MULTISTAGE STOCHASTIC PROGRAMS WITH A RANDOM NUMBER OF STAGES: DYNAMIC PROGRAMMING EQUATIONS, SOLUTION METHODS, AND APPLICATION TO PORTFOLIO SELECTION

Vincent Guigues
School of Applied Mathematics, FGV
Praia de Botafogo, Rio de Janeiro, Brazil
`vincent.guigues@fgv.br`

ABSTRACT. We introduce the class of multistage stochastic optimization problems with a random number of stages. For such problems, we show how to write dynamic programming equations and how to solve these equations using the Stochastic Dual Dynamic Programming algorithm. Finally, we consider a portfolio selection problem over an optimization period of random duration. For several instances of this problem, we show the gain obtained using a policy that takes the randomness of the number of stages into account over a policy built taking a fixed number of stages (namely the maximal possible number of stages).

Keywords: Stochastic programming and Random number of stages and SDDP and Portfolio selection.

AMS subject classifications: 90C15, 90C90.

1. INTRODUCTION

Multistage Stochastic Programs (MSPs) are common in finance and in many areas of engineering, see for instance [44] and references therein. These models are useful when a sequence of decisions has to be taken over an optimization period of T stages knowing that these decisions must satisfy almost surely random constraints and induce some random costs [44, 35, 36, 41, 39]. To the best of our knowledge all MSPs considered so far in the literature have a known (finite in general) number of stages. However, for many applications the number of stages, i.e., the *real* optimization period, is not known in advance. It is easy to name a few of these applications:

- A company may want to determine optimal investments over its lifetime [31, 28, 45, 30]. In this situation, the optimization period ends when the company disappears either because it goes bankrupt, or because it is bought by another company, or because it decides to stop its activities [9]. These three stopping times, which determine the number of stages T, are indeed random.
- A fund manager can decide to stop his investments when the fund reaches a given target. This stopping time is again random, depending on the random returns of the investments and of the investment strategy.
- Multistage stochastic portfolio optimization [10, 2, 12]: an individual may invest his money in financial assets until his death or until he obtains a given amount used for covering some expense [4]. Again, both stoppping times are random.
- An hedge fund may have to deal with longevity risk [32, 8, 42, 27], with payout ratios for a given set of individuals spreading over random time windows .

1

The examples above are examples in finance but in many other areas, for instance logistics [6], power management [33, 34, 19, 15], and capacity planning and expansion [5, 11, 25], MSPs with a random optimization period could be useful, especially for long-term optimization [13] when the optimization period depends on the lifetime of individuals or companies.

It is therefore natural to consider multistage stochastic programs having a random number of stages. The study of these problems passes through two successive steps: (i) a modelling step to define a policy and (ii) an algorithmic step to build a policy, i.e., a solution method allowing us to compute decisions on any realization of the uncertainty. Guided by the fact that there exist efficient solution methods for MSPs based on dynamic programming equations (for instance Stochastic Dual Dynamic Programming (SDDP) [34] and Approximate Dynamic Programming (ADP) [40]), our goal for (i) is to write dynamic programming equations for a multistage stochastic program with a random number of stages. The paper is organized as follows. In Section 2, we define multistage stochastic optimization problems with a random number of stages and show how to write dynamic programming equations for these problems. We show in particular, that compared to the case where the number of stages is fixed, two new features appear: first we need to add an extra state variable, denoted by D_{t-1} for stage t, allowing us to know if the optimization period is already over or not and second for each stage instead of just one cost-to-go function we have two cost-to-go functions, one if the optimization period is already over (this is the null function) and another one when there remains additional stages for the optimization period. In Section 3, we write dynamic programming equations for a slightly larger class of MSPs, still having a random number of stages, but where the cost function for the last (random) stage and the cost functions for the remaining stages are taken from two sets of functions. We provide a portfolio selection model as an example of such problems.

In the case when the underlying stochastic process ξ_t neither depends on its past $(\xi_1, \ldots, \xi_{t-1})$ nor on D_t and D_t only depends on D_{t-1}, we detail in Section 4 the SDDP algorithm to solve the dynamic programming equations written in Section 2. This variant of SDDP, called SDDP-TSto, is very similar to the variants of SDDP presented in [37] where the underlying stochastic process depends on a Markov Chain (process (D_t) in our case) and a value function is used for each stage and in each state of the Markov chain. Finally, in Section 5, we consider a portfolio problem with a random optimization period and the corresponding dynamic programming equations, given in Section 3. We detail the SDDP algorithm applied to these equations and present the results of numerical tests which compare for several instances the performance of a policy that takes the randomness of the number of stages into account with the performance of a policy built taking a fixed value for the number of stages, namely the maximal possible value.

2. Writing dynamic programming equations for multistage stochastic programs with a random number of stages

Consider a risk-neutral multistage stochastic optimization problem with $T_{\max}$ known stages of form

$$(1) \qquad \inf \mathbb{E}_{\xi_2, \ldots, \xi_{T_{\max}}} \Big[\sum_{t=1}^{T_{\max}} f_t(x_t, x_{t-1}, \xi_t) \Big]$$
$$x_t \in X_t(x_{t-1}, \xi_t) \text{ a.s., } x_t \, \mathcal{F}_t\text{-measurable, } t = 1, \ldots, T_{\max},$$

where x_0 is given, ξ_1 is deterministic, $(\xi_t)_{t=2}^{T_{\max}}$ is a stochastic process, $\mathcal{F}_t$ is the sigma-algebra $\mathcal{F}_t := \sigma(\xi_j, j \leq t)$, and $X_t(x_{t-1}, \xi_t)$ is a subset of $\mathbb{R}^n$. In the objective, the expectation is computed with respect to the distribution of $\xi_2, \ldots, \xi_{T_{\max}}$. We assume that the problem above is well defined. We will come back in Section 4 to the necessary assumptions for problem (1) to be well defined and to apply SDDP as a solution method.

Our goal in this section is to define multistage stochastic optimization problems where the number of stages is not fixed ($T_{\max}$ in (1)) anymore but is stochastic, and to derive dynamic programming equations, under several assumptions, for such problems.

We will assume that

(H1) the number of stages T is a discrete random variable taking values in $\{2, \ldots, T_{\max}\}$.

The number of stages T, or stopping time, induces the Bernoulli process D_t, $t = 1, \ldots, T_{\max}$ (a "Death" process), where $D_t = \mathbb{1}_{T>t}$ is the indicator of the event $\{T > t\}$:

$$(2) \qquad D_t = \mathbb{1}_{T>t} = \begin{cases} 0 & \text{if the optimization period ended at } t \text{ or before,} \\ 1 & \text{otherwise.} \end{cases}$$

Therefore T can be written as the following function of process (D_t):

$$(3) \qquad T = \min\left\{1 \leq t \leq T_{\max} : D_t = 0\right\}.$$

Clearly, $D_t, t = 1, \ldots, T_{\max}$, are dependent random variables and the distribution of D_t given D_{t-1} is known as long as the distribution of T is known. More precisely, since we have at least 2 stages, D_1 takes value 1 with probability 1. Next, denoting $p_t = \mathbb{P}(T = t)$ and $q_t = \mathbb{P}(D_t = 0 | D_{t-1} = 1)$, we have $q_2 = \mathbb{P}(T = 2) = p_2$, and for $t \in \{2, \ldots, T_{\max}\}$ we get

$$p_t = \mathbb{P}(T = t) = \mathbb{P}(D_2 = 1; D_3 = 1; \ldots; D_{t-1} = 1; D_t = 0) = q_t \prod_{k=2}^{t-1}(1 - q_k).$$

Therefore transition probabilities $q_t, t = 2, 3, \ldots, T_{\max}$, can be computed using the recurrence

$$(4) \qquad q_t = \frac{p_t}{\displaystyle\prod_{k=2}^{t-1}(1 - q_k)}, t = 3, \ldots, T_{\max},$$

starting from $q_2 = p_2$ (note that $q_{T_{\max}} = 1$).

By definition of D_t, we also have $\mathbb{P}(D_t = 0 | D_{t-1} = 0) = 1$ or equivalently $\mathbb{P}(D_t = 1 | D_{t-1} = 0) = 0$.

We represent on the top left plot of Figure 1 the scenario tree of the realizations of $D_1, D_2, \ldots, D_{T_{\max}}$ (for an example where $T_{\max} = 5$), as well as the transition probabilities between the nodes of this scenario tree. In this scenario tree, realizations are indicated at the nodes of the tree while transition probabilities between two nodes are given above the arrow linking these nodes. Observe that for a node with label 0 all future transition probabilities are 1 and all descendant nodes have label 0.

In the case when the number of stages is stochastic, the decision x_t for stage t is not only a function of the history $\xi_{[t]} = (\xi_1, \xi_2, \ldots, \xi_t)$ of process (ξ_t), as in (1), but also depends on the history of process (D_t). Therefore, we come to the following definition of a risk-neutral multistage stochastic optimization problem with a random number T of stages:

$$(5) \qquad \inf \mathbb{E}_{\xi_2, \ldots, \xi_{T_{\max}}, D_2, \ldots, D_{T_{\max}}}\left[\sum_{t=1}^{T} f_t(x_t, x_{t-1}, \xi_t)\right]$$
$$x_t \in X_t(x_{t-1}, \xi_t) \text{ a.s., } x_t \ \overline{\mathcal{F}}_t\text{-measurable, } t = 1, \ldots, T_{\max},$$

where $\overline{\mathcal{F}}_t$ is the sigma-algebra

$$(6) \qquad \overline{\mathcal{F}}_t = \sigma(\xi_j, D_j, j \leq t)$$

and where T is the function of (D_t) given by (3). Note that in the objective of (5) the expectation is computed with respect to the distribution of $\xi_2, \ldots, \xi_{T_{\max}}, D_2, \ldots, D_{T_{\max}}$. Plugging (3) into (5),
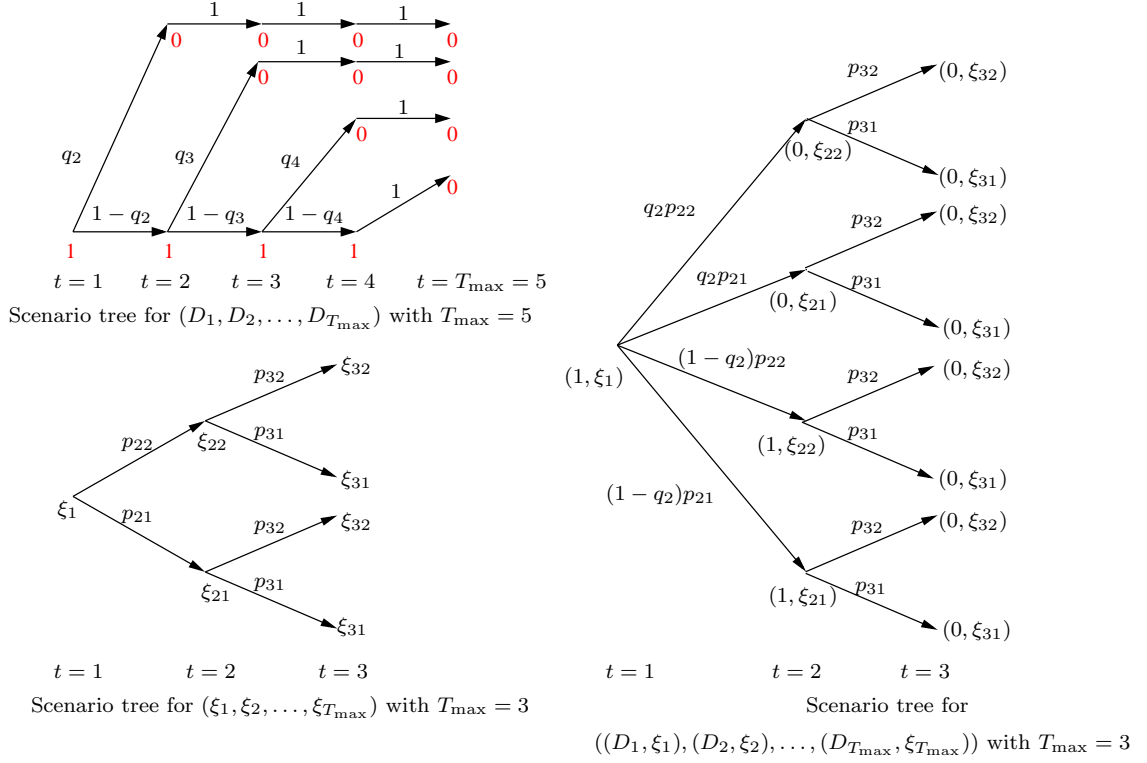
FIGURE 1. Scenario trees (assuming that ξ_t does not depend on $(\xi_{[t-1]}, D_t)$).

