numerical experiments.

The SDDP method became popular for solving linear stochastic multistage programs. In the two stage setting this is the primitive cutting plane algorithm going back to Kelley [18]. In the static (deterministic) setting bundle type cutting plane algorithms are considerably more efficient. Currently the best cutting plane algorithm is the level set bundle method of Lemaréchal, Nemirovskii and Nesterov [22]. It is not clear how more efficient bundle type algorithms could be applied to the multistage setting. The reason is that in the multistage setting the solution is in itself is random and it is not obvious where the regularization should be applied at every stage of the problem. Another question is how to deal with integer variables, this introduces another level of difficulty for solving multistage stochastic integer programs. A step in that direction was made in Zou, Ahmed and Sun [44] by introducing an extension of the SDDP, called Stochastic Dual Dynamic integer Programming (SDDiP), for solving multistage linear stochastic integer programs with binary state variables.

References

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9:203–228, 1999.

[2] E. M. L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society, Series B*, 17:173–184, 1955.

[3] R.E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[4] A. Ben-Tal, A. El Ghaoui, and A. Nemirovski. *Robust Optimizations.* Princeton University Press, 2009.

[5] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99:351–376, 2004.

[6] D.P. Bertsekas. *Dynamic Programming and Optimal Control, Vol.II.* Athena Scientific, Belmont, MA, 4th edition, 2012.

[7] D.P. Bertsekas and S.E. Shreve. *Stochastic Optimal Control, The Discrete Time Case*. Academic Press, New York, 1978.

[8] D. Bertsimas, D. Iancu, and P. Parrilo. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35:363–394, 2010.

[9] T.R. Bielecki, I. Cialenco, and M. Piterat. A survey of time consistency of dynamic risk measures and dynamic performance measures in discrete time: LM-measure perspective. *Probability, Uncertainty and Quantitative Risk*, 3:1–52, 2017.

[10] J.R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33:989–1007, 1985.

[11] M. Bodur and J.R. Luedtke. Two-stage linear decision rules for multi-stage stochastic programming. *Mathematical Programming*, First Online, 2018.

[12] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

[13] E. Delage and D. Iancu. Robust multistage decision making. *Tutorials in Operations Research, INFORMS*, pages 20–46, 2015.

[14] H. Föllmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time*. Walter de Gruyter, Berlin, 2nd edition, 2004.

[15] G.A. Hanasusanto, D. Kuhn, and W. Wiesemann. A comment on computational complexity of stochastic programming problems. *Mathematical Programming*, 159:557–569, 2015.

[16] G.N. Iyengar. Robust Dynamic Programming. *Mathematics of Operations Research*, 30:257–280, 2005.

[17] I. Karatzas and S.E. Shreve. *Methods of Mathematical Finance*. Stochastic Modelling and Applied Probability. Springer, 1998.

[18] J.E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8:703–712, 1960.

[19] M. Koivu and T. Pennanen. Galerkin methods in dynamic stochastic programming. *Optimization*, 59:339–354, 2010.

[20] T.C. Koopmans. Stationary ordinal utility and impatience. *Econometrica*, 28:287–309, 1960.

[21] D. Kuhn, W. Wiesemann, and A. Georghiou. Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130:177–209, 2011.

[22] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69:111–147, 1995.

[23] S. Mannor and H. Xu. Data-Driven Methods for Markov Decision Problems with Parameter Uncertainty. *INFORMS Tutorials in Operations Research*, 2019.

[24] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[25] A. Nilim and L. El Ghaoui. Robust control of Markov Decision Processes with uncertain transition matrices. *Operations Research*, 53:780–798, 2005.

[26] M.V.F. Pereira and L.M.V Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.

[27] M. L. Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[28] H. Rahimian, G. Bayraksan, and T. Homem de Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173:393–430, 2019.

[29] R.T. Rockafellar and S.P. Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2:21–41, 2000.

[30] R.T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, New York, 1998.

[31] A. Ruszczyński and A. Shapiro. Optimization of convex risk functions. *Mathematics of Operations Research*, 31:433–452, 2006.

[32] H. Scarf. A min-max solution of an inventory problem. In *Studies in the Mathematical Theory of Inventory and Production*, pages 201–209. Stanford University Press, 1958.

[33] A. Shapiro. Rectangular sets of probability measures. *Operations Research*, 64:528–541, 2016.

[34] A. Shapiro. Distributionally robust stochastic programming. *SIAM J. Optimization*, 27:2258–2275, 2017.

[35] A. Shapiro. Interchangeability principle and dynamic equations in risk averse stochastic programming. *Operations Research Letters*, 45:377–381, 2017.

[36] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, 2nd edition, 2014.

[37] A. Shapiro and A. Nemirovski. On complexity of stochastic programming problems. In V. Jeyakumar and A.M. Rubinov, editors, *Continuous Optimization: Current Trends and Applications: Current Trends and Applications*, pages 111–144. Springer, 2005.

[38] A. Shapiro, W. Tekaya, J.P. da Costa, and M. Pereira Soares. Risk neutral and risk averse stochastic dual dynamic programming method. *European Journal of Operational Research*, 224:375–391, 2013.

[39] A. Shapiro and L. Xin. Time inconsistency of optimal policies of distributionally robust inventory models. *Operations Research*, forthcoming.

[40] J. Žáčková. On minimax solution of stochastic linear programming problems. *Cas. Pest. Mat.*, 91:423–430, 1966.

[41] W. Wiesemann, D. Kuhn, and M. Sim. Distributionally robust convex optimization. *Operations Research*, 62:1358–1376, 2014.

[42] I. Yanikoğlu, B.L. Gorissen, and D. Hertog. A survey of adjustable robust optimization. *European Journal of Operational Research*, 277:799–813, 2019.

[43] P.H. Zipkin. *Foundation of inventory management.* McGraw-Hill, 2000.

[44] J. Zou, S. Ahmed, and X.A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175:461–502, 2019.