problem (5) can be written

$$
\begin{gathered}
(7) \qquad \inf \mathbb{E}_{\xi_2,\ldots,\xi_{T_{\max}},D_2,\ldots,D_{T_{\max}}}\left[\sum_{1\leq t\leq\min\{1\leq\tau\leq T_{\max}:D_\tau=0\}} f_t(x_t, x_{t-1}, \xi_t)\right] \\
x_t \in X_t(x_{t-1}, \xi_t) \text{ a.s., } x_t\ \overline{\mathcal{F}}_t\text{-measurable}, t = 1,\ldots,T_{\max}.
\end{gathered}
$$

To write dynamic programming equations for (7) we now define the state vectors. The state vector at stage t is given by x_{t-1} (decision taken at the previous stage) and the relevant history of processes (ξ_t) and (D_t). Though all the history $D_{[t-1]} = (D_1,\ldots,D_{t-1})$ of process (D_t) until stage $t-1$ may be necessary, we argue that it is enough to put in the state vector for stage t past value D_{t-1} of (D_t). Indeed,

- if $D_{t-1} = 1$ then the whole history of (D_t) until $t-1$ is known: we know that $D_j = 1$ for $1 \leq j \leq t-1$;
- on the other hand, if $D_{t-1} = 0$ then whatever the history of (D_t) until $t-1$, we know that the cost function is null for stage t because the optimization period ended at $t-1$ or before.

Consequently the state vector at stage t is $(x_{t-1}, \xi_{[t-1]}, D_{t-1})$ and we introduce for each stage $t = 2,\ldots,T_{\max}$, two functions:

- $\mathfrak{Q}_t$ such that $\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}, \xi_t, D_t)$ is the optimal mean cost from t on starting at t from state $(x_{t-1}, \xi_{[t-1]}, D_{t-1})$ and knowing the values ξ_t and D_t of processes (ξ_t) and (D_t) at t;

- $\mathcal{Q}_t$ given by

(8) $$\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}) = \mathbb{E}_{\xi_t, D_t}\Big[\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}, \xi_t, D_t)|D_{t-1}, \xi_{[t-1]}\Big],$$

i.e., $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1})$ is the optimal mean cost from t on starting at t from state $(x_{t-1}, \xi_{[t-1]}, D_{t-1})$.

We also set $\mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, \xi_{[T_{\max}]}, D_{T_{\max}}) \equiv 0$. With these definitions, clearly for $t = 2, \ldots, T_{\max}$, we have

(9) $$\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, 0) = 0.$$

Next, for $t = 2, \ldots, T_{\max}$, functions $\mathcal{Q}_t(\cdot, \cdot, 1)$ satisfy the recurrence

(10) $$\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, 1) = \mathbb{E}_{\xi_t, D_t}\Big[\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, 1, \xi_t, D_t)|D_{t-1} = 1, \xi_{[t-1]}\Big]$$

where

(11) $$\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, 1, \xi_t, 0) = \begin{cases} \inf_{x_t}\ f_t(x_t, x_{t-1}, \xi_t) \\ x_t \in X_t(x_{t-1}, \xi_t), \end{cases}$$

and

(12) $$\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, 1, \xi_t, 1) = \begin{cases} \inf_{x_t}\ f_t(x_t, x_{t-1}, \xi_t) + \mathcal{Q}_{t+1}(x_t, \xi_{[t-1]}, \xi_t, 1) \\ x_t \in X_t(x_{t-1}, \xi_t). \end{cases}$$

The reasons for equations (10)-(12) are clear:

- $\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, 1, \xi_t, 0)$ is the optimal mean cost from stage t on, knowing that the optimization period ends at t and that ξ_t is the value of process (ξ_t) at stage t. Therefore it is obtained minimizing the immediate stage t cost while satisfying the constraints for stage t.
- $\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, 1, \xi_t, 1)$ is the optimal mean cost from stage t on, knowing that the optimization period continues after t and that ξ_t is the value of process (ξ_t) at stage t. Therefore it is obtained minimizing the immediate stage t cost plus the future optimal mean cost (which is $\mathcal{Q}_{t+1}(x_t, \xi_{[t]}, D_t) = \mathcal{Q}_{t+1}(x_t, \xi_{[t-1]}, \xi_t, 1)$ since $D_t = 1$) while satisfying the constraints for stage t.

We observe that equations (9)-(12) can be written under the following compact form: for $t = 2, \ldots, T_{\max}$,

(13) $$\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}) = \mathbb{E}_{\xi_t, D_t}\Big[\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}, \xi_t, D_t)|D_{t-1}, \xi_{[t-1]}\Big]$$

where

(14) $$\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}, \xi_t, D_t) = \begin{cases} \inf_{x_t}\ D_{t-1}f_t(x_t, x_{t-1}, \xi_t) + \mathcal{Q}_{t+1}(x_t, \xi_{[t]}, D_t) \\ x_t \in X_t(x_{t-1}, \xi_t). \end{cases}$$

Setting $D_0 = 1$, recalling that $D_1 = 1$ and that $\overline{\mathcal{F}}_t$ is given by (6), it is straighforwardly seen that the optimal value of (7) can be expressed as

(15) $$\begin{array}{c} \inf_{x_1}\ D_0 f_1(x_1, x_0, \xi_1) + \mathcal{Q}_2(x_1, \xi_1, D_1) \\ x_1 \in X_1(x_0, \xi_1), \end{array}$$

and that (13)-(14) are dynamic programming equations for the problem

(16) $$\inf \mathbb{E}_{\xi_2, \ldots, \xi_{T_{\max}}, D_2, \ldots, D_{T_{\max}}}\Big[\sum_{t=1}^{T_{\max}} D_{t-1} f_t(x_t, x_{t-1}, \xi_t)\Big]$$
$$x_t \in X_t(x_{t-1}, \xi_t) \text{ a.s., } x_t\ \overline{\mathcal{F}}_t\text{-measurable, } t = 1, \ldots, T_{\max},$$

which is an equivalent reformulation of (7).

The impact of the randomness of T on the dynamic programming equations is clear from reformulation (16) of problem (7). In this reformulation, the number of stages is fixed and known: it is the maximal possible number of stages $T_{\max}$ for T. Therefore, it takes the form of a "usual" multistage stochastic optimization problem where random variable T was replaced by the interstage dependent random process (D_t) and the cost function f_t at stage t was replaced by the random cost function $D_{t-1}f_t$. Indeed, when the optimization period ended at $t-1$ or before, the cost function is null for stage t or equivalently can be expressed as $D_{t-1}f_t$ since $D_{t-1} = 0$ in this case. On the other hand, if the optimization period did not end at $t-1$ then $D_{t-1} = 1$ and again the cost function for stage t can be expressed as $D_{t-1}f_t$ $(= f_t$ in this case).

Note that in these equations, (ξ_t, D_t) can depend on $\xi_{[t-1]}$. Clearly D_t depends on D_{t-1} but (ξ_t, D_t) can be independent on $\xi_{[t-1]}$. In this situation, $\xi_{[t-1]}$ is not needed in the state vector at t and the dynamic programming equations simplify as follows: $\mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, D_{T_{\max}}) \equiv 0$ and for $t = 2, \ldots, T_{\max}$, we have

$$(17) \qquad \mathcal{Q}_t(x_{t-1}, D_{t-1}) = \mathbb{E}_{\xi_t, D_t}\left[\mathfrak{Q}_t(x_{t-1}, D_{t-1}, \xi_t, D_t)|D_{t-1}\right]$$

where

$$(18) \qquad \mathfrak{Q}_t(x_{t-1}, D_{t-1}, \xi_t, D_t) = \begin{cases} \inf_{x_t} \ D_{t-1}f_t(x_t, x_{t-1}, \xi_t) + \mathcal{Q}_{t+1}(x_t, D_t) \\ x_t \in X_t(x_{t-1}, \xi_t). \end{cases}$$

Finally, let us consider the case when ξ_t does not depend on $(\xi_{[t-1]}, D_t)$ and D_t only depends on D_{t-1}. In this setting, (D_t) is an inhomogeneous Markov chain with two states: an absorbing state corresponding to the case when the optimization period is over and a second state where the optimization period is still not over. We assume that the distribution of ξ_t is discrete with finite support $\{\xi_{t1}, \ldots, \xi_{tM_t}\}$ with $p_{tj} = \mathbb{P}(\xi_t = \xi_{tj})$. The scenario trees for $(\xi_1, \ldots, \xi_{T_{\max}})$ and $((D_1, \xi_1), (D_2, \xi_2), \ldots, (D_{T_{\max}}, \xi_{T_{\max}}))$ (nodes and transition probabilities) are represented in Figure 1 (right and bottom left plots) on an example where $T_{\max} = 3$ and where ξ_t has two possible realizations for all $t = 2, \ldots, T_{\max}$.

With these assumptions dynamic programming equations (17)-(18) can be written as follows: $\mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, D_{T_{\max}}) \equiv 0$,

$$(19) \qquad \mathcal{Q}_t(x_{t-1}, 0) = 0, \ t = 2, \ldots, T_{\max},$$

and

$$(20) \qquad \mathcal{Q}_t(x_{t-1}, 1) = (1 - q_t)\sum_{j=1}^{M_t} p_{tj}\mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 1) + q_t\sum_{j=1}^{M_t} p_{tj}\mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 0),$$

where q_t is given by (4),

$$(21) \qquad \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 1) = \begin{cases} \inf_{x_t} \ f_t(x_t, x_{t-1}, \xi_{tj}) + \mathcal{Q}_{t+1}(x_t, 1) \\ x_t \in X_t(x_{t-1}, \xi_{tj}), \end{cases}$$

and

$$(22) \qquad \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 0) = \begin{cases} \inf_{x_t} \ f_t(x_t, x_{t-1}, \xi_{tj}) \\ x_t \in X_t(x_{t-1}, \xi_{tj}). \end{cases}$$

Remark 2.1. *Observe that the dynamic programming equations above correspond to a model that minimizes the expected cost with respect to the distribution of $(\xi_1, \xi_2, \ldots, \xi_{T_{\max}}, D_1, D_2, \ldots, D_{T_{\max}})$. From the Law of Large Numbers, this model is useful when the corresponding policy is repeatedly applied by individuals sharing the same distribution of the number of stages T. An example would be a group of companies sharing the same distribution for their lifetime, see* [9] *and Section 5.2.*

3. Dynamic programming equations for more general models

3.1. **Writing dynamic programming equations.** In the previous section, we considered models where the cost functions for stages $t \leq T$ are taken from the collection of functions (f_t), namely f_t for stage t as long as $t \leq T$. It is possible to write dynamic programming equations for more general risk-neutral stochastic programming models having a random number T of stages where the cost function for stage t is taken from collection of functions $(f_{t,1}, f_{t,2}, f_{t,3}, \ldots,)$ as follows: there is some random variable T_1 such that for $1 \leq t \leq \min(T_1, T)$ the cost function is $f_{t,1}$; next there is some random variable T_2 such that for $\min(T_1, T) + 1 \leq t \leq \min(T_1 + T_2, T)$ the cost function is $f_{t,2}$, and so on...

As a special case, assume that $T_1 = T - 1$: for $t = 1, \ldots, T-1$, the cost function is $f_t(x_t, x_{t-1}, \xi_t)$ and for $t = T$ the cost function is $\overline{f}_t(x_t, x_{t-1}, \xi_t)$. In this situation, recalling definition (2) of D_t, the cost function for stage t can be written

(23) $$D_t f_t(x_t, x_{t-1}, \xi_t) + (D_{t-1} - D_t)\overline{f}_t(x_t, x_{t-1}, \xi_t).$$

Indeed,

- if $t < T$ we have $D_{t-1} = D_t = 1$ and $D_t f_t(x_t, x_{t-1}, \xi_t) + (D_{t-1} - D_t)\overline{f}_t(x_t, x_{t-1}, \xi_t) = f_t(x_t, x_{t-1}, \xi_t)$;
- if $t = T$ we have $D_{t-1} = 1, D_t = 0$, and $D_t f_t(x_t, x_{t-1}, \xi_t) + (D_{t-1} - D_t)\overline{f}_t(x_t, x_{t-1}, \xi_t) = \overline{f}_t(x_t, x_{t-1}, \xi_t)$;
- if $t > T$ no costs are incurred, we have $D_{t-1} = D_t = 0$ and $D_t f_t(x_t, x_{t-1}, \xi_t) + (D_{t-1} - D_t)\overline{f}_t(x_t, x_{t-1}, \xi_t) = 0$;

as required. In the next section we present a simple portfolio problem modelled by a MSP of this type. Therefore, for the "special case" we are dealing with now, with cost function (23) for stage t, we obtain the multistage stochastic program

(24) $$\inf \mathbb{E}_{\xi_2, \ldots, \xi_{T_{\max}}, D_2, \ldots, D_{T_{\max}}} \left[\sum_{t=1}^{T_{\max}} D_t f_t(x_t, x_{t-1}, \xi_t) + (D_{t-1} - D_t)\overline{f}_t(x_t, x_{t-1}, \xi_t) \right]$$
$$x_t \in X_t(x_{t-1}, \xi_t) \text{ a.s.}, x_t \, \overline{\mathcal{F}}_t\text{-measurable}, t = 1, \ldots, T_{\max},$$

where $\overline{\mathcal{F}}_t$ is the sigma-algebra given by (6). Observe that when

$$\overline{f}_t(x_t, x_{t-1}, \xi_t) = f_t(x_t, x_{t-1}, \xi_t),$$

we are back to the stochastic programs considered in the previous section, i.e., problem (24) becomes problem (16). Clearly, (24) is obtained replacing in (16) the cost function $D_t f_t(x_t, x_{t-1}, \xi_t)$ for stage t by (23). Therefore dynamic programming equations for (24) are obtained updating correspondingly the cost functions in dynamic programming equations (13)-(14) for (16), i.e., the dynamic programming equations for (24) are: $\mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, \xi_{[T_{\max}]}, D_{T_{\max}}) \equiv 0$ and for $t = 2, \ldots, T_{\max}$,

(25) $$\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}) = \mathbb{E}_{\xi_t, D_t} \left[\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}, \xi_t, D_t) | D_{t-1}, \xi_{[t-1]} \right]$$

where
(26)
$$\mathfrak{Q}_t(x_{t-1}, \xi_{[t-1]}, D_{t-1}, \xi_t, D_t) = \begin{cases} \inf_{x_t} \, D_t f_t(x_t, x_{t-1}, \xi_t) + (D_{t-1} - D_t)\overline{f}_t(x_t, x_{t-1}, \xi_t) + \mathcal{Q}_{t+1}(x_t, \xi_{[t]}, D_t) \\ x_t \in X_t(x_{t-1}, \xi_t). \end{cases}$$

Now assume that ξ_t does not depend on $(\xi_{[t-1]}, D_t)$, D_t only depends on D_{t-1}, and the distribution of ξ_t is discrete with finite support $\{\xi_{t1} \ldots, \xi_{tM_t}\}$. Denoting $p_{tj} = \mathbb{P}(\xi_t = \xi_{tj})$, equations (25)-(26) simplify as follows: $\mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, 0) = \mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, 1) \equiv 0$, for $t = 2, \ldots, T_{\max}$, $\mathcal{Q}_t(x_{t-1}, 0) \equiv 0$,

	$x_t(i)$	Dollar value of risky asset i at end of t
Decision variables	$y_t(i)$	Amount (in dollars) of risky asset i sold at end of t
	$z_t(i)$	Amount (in dollars) of risky asset i bought at end of t
	$x_t(n+1)$	Amount (in dollars) held in cash at end of t
Stochastic parameters	$\xi_t(i)$	Return of risky asset i for stage t
	$\xi_t(n+1)$	Cash return for stage t
	$u(i)$	Maximal proportion of capital invested in risky asset i
Deterministic parameters	$\eta_t(i)$	Selling transaction cost for risky asset i stage t
	$\nu_t(i)$	Buying transaction cost for risky asset i stage t
	x_0	Initial portfolio

TABLE 1. Parameters and decision variables for the portfolio problem.

and for $t = 2, \ldots, T_{\max}$, we have

$$(27) \qquad \mathcal{Q}_t(x_{t-1}, 1) = (1 - q_t) \sum_{j=1}^{M_t} p_{tj} \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 1) + q_t \sum_{j=1}^{M_t} p_{tj} \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 0),$$

where q_t is given by (4),

$$(28) \qquad \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 1) = \begin{cases} \inf_{x_t} \ f_t(x_t, x_{t-1}, \xi_{tj}) + \mathcal{Q}_{t+1}(x_t, 1) \\ x_t \in X_t(x_{t-1}, \xi_{tj}), \end{cases}$$

and

$$(29) \qquad \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 0) = \begin{cases} \inf_{x_t} \ \overline{f}_t(x_t, x_{t-1}, \xi_{tj}) \\ x_t \in X_t(x_{t-1}, \xi_{tj}). \end{cases}$$

3.2. **Example: a simple portfolio problem.** We consider the portfolio selection problem with direct transaction costs given in [22]. When the number of stages is random we obtain a problem from the class of problems introduced in the previous Section 3.1. We first recall the dynamic programming equations for this model when the number of stages $T_{\max}$ is fixed and known. We refer to Table 1 for the list of corresponding parameters and decision variables. Let $x_t(i)$ be the dollar value of asset $i = 1, \ldots, n+1$, at the end of stage $t = 1, \ldots, T_{\max}$, where asset $n+1$ is cash; let $\xi_t(i)$ be the return of asset i at t; let $y_t(i)$ be the amount of asset i sold at the end of t; let $z_t(i)$ be the amount of asset i bought at the end of t with $\eta_t(i) > 0$ and $\nu_t(i) > 0$ the respective proportional selling and buying transaction costs at t. Each component $x_0(i), i = 1, \ldots, n+1$, of x_0 is known. The budget available at the beginning of the investment period is $\sum_{i=1}^{n+1} \xi_1(i)x_0(i)$ and $u(i)$ represents the maximal proportion of capital that can be invested in asset i. For $t = 1, \ldots, T_{\max}$, given a portfolio $x_{t-1} = (x_{t-1}(1), \ldots, x_{t-1}(n), x_{t-1}(n+1))$ and ξ_t, we define the set $X_t(x_{t-1}, \xi_t)$ as the set of portfolios $(x_t, y_t, z_t) \in \mathbb{R}^{n+1} \times \mathbb{R}^n \times \mathbb{R}^n$ satisfying

$$
\begin{aligned}
&x_t(n+1) = \xi_t(n+1)x_{t-1}(n+1) + \sum_{i=1}^{n} \Big((1 - \eta_t(i))y_t(i) - (1 + \nu_t(i))z_t(i) \Big), \\
&x_t(i) = \xi_t(i)x_{t-1}(i) - y_t(i) + z_t(i), i = 1, \ldots, n, \\
&x_t(i) \leq u(i) \sum_{j=1}^{n+1} \xi_t(j)x_{t-1}(j), i = 1, \ldots, n, \\
&x_t(i) \geq 0, y_t(i) \geq 0, z_t(i) \geq 0, i = 1, \ldots, n.
\end{aligned}
$$

With this notation, the following dynamic programming equations of a risk-neutral portfolio model can be written: for $t = T_{\max}$, we solve the problem

(30)

$$\mathfrak{Q}_{T_{\max}}(x_{T_{\max}-1}, \xi_{T_{\max}}) = \begin{cases} \inf \; \overline{f}_{T_{\max}}(x_{T_{\max}}, x_{T_{\max}-1}, \xi_{T_{\max}}) := -\mathbb{E}\Big[\sum_{i=1}^{n+1} \xi_{T_{\max}+1}(i) x_{T_{\max}}(i)\Big] \\ X_{T_{\max}} \in X_{T_{\max}}(x_{T_{\max}-1}, \xi_{T_{\max}}), \end{cases}$$

while at stage $t = T_{\max} - 1, \ldots, 1$, we solve

(31)
$$\mathfrak{Q}_t(x_{t-1}, \xi_t) = \begin{cases} \inf \; Q_{t+1}(x_t) \\ x_t \in X_t(x_{t-1}, \xi_t), \end{cases}$$

where

(32)
$$\mathcal{Q}_t(x_{t-1}) = \mathbb{E}_{\xi_t}[\mathfrak{Q}_t(x_{t-1}, \xi_t)], \; t = 2, \ldots, T_{\max}.$$

Now for $t = 1, \ldots, T_{\max}$, define

(33)
$$f_t(x_t, x_{t-1}, \xi_t) \equiv 0 \text{ and } \overline{f}_t(x_t, x_{t-1}, \xi_t) = -\mathbb{E}\Big[\sum_{i=1}^{n+1} \xi_{t+1}(i) x_t(i)\Big].$$

Since the number of stages is fixed to $T_{\max}$ then $D_t = 1, t = 1, \ldots, T_{\max} - 1, D_{T_{\max}} = 0$ almost surely, and the porfolio problem we have just described is of form (24) with $f_t, \overline{f}_t$ as in (33) and $D_t = 1, t = 1, \ldots, T_{\max} - 1, D_{T_{\max}} = 0$ almost surely, i.e., we obtain the portfolio problem

(34)
$$\begin{array}{l} \inf \mathbb{E}_{\xi_2, \ldots, \xi_{T_{\max}}}\Big[\overline{f}_{T_{\max}}(x_{T_{\max}}, x_{T_{\max}-1}, \xi_{T_{\max}})\Big] \\ x_t \in X_t(x_{t-1}, \xi_t) \text{ a.s., } x_t \; \mathcal{F}_t\text{-measurable, } t = 1, \ldots, T_{\max}. \end{array}$$

With this model, we minimize the expected loss of the portfolio (or equivalently maximize the mean income) taking into account the transaction costs, non-negativity constraints, and bounds imposed on the different securities.

Now assume that the number of stages is random with discrete distribution on $\{2, \ldots, T_{\max}\}$ and define D_t by (2). We obtain the portfolio problem (24) with $f_t, \overline{f}_t$ as in (33). If ξ_t does not depend on $(\xi_{[t-1]}, D_t)$, D_t only depends on D_{t-1}, and the distribution of ξ_t is discrete with finite support $\{\xi_{t1} \ldots, \xi_{tM_t}\}$, denoting $p_{tj} = \mathbb{P}(\xi_t = \xi_{tj})$, we can write the following dynamic programming equations for the corresponding portfolio problem: $\mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, 0) = \mathcal{Q}_{T_{\max}+1}(x_{T_{\max}}, 1) \equiv 0$, for $t = 2, \ldots, T_{\max}, \mathcal{Q}_t(x_{t-1}, 0) \equiv 0$ and for $t = 2, \ldots, T_{\max}$, we have

(35)
$$\mathcal{Q}_t(x_{t-1}, 1) = (1 - q_t) \sum_{j=1}^{M_t} p_{tj} \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 1) + q_t \sum_{j=1}^{M_t} p_{tj} \mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 0),$$

where q_t is given by (4),

(36)
$$\mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 1) = \begin{cases} \inf_{x_t} \; \mathcal{Q}_{t+1}(x_t, 1) \\ x_t \in X_t(x_{t-1}, \xi_{tj}), \end{cases}$$

and

(37)
$$\mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 0) = \begin{cases} \inf_{x_t} \; -\mathbb{E}[\xi_{t+1}^T x_t] \\ x_t \in X_t(x_{t-1}, \xi_{tj}). \end{cases}$$

In the case when the number of stages is $T_{\max}$ (deterministic) then $q_t = 0$ for $t = 2, \ldots, T_{\max} - 1$, $q_{T_{\max}} = 1$ and dynamic programming equations (35), (36), (37) become, as expected, (30), (31), (32) (with the notation $\mathcal{Q}_t(x_{t-1})$ instead of $\mathcal{Q}_t(x_{t-1}, 1)$, $\mathfrak{Q}_t(x_{t-1}, \xi_{tj})$ instead of $\mathfrak{Q}_t(x_{t-1}, 1, \xi_{tj}, 1)$, and $\mathfrak{Q}_{T_{\max}}(x_{T_{\max}-1}, \xi_{T_{\max}})$ instead of $\mathfrak{Q}_{T_{\max}}(x_{T_{\max}-1}, 1, \xi_{T_{\max}}, 0)$).

4. SDDP for multistage stochastic risk-neutral programs with a random number of stages

4.1. **Assumptions.** Consider optimization problem (16) where ξ_t does not depend on $(\xi_{[t-1]}, D_t)$ and D_t only depends on D_{t-1}. We assume that the distributions of T and ξ_t are discrete: the support of T is $\{2, \ldots, T_{\max}\}$ and the support of ξ_t is $\Theta_t = \{\xi_{t1}, \ldots, \xi_{tM_t}\}$ with $p_{ti} = \mathbb{P}(\xi_t = \xi_{ti}) > 0, i = 1, \ldots, M_t$. In this context, equations (19), (20), (21), (22) are the dynamic programming equations for (16). We can now apply Stochastic Dual Dynamic Programming (SDDP, [34]), to solve these dynamic programming equations as long as recourse functions $\mathcal{Q}_t(\cdot, 1)$ are convex. SDDP has been used to solve many real-life problems and several extensions of the method have been considered such as DOASA [38], CUPPS [7], ReSA [23], AND [3], risk-averse ([20, 21, 26, 37, 43, 44]) or inexact ([18]) variants; see also [16, 24] for adaptations to interstage dependent processes and [46] for extensions for integer stochastic programs.

SDDP builds approximations for the cost-to-go functions which take the form of a maximum of affine functions.

To ensure convexity of functions $\mathcal{Q}_t(\cdot, 1)$, we need convexity of functions $f_t(\cdot, \cdot, \xi_t)$ and of multifunctions $X_t(\cdot, \xi_t)$ for almost every ξ_t. We will consider two settings: linear and nonlinear programs.

Linear problems. In this setting, $f_t(x_t, x_{t-1}, \xi_t) = c_t^T x_t$ is linear,

$$(38) \qquad X_t(x_{t-1}, \xi_t) := \{x_t \in \mathbb{R}^n : A_t x_t + B_t x_{t-1} = b_t,\ x_t \geq 0\},$$

and random vector ξ_t corresponds to the concatenation of the elements in random matrices A_t, B_t which have a known finite number of rows and random vectors b_t, c_t. We assume:

(H2-L) The set $X_1(x_0, \xi_1)$ is nonempty and bounded and for every $x_1 \in X_1(x_0, \xi_1)$, for every $t = 2, \ldots, T$, for every realization $\tilde{\xi}_2, \ldots, \tilde{\xi}_t$ of $\xi_2, \ldots, \xi_t$, for every $x_\tau \in X_\tau(x_{\tau-1}, \tilde{\xi}_\tau), \tau = 2, \ldots, t-1$, the set $X_t(x_{t-1}, \tilde{\xi}_t)$ is nonempty and bounded.

Nonlinear problems. In this case,

$$(39) \qquad X_t(x_{t-1}, \xi_t) = \{x_t \in \mathbb{R}^n : x_t \in \mathcal{X}_t,\ g_t(x_t, x_{t-1}, \xi_t) \leq 0,\ A_t x_t + B_t x_{t-1} = b_t\},$$

and ξ_t contains in particular the random elements in matrices A_t, B_t, and vector b_t. Of course, as a special case (and as is often the case in applications), the nonlinear problems we are interested in can have nonlinear cost and constraint functions for stage t that do not depend on x_{t-1}, namely of form $f_t(x_t, \xi_t)$ and $g_t(x_t, \xi_t)$.

We assume that for $t = 1, \ldots, T$, there exists $\varepsilon_t > 0$ such that:

(H2-NL)-(a) $\mathcal{X}_t$ is nonempty, convex, and compact.

(H2-NL)-(b) For every $j = 1, \ldots, M_t$, the function $f_t(\cdot, \cdot, \xi_{tj})$ is convex, lower semicontinuous, and finite on $\mathcal{X}_t \times \mathcal{X}_{t-1}^{\varepsilon_t}$ where $\mathcal{X}_{t-1}^{\varepsilon_t} = \mathcal{X}_{t-1} + \{x \in \mathbb{R}^n : \|x\|_2 \leq \varepsilon_t\}$.

(H2-NL)-(c) for every $j = 1, \ldots, M_t$, each component $g_{t,i}(\cdot, \cdot, \xi_{tj}), i = 1, \ldots, p$, of the function $g_t(\cdot, \cdot, \xi_{tj})$ is convex, lower semicontinuous, and finite on $\mathcal{X}_t \times \mathcal{X}_{t-1}^{\varepsilon_t}$.

(H2-NL)-(d) For every $j = 1, \ldots, M_t$, for every $x_{t-1} \in \mathcal{X}_{t-1}^{\varepsilon_t}$, the set $X_t(x_{t-1}, \xi_{tj})$ is nonempty.

(H2-NL)-(e) If $t \geq 2$, for every $j = 1, \ldots, M_t$, there exists

$$\bar{x}_{t,j} = (\bar{x}_{t,j,t}, \bar{x}_{t,j,t-1}) \in \mathrm{ri}(\mathcal{X}_t) \times \mathcal{X}_{t-1} \cap \mathrm{ri}(\{g_t(\cdot, \cdot, \xi_{tj}) \leq 0\})$$

such that $\bar{x}_{t,j,t} \in X_t(\bar{x}_{t,j,t-1}, \xi_{tj})$.

Assumptions (H2-NL)-(a),(b),(c) in the nonlinear case imply the convexity of cost-to-go functions $\mathcal{Q}_t(\cdot, 1)$. The assumptions above also ensure (both in the linear and nonliner cases) that SDDP applied to dynamic programming equations (20), (21), (22) will converge, as long as samples in the forward passes are independent, see [38, 17, 14] for details.

4.2. **Algorithm.** We now describe the steps of SDDP applied to dynamic programming equations (19), (20), (21), (22). We denote by SDDP-TSto this SDDP method for solving (16)[1]. SDDP-TSto is similar to the variant of SDDP presented in [37, 29] where the underlying stochastic process depends on a Markov Chain.[2] This Markov chain (process (D_t) for our DP equations) has only two states in our case. The cost-to-go function is null in one of these states (when $D_{t-1} = 0$) and the goal of SDDP-TSto is to approximate the cost-to-go function in the other state (when $D_{t-1} = 1$) for all stages, i.e., cost-to-go functions $\mathcal{Q}_t(\cdot, 1), t = 2, \ldots, T_{\max}$.

In the end of iteration k, the algorithm has computed for cost-to-go functions $\mathcal{Q}_t(\cdot, 1), t = 2, \ldots, T_{\max}$, the approximations $\mathcal{Q}_t^k(\cdot, 1), t = 2, \ldots, T_{\max}$, which are maximum of $k+1$ affine functions called cuts:

$$(40) \qquad \mathcal{Q}_t^k(x_{t-1}, 1) = \max_{0 \leq j \leq k} \theta_t^j + \langle \beta_t^j, x_{t-1} \rangle.$$

At iteration k, a realization of the number of stages and a sample for $(\xi_t)_{1 \leq t \leq T_{\max}}$, are generated. Decisions $x_t^k, t = 1, \ldots, T_{\max}$, are computed on this sample in a forward pass replacing (unknown) function $\mathcal{Q}_t(x_{t-1}, 1)$ by $\mathcal{Q}_t^{k-1}(x_{t-1}, 1)$. In the backward pass of iteration k, decisions x_t^k are then used to compute coefficients $\theta_t^k, \beta_t^k, t = 2, \ldots, T_{\max}$.

SDDP-TSto, Step 1: Initialization. Fix integer $N > 1$. For $t = 2, \ldots, T_{\max}$, take for $\mathcal{Q}_t^0(\cdot, 1)$ a known lower bounding affine function $\theta_t^0 + \langle \beta_t^0, \cdot \rangle$ for $\mathcal{Q}_t(\cdot, 1)$. Set the iteration count k to 1 and $\mathcal{Q}_{T_{\max}+1}^0(\cdot, 1) = \mathcal{Q}_{T_{\max}+1}^0(\cdot, 0) \equiv 0, \mathcal{Q}_t^0(\cdot, 0) \equiv 0, t = 2, \ldots, T_{\max}$. Fix a parameter $0 < \text{Tol} < 1$ (for the stopping criterion). Compute $q_t, t = 2, \ldots, T_{\max}$ using (4) starting from $q_2 = \mathbb{P}(T = 2)$.

SDDP-TSto, Step 2: Forward pass. We generate a sample

$$((\tilde{\xi}_1^k, \tilde{D}_1^k), (\tilde{\xi}_2^k, \tilde{D}_2^k), \ldots, (\tilde{\xi}_{T_{\max}}^k, \tilde{D}_{T_{\max}}^k)),$$

from the distribution of

$$\gamma^k = ((\xi_1^k, D_1^k), (\xi_2^k, D_2^k), \ldots, (\xi_{T_{\max}}^k, D_{T_{\max}}^k)) \sim ((\xi_1, D_1), (\xi_2, D_2), \ldots, (\xi_{T_{\max}}, D_{T_{\max}})),$$

with the convention that $\tilde{\xi}_1^k = \xi_1, \tilde{D}_1^k = 1$.
$\text{Cost}_k = 0$. $t \leftarrow 1$.
While $\tilde{D}_t^k = 1$, we compute an optimal solution x_t^k of

$$(41) \qquad \begin{cases} \inf_{x_t \in \mathbb{R}^n} f_t(x_t, x_{t-1}^k, \tilde{\xi}_t^k) + \mathcal{Q}_{t+1}^{k-1}(x_t, 1) \\ x_t \in X_t(x_{t-1}^k, \tilde{\xi}_t^k) \end{cases}$$

where $x_0^k = x_0$.
$\text{Cost}_k \leftarrow \text{Cost}_k + f_t(x_t^k, x_{t-1}^k, \tilde{\xi}_t^k)$.
$t \leftarrow t + 1$.

---

[1]TSto in acronym SDDP-TSto refers to the fact that this SDDP method solves stochastic programs with T stochastic, T being the number of stages.

[2]However, in [37, 29], problems are linear whereas we will detail SDDP-TSto both for linear and nonlinear stochastic programs.

End While

We compute an optimal solution x_t^k of

(42)
$$\begin{cases} \inf_{x_t \in \mathbb{R}^n} f_t(x_t, x_{t-1}^k, \tilde{\xi}_t^k) \\ x_t \in X_t(x_{t-1}^k, \tilde{\xi}_t^k). \end{cases}$$

$\mathrm{Cost}_k \leftarrow \mathrm{Cost}_k + f_t(x_t^k, x_{t-1}^k, \tilde{\xi}_t^k)$.

$t \leftarrow t + 1$.

While $(t \leq T_{\max} - 1)$, we compute an optimal solution x_t^k of

(43)
$$\begin{cases} \inf_{x_t \in \mathbb{R}^n} 0 \\ x_t \in X_t(x_{t-1}^k, \tilde{\xi}_t^k) \end{cases}$$

(note that $\tilde{D}_t^k = 0$ and the objective function is null, we only need a feasible point).

$t \leftarrow t + 1$.

End While

Upper bound computation: If $k \geq N$ compute

$$\overline{\mathrm{Cost}}_k = \frac{1}{N} \sum_{j=k-N+1}^{k} \mathrm{Cost}_j, \quad \hat{\sigma}_{N,k}^2 = \frac{1}{N} \sum_{j=k-N+1}^{k} [\mathrm{Cost}_j - \overline{\mathrm{Cost}}_k]^2$$

and the upper bound

$$\overline{U}_k = \overline{\mathrm{Cost}}_k + \frac{\hat{\sigma}_{N,k}}{\sqrt{N}} t_{N-1, 1-\alpha}$$

where $t_{N-1,1-\alpha}$ is the $(1-\alpha)$-quantile of the Student distribution with $N-1$ degrees of freedom.

SDDP-TSto, Step 3: Backward pass. Let $\underline{\mathfrak{Q}}_t^k(x_{t-1}, D_{t-1}, \xi_t, D_t)$ be the function given by

(44)
$$\underline{\mathfrak{Q}}_t^k(x_{t-1}, D_{t-1}, \xi_t, D_t) = \begin{cases} \inf_{x_t} D_{t-1} f_t(x_t, x_{t-1}, \xi_t) + \mathcal{Q}_{t+1}^k(x_t, D_t) \\ x_t \in X_t(x_{t-1}, \xi_t). \end{cases}$$

Set $\mathcal{Q}_{T_{\max}+1}^k(\cdot, 1) = \mathcal{Q}_{T_{\max}+1}^k(\cdot, 0) \equiv 0$.

For $t = T_{\max}$ down to $t = 2$,

Set $\mathcal{Q}_t^k(\cdot, 0) \equiv 0$.

For $j = 1, \ldots, M_t$,

Compute $\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, 1, \xi_{tj}, 1)$, compute

$$\mathfrak{Q}_t(x_{t-1}^k, 1, \xi_{tj}, 0) = \begin{cases} \inf_{x_t} f_t(x_t, x_{t-1}^k, \xi_{tj}) \\ x_t \in X_t(x_{t-1}^k, \xi_{tj}), \end{cases}$$

compute a subgradient β_{tj}^k of $\underline{\mathfrak{Q}}_t^k(\cdot, 1, \xi_{tj}, 1)$ at x_{t-1}^k and a subgradient γ_{tj}^k of $\mathfrak{Q}_t(\cdot, 1, \xi_{tj}, 0)$ at x_{t-1}^k.

End For

Compute

(45)
$$\begin{aligned} \theta_t^k &= (1 - q_t) \sum_{j=1}^{M_t} p_{tj} \Big(\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, 1, \xi_{tj}, 1) - \langle \beta_{tj}^k, x_{t-1}^k \rangle \Big) \\ &\quad + q_t \sum_{j=1}^{M_t} p_{tj} \Big(\mathfrak{Q}_t(x_{t-1}^k, 1, \xi_{tj}, 0) - \langle \gamma_{tj}^k, x_{t-1}^k \rangle \Big), \\ \beta_t^k &= (1 - q_t) \sum_{j=1}^{M_t} p_{tj} \beta_{tj}^k + q_t \sum_{j=1}^{M_t} p_{tj} \gamma_{tj}^k. \end{aligned}$$

End For

Lower bound computation: compute the lower bound $\underline{L}_k$ on the optimal value of (16) given by

$$\underline{L}_k = \begin{cases} \inf_{x_1} \ f_1(x_1, x_0, \xi_1) + \mathcal{Q}_2^k(x_1, 1) \\ x_1 \in X_1(x_0, \xi_1). \end{cases}$$

SDDP-TSto, Step 4: If $k \geq N$ and $\frac{\overline{U}_k - \underline{L}_k}{\overline{U}_k} \leq$ Tol then stop otherwise do $k \leftarrow k+1$ and go to Step 2.

We now show that the cuts computed by SDDP-TSto are valid, that $\underline{L}_k$ is a lower bound on the optimal value of the problem and that the sequence of approximate first stage problems optimal values converges almost surely to the optimal value of (16).

Theorem 4.1. *Consider optimization problem (16) where for $t = 1, \ldots, T_{\max}$, ξ_t does not depend on $(\xi_{[t-1]}, D_t)$ and D_t only depends on D_{t-1}. Assume that (H1) holds and that the distribution of ξ_t is discrete for $t = 1, \ldots, T_{\max}$. In the case of linear problems (X_t as in (38)) assume that (H2-L) holds and in the case of nonlinear problems (X_t as in (39)) assume that (H2-NL)-(a)-(e) holds. Consider the sequences $(x_t^k)_{k \geq 1}, t = 1, \ldots, T_{\max}$ and $(\mathcal{Q}_t^k(\cdot, 1))_{k \geq 0}, t = 2, \ldots, T_{\max}$, generated by SDDP-TSto to solve the corresponding dynamic programming equations (19), (20), (21), (22).*

Assume that samples in the forward passes are independent: the sample

$$((\tilde{\xi}_1^k, \tilde{D}_1^k), (\tilde{\xi}_2^k, \tilde{D}_2^k), \ldots, (\tilde{\xi}_{T_{\max}}^k, \tilde{D}_{T_{\max}}^k))$$

in the forward pass of iteration k is a realization of random vector

$$\gamma^k = ((\xi_1^k, D_1^k), (\xi_2^k, D_2^k), \ldots, (\xi_{T_{\max}}^k, D_{T_{\max}}^k))$$

which has the distribution of $((\xi_1, D_1), (\xi_2, D_2), \ldots, (\xi_{T_{\max}}, D_{T_{\max}}))$ and $\gamma^1, \gamma^2, \ldots$ are independent. Then

(i) *for $t = 2, \ldots, T_{\max} + 1$, for all $k \geq 0$, $\mathcal{Q}_t^k(\cdot, 1)$ is a lower bounding function for $\mathcal{Q}_t(\cdot, 1)$: for all x_{t-1} we have $\mathcal{Q}_t(x_{t-1}, 1) \geq \mathcal{Q}_t^k(x_{t-1}, 1)$ almost surely.*

(ii) *$\underline{L}_k$ computed in Step 3 of SDDP-TSto is a lower bound on the optimal value of (16).*

(iii) *Almost surely the limit of the sequence $(f_1(x_1^k, x_0, \xi_1) + \mathcal{Q}_2^k(x_1^k, 1))_{k \geq 1}$ is the optimal value of (16).*

Proof. (i) The proof is by induction on k and t. For $k = 0$, we have $\mathcal{Q}_t(\cdot, 1) \geq \mathcal{Q}_t^0(\cdot, 1), t = 2, \ldots, T_{\max} + 1$. Now assume that

(46) $$\mathcal{Q}_t(\cdot, 1) \geq \mathcal{Q}_t^{k-1}(\cdot, 1), t = 2, \ldots, T_{\max} + 1,$$

for some $k \geq 1$. We show by backward induction on t that $\mathcal{Q}_t(\cdot, 1) \geq \mathcal{Q}_t^k(\cdot, 1), t = 2, \ldots, T_{\max} + 1$. For $t = T_{\max} + 1$ we have $\mathcal{Q}_t(\cdot, 1) = \mathcal{Q}_t^k$ (both functions are null). Now assume that $\mathcal{Q}_{t+1}(\cdot, 1) \geq \mathcal{Q}_{t+1}^k(\cdot, 1)$ for some $t \in \{2, \ldots, T_{\max}\}$ (induction hypothesis). We want to show that

(47) $$\mathcal{Q}_t(\cdot, 1) \geq \mathcal{Q}_t^k(\cdot, 1).$$

The induction hypothesis, together with the definitions of $\underline{\mathfrak{Q}}_t$ and $\mathfrak{Q}_t^k$ imply that for all $j = 1, \ldots, M_t$:

(48) $$\mathfrak{Q}_t(\cdot, 1, \xi_{tj}, 1) \geq \underline{\mathfrak{Q}}_t^k(\cdot, 1, \xi_{tj}, 1).$$

Therefore, we get

$$
\begin{aligned}
\mathcal{Q}_t(\cdot, 1) \;\overset{(20)}{=}\;& (1-q_t) \sum_{j=1}^{M_t} p_{tj} \mathfrak{Q}_t(\cdot, 1, \xi_{tj}, 1) + q_t \sum_{j=1}^{M_t} p_{tj} \mathfrak{Q}_t(\cdot, 1, \xi_{tj}, 0), \\
\overset{(48)}{\geq}\;& (1-q_t) \sum_{j=1}^{M_t} p_{tj} \underline{\mathfrak{Q}}_t^k(\cdot, 1, \xi_{tj}, 1) + q_t \sum_{j=1}^{M_t} p_{tj} \mathfrak{Q}_t(\cdot, 1, \xi_{tj}, 0), \\
\geq\;& (1-q_t) \sum_{j=1}^{M_t} p_{tj} \Big[\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, 1, \xi_{tj}, 1) + \langle \beta_{tj}^k, \cdot - x_{t-1}^k \rangle \Big] \\
& + q_t \sum_{j=1}^{M_t} p_{tj} \Big[\mathfrak{Q}_t(x_{t-1}^k, 1, \xi_{tj}, 0) + \langle \gamma_{tj}^k, \cdot - x_{t-1}^k \rangle \Big], \\
=\;& \theta_t^k + \langle \beta_t^k, \cdot \rangle \text{ by definition of } \theta_t^k, \beta_t^k,
\end{aligned}
$$

(49)

where for the second inequality we have used the subgradient inequality and the definition of $\beta_{tj}^k, \gamma_{tj}^k$. Combining (46), (49), and the relation $\mathcal{Q}_t^k(\cdot, 1) = \max(\mathcal{Q}_t^{k-1}(\cdot, 1), \theta_t^k + \langle \beta_t^k, \cdot \rangle)$, we obtain (47), which achieves the induction step and the proof of (i).

(ii) It suffices to observe that the optimal value of (16) is the optimal value of (15) and that, due to (i), $\mathcal{Q}_2(x_1, 1) \geq \mathcal{Q}_2^k(x_1, 1)$ (recall that under our assumptions $\mathcal{Q}_2$ does not depend on ξ_1).

(iii) can be proved following the convergence proofs of SDDP from [38] in the linear case and from [17] in the nonlinear case which apply under our assumptions. $\qquad\square$

In the steps of SDDP-TSto above, we have not detailed the computation of β_{tj}^k and γ_{tj}^k. In the linear and nonlinear settings mentioned above, the formulas for these coefficients are given below. When $\xi_t = \xi_{tj}$, we will denote by A_{tj}, B_{tj}, and b_{tj} the realizations of A_t, B_t, and b_t, respectively.

Computation of β_{tj}^k and γ_{tj}^k in the nonlinear case. Formulas for cuts computed by SDDP when X_t is of form (39) were given in Lemma 2.1 in [17]. We recall these formulas below. For the optimization problem

$$
\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, 1, \xi_{tj}, 1) = \begin{cases}
\inf_{x_t} \ f_t(x_t, x_{t-1}^k, \xi_{tj}) + \mathcal{Q}_{t+1}^k(x_t, 1) \\
A_{tj} x_t + B_{tj} x_{t-1}^k = b_{tj}, \quad [\lambda_{tj}^{k1}] \\
g_t(x_t, x_{t-1}^k, \xi_{tj}) \leq 0, \quad [\mu_{tj}^{k1}] \\
x_t \in \mathcal{X}_t,
\end{cases}
$$

denote by x_{tj}^{k1} an optimal solution, consider the Lagrangian

$$
L(x_t, \lambda, \mu; x_{t-1}^k, \xi_{tj}) = f_t(x_t, x_{t-1}^k, \xi_{tj}) + \mathcal{Q}_{t+1}^k(x_t, 1) + \lambda^T (b_{tj} - A_{tj} x_t - B_{tj} x_{t-1}^k) + \mu^T g_t(x_t, x_{t-1}^k, \xi_{tj}),
$$

and optimal Lagrange multipliers $(\lambda_{tj}^{k1}, \mu_{tj}^{k1})$. Similarly, for the optimization problem

$$
\mathfrak{Q}_t(x_{t-1}^k, 1, \xi_{tj}, 0) = \begin{cases}
\inf_{x_t} \ f_t(x_t, x_{t-1}^k, \xi_{tj}) \\
A_{tj} x_t + B_{tj} x_{t-1}^k = b_{tj}, \quad [\lambda_{tj}^{k2}] \\
g_t(x_t, x_{t-1}^k, \xi_{tj}) \leq 0, \quad [\mu_{tj}^{k2}] \\
x_t \in \mathcal{X}_t,
\end{cases}
$$

denote by x_{tj}^{k2} an optimal solution, consider the Lagrangian

$$
L(x_t, \lambda, \mu; x_{t-1}^k, \xi_{tj}) = f_t(x_t, x_{t-1}^k, \xi_{tj}) + \lambda^T (b_{tj} - A_{tj} x_t - B_{tj} x_{t-1}^k) + \mu^T g_t(x_t, x_{t-1}^k, \xi_{tj}),
$$

and optimal Lagrange multipliers $(\lambda_{tj}^{k2}, \mu_{tj}^{k2})$.

Let $f'_{t, x_{t-1}}(x_{tj}^{k1}, x_{t-1}^k, \xi_{tj})$ (resp. $f'_{t, x_{t-1}}(x_{tj}^{k2}, x_{t-1}^k, \xi_{tj})$) be a subgradient of convex function $f_t(x_{tj}^{k1}, \cdot, \xi_{tj})$ (resp. $f_t(x_{tj}^{k2}, \cdot, \xi_{tj})$) at x_{t-1}^k. Let $g'_{t, i, x_{t-1}}(x_{tj}^{k1}, x_{t-1}^k, \xi_{tj})$ (resp. $g'_{t, i, x_{t-1}}(x_{tj}^{k2}, x_{t-1}^k, \xi_{tj})$)

be a subgradient of convex function $g_{t,i}(x_{tj}^{k1}, \cdot, \xi_{tj})$ (resp. $g_{t,i}(x_{tj}^{k2}, \cdot, \xi_{tj})$) at x_{t-1}^k. With this notation, setting

$$\beta_{tj}^k = f'_{t,x_{t-1}}(x_{tj}^{k1}, x_{t-1}^k, \xi_{tj}) - B_{tj}^T \lambda_{tj}^{k1} + \sum_{i=1}^m \mu_{tj}^{k1}(i) g'_{t,i,x_{t-1}}(x_{tj}^{k1}, x_{t-1}^k, \xi_{tj}),$$

$$\gamma_{tj}^k = f'_{t,x_{t-1}}(x_{tj}^{k2}, x_{t-1}^k, \xi_{tj}) - B_{tj}^T \lambda_{tj}^{k2} + \sum_{i=1}^m \mu_{tj}^{k2}(i) g'_{t,i,x_{t-1}}(x_{tj}^{k2}, x_{t-1}^k, \xi_{tj}),$$

then β_{tj}^k is a subgradient of $\underline{\mathfrak{Q}}_t^k(\cdot, 1, \xi_{tj}, 1)$ at x_{t-1}^k and γ_{tj}^k is a subgradient of $\mathfrak{Q}_t(\cdot, 1, \xi_{tj}, 0)$ at x_{t-1}^k.

Computation of β_{tj}^k and γ_{tj}^k in the linear case. Formulas for the cuts in the linear case are well known. Due to (H1-L) the optimal value of the linear program

$$\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, 1, \xi_{tj}, 1) = \begin{cases} \inf_{x_t} \ c_{tj}^T x_t + \mathcal{Q}_{t+1}^k(x_t, 1) \\ A_{tj} x_t + B_{tj} x_{t-1}^k = b_{tj}, \quad [\lambda_{tj}^{k1}] \\ x_t \geq 0, \end{cases}$$

is the optimal value of the corresponding dual problem:

(50) $$\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, 1, \xi_{tj}, 1) = \begin{cases} \sup_{\lambda,\mu} \ \lambda^T(b_{tj} - B_{tj} x_{t-1}^k) + \sum_{i=1}^k \mu_i \theta_{t+1}^i \\ A_{tj}^T \lambda + \sum_{i=1}^k \mu_i \beta_{t+1}^i \leq c_{tj}, \ \sum_{i=1}^k \mu_i = 1, \\ \mu_i \geq 0, i = 1, \ldots, k. \end{cases}$$

Let $(\lambda_{tj}^{k1}, \mu_{tj}^{k1})$ be an optimal solution of dual problem (50).

Similarly, due to (H1-L) the optimal value of the linear program

$$\mathfrak{Q}_t(x_{t-1}^k, 1, \xi_{tj}, 0) = \begin{cases} \inf_{x_t} \ c_{tj}^T x_t \\ A_{tj} x_t + B_{tj} x_{t-1}^k = b_{tj}, \quad [\lambda_{tj}^{k2}] \\ x_t \geq 0, \end{cases}$$

is the optimal value of the dual problem

(51) $$\mathfrak{Q}_t(x_{t-1}^k, 1, \xi_{tj}, 0) = \begin{cases} \sup_\lambda \ \lambda^T(b_{tj} - B_{tj} x_{t-1}^k) \\ A_{tj}^T \lambda \leq c_{tj}. \end{cases}$$

Let (λ_{tj}^{k2}) be an optimal solution of dual problem (51).

With this notation, setting

$$\beta_{tj}^k = -B_{tj}^T \lambda_{tj}^{k1} \text{ and } \gamma_{tj}^k = -B_{tj}^T \lambda_{tj}^{k2},$$

then β_{tj}^k is a subgradient of $\underline{\mathfrak{Q}}_t^k(\cdot, 1, \xi_{tj}, 1)$ at x_{t-1}^k and γ_{tj}^k is a subgradient of $\mathfrak{Q}_t(\cdot, 1, \xi_{tj}, 0)$ at x_{t-1}^k.

5. NUMERICAL EXPERIMENTS: PORTFOLIO SELECTION WITH A RANDOM INVESTMENT PERIOD

5.1. **SDDP-TSto for the portfolio problem.** We consider the portfolio problem given in Section 3.2 and the corresponding dynamic programming equations (35), (36), (37) when the number of stages is stochastic. We now specialize the SDDP-TSto algorithm applied to these DP equations. In particular, we explain how to compute coefficients γ_{tj}^k and β_{tj}^k for this problem.

In the forward pass of SDDP-TSto, problem (41) becomes

(52) $$\begin{cases} \inf_{x_t \in \mathbb{R}^n} \ \mathcal{Q}_{t+1}^{k-1}(x_t, 1) \\ x_t \in X_t(x_{t-1}^k, \tilde{\xi}_t^k) \end{cases}$$

while problem (42) specializes to

$$(53) \qquad \begin{cases} \inf_{x_t \in \mathbb{R}^n} -\mathbb{E}[\sum_{i=1}^{n+1} \xi_{t+1}(i)x_t(i)] \\ x_t \in X_t(x_{t-1}^k, \tilde{\xi}_t^k). \end{cases}$$

The backward pass is given below.

Backward pass of SDDP-TSto for the portfolio problem. Set $\mathcal{Q}_{T_{\max}+1}^k(\cdot, 1) \equiv 0$.
For $t = T_{\max}$ down to $t = 2$,
  For $j = 1, \dots, M_t$,
    Solve the optimization problem

$$\begin{cases} \inf -\mathbb{E}[\sum_{i=1}^{n+1} \xi_{t+1}(i)x_t(i)] \\ x_t \in X_t(x_{t-1}^k, \xi_{tj}), \end{cases}$$

    with Lagrangian $L(x_t, y_t, z_t, \lambda_1, \mu_1, \delta_1)$ given by

$$-\mathbb{E}[\sum_{i=1}^{n+1} \xi_{t+1}(i)x_t(i)] + \left\langle \lambda_1, \xi_{tj} \circ x_{t-1} - x_t + \left[\begin{array}{c} -y_t + z_t \\ (\mathbf{e} - \eta_t)^T y_t - (\mathbf{e} + \nu_t)^T z_t \end{array} \right] \right\rangle$$
$$+ \langle \mu_1, y_t - \xi_{tj}(1:n) \circ x_{t-1}(1;n) \rangle + \langle \delta_1, x_t(1:n) - (\xi_{tj}^T x_{t-1})u \rangle$$

    where $\mathbf{e}$ is a vector in $\mathbb{R}^n$ of ones, $\lambda_1 \in \mathbb{R}^{n+1}, \mu_1, \delta_1 \in \mathbb{R}^n$, and for vectors x, y,
    the vector $x \circ y$ has components $(x \circ y)(i) = x(i)y(i)$ and $\langle x, y \rangle = x^T y$.
    For this Lagrangian, let $(\lambda_{1tj}^k, \mu_{1tj}^k, \delta_{1tj}^k)$ be optimal Lagrange multipliers.
    Solve the optimization problem

$$\begin{cases} \inf \mathcal{Q}_{t+1}^k(x_t, 1) \\ x_t \in X_t(x_{t-1}^k, \xi_{tj}), \end{cases}$$

    with Lagrangian $L(x_t, y_t, z_t, \lambda_2, \mu_2, \delta_2)$ given by

$$\mathcal{Q}_{t+1}^k(x_t, 1) + \left\langle \lambda_2, \xi_{tj} \circ x_{t-1} - x_t + \left[\begin{array}{c} -y_t + z_t \\ (\mathbf{e} - \eta_t)^T y_t - (\mathbf{e} + \nu_t)^T z_t \end{array} \right] \right\rangle$$
$$+ \langle \mu_2, y_t - \xi_{tj}(1:n) \circ x_{t-1}(1;n) \rangle + \langle \delta_2, x_t(1:n) - (\xi_{tj}^T x_{t-1})u \rangle$$

    where $\mathbf{e}$ is a vector in $\mathbb{R}^n$ of ones, $\lambda_2 \in \mathbb{R}^{n+1}, \mu_2, \delta_2 \in \mathbb{R}^n$.
    For this Lagrangian, let $(\lambda_{2tj}^k, \mu_{2tj}^k, \delta_{2tj}^k)$ be optimal Lagrange multipliers.
    Compute

$$\gamma_{tj}^k = \left(\lambda_{1tj}^k - (u^T \delta_{1tj}^k)\mathbf{e} - \left[\begin{array}{c} \mu_{1tj}^k \\ 0 \end{array} \right] \right) \circ \xi_{tj},$$
$$\beta_{tj}^k = \left(\lambda_{2tj}^k - (u^T \delta_{2tj}^k)\mathbf{e} - \left[\begin{array}{c} \mu_{2tj}^k \\ 0 \end{array} \right] \right) \circ \xi_{tj},$$

    where $\mathbf{e}$ is a vector in $\mathbb{R}^{n+1}$ of ones.
  End For
  Compute[3]

$$\theta_t^k = 0 \text{ and } \beta_t^k = (1 - q_t)\sum_{j=1}^{M_t} p_{tj}\beta_{tj}^k + q_t \sum_{j=1}^{M_t} p_{tj}\gamma_{tj}^k.$$

End For

---

[3]Observe that the intercept for the cuts is zero for that application.

5.2. **Numerical results.** Our goal in this section is to compare SDDP run on an optimization period of $T_{\max}$ stages (i.e., where we assume that the number of stages is known and fixed to $T_{\max}$) and SDDP-TSto on the risk-neutral portfolio problem with direct transaction costs given in Section 3.2. All subproblems in the forward and backward passes of SDDP and SDDP-TSto were solved numerically using the interior point solver of the Mosek Optimization Toolbox [1] and the Matlab code was run on a Xeon E5-2670 processor with 384 GB of RAM. The following parameters are chosen for our experiments.

Distributions of T and of returns (ξ_t). We consider two distributions for random variable T: a uniform distribution over the set $\{2, 3, \ldots, T_{\max}\}$ and a truncated exponential discretized distribution. The reason for choosing this latter distribution for a portfolio problem is motivated by reference [9], where the lifetime of more than 25 000 publicly traded North American companies, from 1950 to 2009, was analyzed. It was shown that *mortality rates are independent of the company's age, the typical half-life of a publicly traded company is about a decade*, and the exponential distribution is a good fit for the lifetime of these companies on this period. Therefore, for such companies, the exponential distribution makes sense for the duration of an optimization period of a portfolio problem. However, since the number of stages is almost surely in the set $\{2, \ldots, T_{\max}\}$, instead of an exponential distribution, we take for T a translation of a discretization of an exponential distribution conditioned on the event that this exponential distribution belongs to $[\frac{1}{2}, T_{\max} - \frac{1}{2}[$. In what follows, we call that distribution truncated exponential discretized.[4] More precisely, let $X \sim \mathcal{E}(\lambda)$ be the exponential distribution with parameter $\lambda = 0.15$ with expectation $\mathbb{E}[X] = \frac{1}{\lambda} \approx 6.67$. Setting $Y = X|A$ where A is the event $A = \{\omega : \frac{1}{2} \leq X(\omega) \leq T_{\max} - \frac{1}{2}\}$ and defining the random variable T by $T = t + 1$ if and only if $t - \frac{1}{2} \leq Y < t + \frac{1}{2}$ for $t = 1, \ldots, T_{\max} - 1$, then the distribution of the number of stages T is given by

$$\mathbb{P}(T = t + 1) = \mathbb{P}\left(t - \frac{1}{2} \leq Y < t + \frac{1}{2}\right) = \frac{\mathbb{P}(t - \frac{1}{2} \leq X < t + \frac{1}{2})}{\mathbb{P}(A)} = \frac{e^{-\lambda(t - \frac{1}{2})} - e^{-\lambda(t + \frac{1}{2})}}{e^{-\lambda/2} - e^{-\lambda(T_{\max} - \frac{1}{2})}},$$

for $t = 1, \ldots, T_{\max} - 1$. The histogram of the distribution of T when $T_{\max} = 10$ is represented in Figure 2 together with the graph of the density of $2 + X$ over the interval $[2, 10]$.

The return of the risk-free asset $n + 1$ is 1.01 for every stage. Returns $\xi_t, t = 2, \ldots, T_{\max}$, have discrete distributions with $M_t = M$ realizations, each having probability $\frac{1}{M}$ and realizations $\xi_1(1 : n), \xi_{t1}(1 : n), \ldots, \xi_{tM}(1 : n)$ obtained sampling from a normal distribution with mean and standard deviation chosen randomly in respectively the intervals $[0.9, 1.4]$ and $[0.1, 0.2]$.

Remaining parameters of the portfolio problem. The initial portfolio x_0 has components $x_0(i), i = 1, \ldots, n + 1$, uniformly distributed in $[0, 1000]$ (vector of initial wealth in each asset). Transaction costs are known with $\nu_t(i) = \mu_t(i)$ obtained sampling from the distribution of the random variable $0.08 + 0.06 \cos(\frac{2\pi}{T} U_{T_{\max}})$ where $U_{T_{\max}}$ is a random variable with a discrete distribution over the set of integers $\{1, 2, \ldots, T_{\max}\}$. The largest position in any security is set to 80%, i.e., $u(i) = 0.8$ for $i = 1, \ldots, n$.

Parameters of SDDP and SDDP-TSto methods. Using the notation of the previous section, SDDP and SDDP-TSto are run with parameters $N = 400$, $\alpha = 0.05$, and Tol=0.05.

---

[4]Considering the values chosen for the realizations of the returns and today's "usual" asset returns, we can consider that a stage corresponds to a few years and that the maximal duration of the optimization period $T_{\max} = 10$ is a few decades.
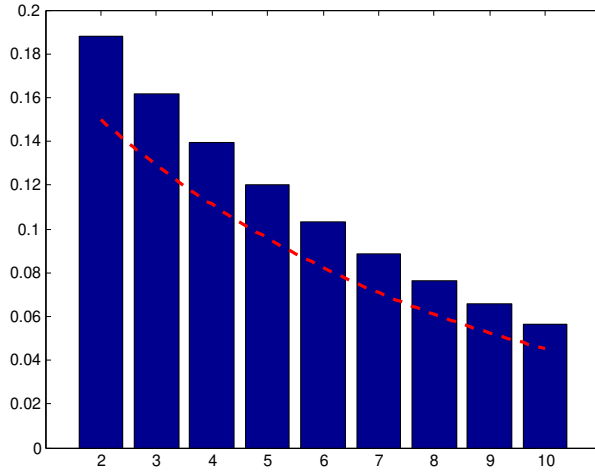
FIGURE 2. Histogram of the distribution of T with support $\{2, \ldots, T_{\max}\} = \{2, \ldots, 10\}$ and density of $2 + \mathcal{E}(\lambda)$ (in dotted line) on the interval $[2, 10]$ with $\lambda = 0.15$.

Comparing SDDP and SDDP-TSto policies on Monte-Carlo simulations.

We consider 6 instances of the portfolio problem with random number of stages T that we have just presented where the distribution of T is either the truncated exponential discretized distribution given above or the uniform distribution on the set $\{2 \ldots, T_{\max}\}$. For each distribution of T, we take 3 values for $(T_{\max}, n, M)$ given in Table 2. This defines 6 instances for which we compute two policies.

The first policy, denoted by SDDP, is obtained running SDDP on the portfolio problem assuming, as has usually been the case so far in applications of stochastic programming, that the number of stages is fixed. Since we need a policy defined for all possible stages from $t = 1$ to $t = T_{\max}$, this SDDP policy is obtained running SDDP on a portfolio problem with a deterministic number of stages fixed to $T_{\max}$.

The second policy is obtained running the SDDP-Tsto algorithm from Section 4.2 on the portfolio problem with random number of stages T. With a slight abuse of notation, we call the corresponding policy SDDP-TSto.

For each instance, we then simulate SDDP and SDDP-TSto policies on 5 000 scenarios for $((D_1, \xi_1), (D_2, \xi_2), \ldots, (D_{T_{\max}}, \xi_{T_{\max}}))$. For each instance, the empirical mean income with both policies on these 5 000 scenarios is given in Table 2. We see that, as expected, the mean income is larger (sometimes much larger) with SDDP-TSto. We also report in this table the p-value of the paired Student t-test

$$(54) \qquad H_0 : \mathbb{E}[I_{\mathrm{SDDP}}] \geq \mathbb{E}[I_{\mathrm{SDDP\text{-}TSto}}] \text{ against } H_1 : \mathbb{E}[I_{\mathrm{SDDP}}] < \mathbb{E}[I_{\mathrm{SDDP\text{-}TSto}}],$$

computed using the 5 000 realizations of income of SDDP-TSto and SDDP policies. In this test, I_{SDDP} and $I_{\mathrm{SDDP\text{-}TSto}}$ are the income with respectively SDDP and SDDP-TSto policies. We observe that for the three instances the p-value is small. More precisely, on four instances the p-value is less than 5% meaning that at the significance level 5%, we accept for these instances the hypothesis H_1, i.e., the hypothesis that the mean income with SDDP-TSto is greater than the mean income with SDDP policy. For the remaining two instances, the p-values are 6.8% and 7.2%, i.e., we accept for these 2 instances the hypothesis that SDDP-TSto income is higher than SDPP income at any significance level above 7.2%.

Distribution of T	$T_{\max}$	n	M	Empirical mean income SDDP-TSto	Empirical mean income SDDP	p-value
Uniform on $2, \ldots, T_{\max}$	10	10	50	14 973	14 563	0.035
Truncated exponential discretized	10	10	50	12 466	12 138	0.002
Uniform on $2, \ldots, T_{\max}$	10	20	50	34 410	33 875	0.072
Truncated exponential discretized	10	20	50	32 783	31 121	6.5×10^{-8}
Uniform on $2, \ldots, T_{\max}$	10	10	100	12 977	12 797	0.068
Truncated exponential discretized	20	10	20	24 181	23 700	0.040

TABLE 2. Empirical mean income for SDDP-TSto and SDDP policies on 6 instances of the portfolio problem with a random number of stages and p-values of the test $H_0 : \mathbb{E}[I_{\text{SDDP}}] \geq \mathbb{E}[I_{\text{SDDP-TSto}}]$ against $H_1 : \mathbb{E}[I_{\text{SDDP}}] < \mathbb{E}[I_{\text{SDDP-TSto}}]$ for these instances, on the basis of samples of size 5 000 of the income of SDDP-TSto and SDDP policies. In H_0 and H_1, I_{SDDP} and $I_{\text{SDDP-TSto}}$ are the income with respectively SDDP and SDDP-TSto policies.

Remark 5.1. *We performed the same Monte Carlo simulations on several other instances with larger values of n and M. For all our experiments, the empirical mean income with SDDP-TSto was higher than the empirical mean income with SDDP but the p-value of paired t-test (54) could be as high as 0.4. From these experiments, it seems that the larger the size of the deterministic equivalent (in particular the larger n and M), the larger the standard deviation of the income and therefore the less paired t-test (54) will accept (at the significance level 5%) the hypothesis that SDDP-TSto income is higher.*

Comparing the empirical distribution of the solution time for SDDP and SDDP-TSto.

When we face a multistage stochastic program with a random number of stages T with support the set of integers $\{2, 3, \ldots, T_{\max}\}$, we also expect SDDP-TSto policy to be computed quicker than SDDP policy, recalling that the latter is obtained running SDDP assuming the number of stages is fixed and equal to $T_{\max}$. Indeed, in the forward pass of iteration k of SDDP-TSto, once we reach the first stage t_0 such that $\tilde{D}_{t_0}^k = 0$, for subproblems of subsequent stages $t > t_0$, the objective function is null, i.e., all that the trial points have to satisfy is to be feasible. In the case of our portfolio problem, we can even obtain a drastic reduction in the time needed to compute SDDP-TSto policy as can be seen on Figure 3 which reports the empirical distribution of the computational time needed to solve instances of the portfolio problem (with truncated exponential discretized distribution for T) 10 times with SDDP and SDDP-TSto. Accross the 10 runs, the time needed to compute a given policy does not vary much and the computational bulk with SDDP-TSto is around 10 times inferior to the computational bulk with SDDP.

Summary of the advantages of the proposed methodology and solution method.

$M = 50, n = 10, T_{\max} = 10$

$M = 50, n = 20, T_{\max} = 10$
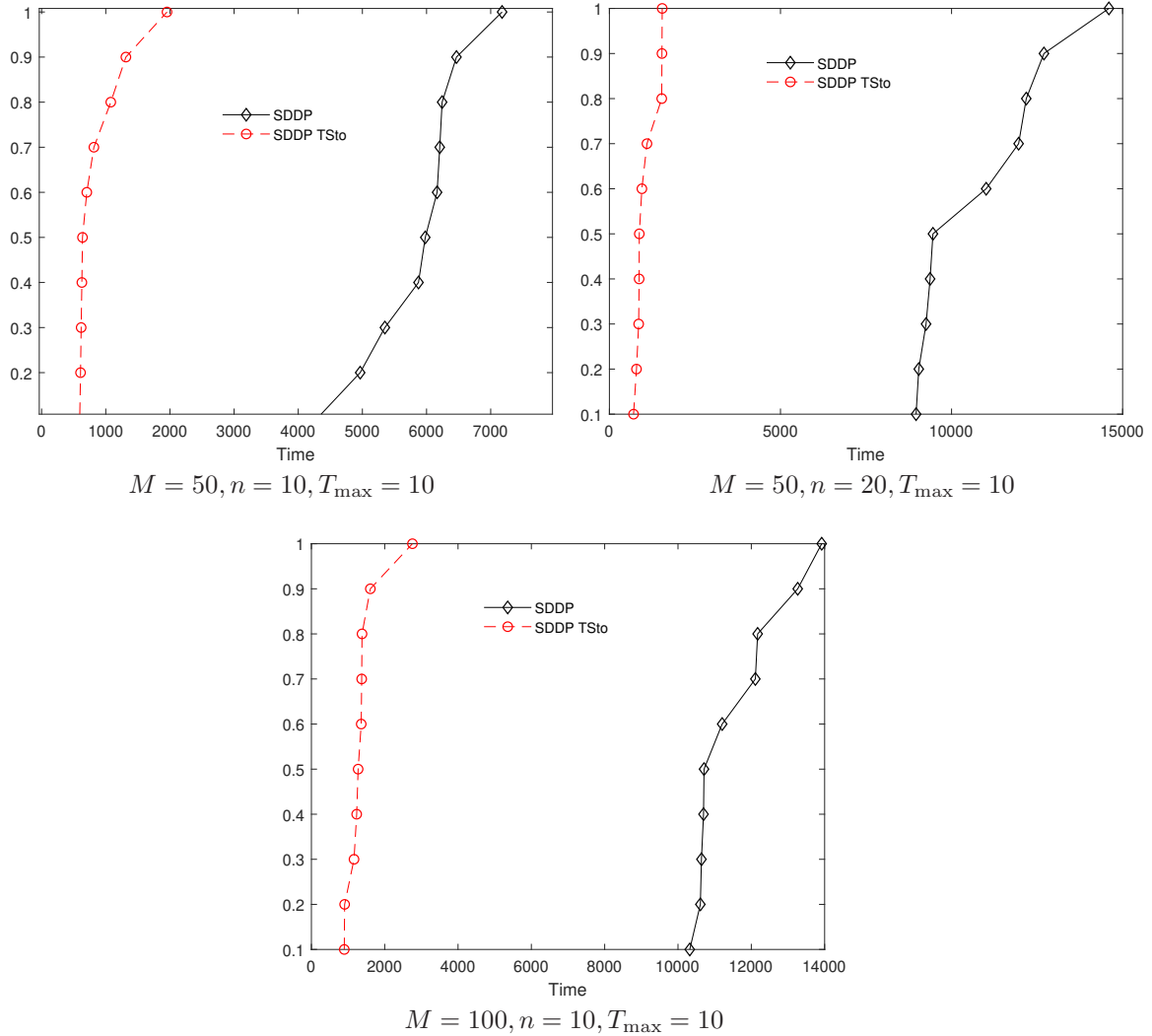
$M = 100, n = 10, T_{\max} = 10$

FIGURE 3. Empirical distribution of the computational time needed to solve instances of the portfolio problem with truncated exponential discretized distribution for T (for each instance, SDDP and SDDP-TSto are run 10 times).

So far, to our knowledge, stochastic programs with a random number of stages had not been considered in the stochastic programming community. Therefore, in the presence of a multistage stochastic program with a random number T of stages upper bounded by $T_{\max}$, the traditional approach would be in fact to ignore the stochasticity of T, to assume that the number of stages is fixed to $T_{\max}$ so that a policy can be defined for all possible stages $t = 1, 2, \ldots, T_{\max}$, and, as long as the problem is convex, a popular approach to solve such a problem would be to use SDDP. Compared to this approach, the tools proposed in this article offer the following advantages:

- The Dynamic Programming equations written in Section 2 define the appropriate Bellman functions for the problem under consideration, which, obviously, yield an optimal cost lower than the one obtained using Bellman functions defined by Dynamic Programming equations

for a problem with $T_{\max}$ fixed stages. Observe also that the model and Dynamic Programming equations of Section 2 do not require any assumption of convexity.

- If the stochastic program is convex and as long as SDDP-TSto algorithm given in Section 4 and SDDP (run on the Dynamic Programming equations with $T_{\max}$ fixed stages) are run for a sufficient amount of iterations to satisfy a stopping criterion with small values of α and Tol, SDDP-TSto policy will be better (and can be much better) than SDDP policy, see the experiments reported in this section on the portfolio problem.
- We expect SDDP-TSto policy to be computed quicker than SDDP policy (the latter being run on a problem with $T_{\max}$ stages). For our portfolio problem with a random number of stages, SDDP-TSto was computed up to 10 times faster than SDDP, see Figure 3.

6. Conclusion

We introduced the class of multistage stochastic programs with a random number of stages. We explained how to write dynamic programming equations for such problems and detailed the SDDP algorithm to solve these dynamic programming equations. We have shown the applicability and interest of the proposed models and methodology for portfolio selection.

As a future work, it would be interesting to consider more general "hybrid" stochastic programs with transition probabilities between objective and cost functions, meaning that at each stage not only parameters but also cost and constraint functions are random, possibly depending on past values of parameters and cost and constraint functions. Other possible extensions could consider stages of random durations or problems where decisions are only taken at random times driven by an exogenous stochastic process. The random times at which these decisions would be taken could also depend on previous decisions. Finally, it would be interesting to use the proposed models and methodology for other applications, for instance Asset Liability Management or the applications mentioned in the introduction.

Acknowledgments

References

[1] E. D. Andersen and K.D. Andersen. *The MOSEK optimization toolbox for MATLAB manual. Version 7.0*, 2013. https://www.mosek.com/.

[2] A. Ben-Tal, T. Margalit, and A. Nemirovski. Robust modeling of multi-stage portfolio problems. *in: H. Frenk, K. Roos, T. Terlaky, S. Zhang, Eds., High Performance Optimization, Kluwer Academic Publishers*, pages 303–328, 2000.

[3] J.R. Birge and C. J. Donohue. The Abridged Nested Decomposition Method for Multistage Stochastic Linear Programs with Relatively Complete Recourse. *Algorithmic of Operations Research*, 1:20–30, 2001.

[4] S.P. Bradley and D.B. Crane. A dynamic model for bond portfolio management. *Management Science*, 19:139–151, 1972.

[5] L. Chen and T. Homem de Mello. A scenario-based stochastic programming approach for technology and capacity planning. *Computers & Operations Research*, 29:781–806, 2002.

[6] L. Chen and T. Homem de Mello. Re-solving stochastic programming models for airline revenue management. *Annals of Operations Research*, 177:91–114, 2010.

[7] Z.L. Chen and W.B. Powell. Convergent Cutting-Plane and Partial-Sampling Algorithm for Multistage Stochastic Linear Programs with Recourse. *J. Optim. Theory Appl.*, 102:497–524, 1999.

[8] G. Consigli, D. Kuhn, and P. Brandimarte. *Optimal Financial Decision Making under Uncertainty, International Series in Operations Research & Management Science*. Springer, 2017.

[9] M.I.G. Daepp, M.J. Hamilton, G.B. West, and L.M.A. Bettencourt. The mortality of companies. *J. R. Soc. Interface*, 12:1–8, 2015.

[10] G.B. Dantzig and G. Infanger. Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research*, 45:59–76, 1993.

[11] M.H.A. Davis, M.A.H. Dempster, S.P. Sethi, and D. Vermes. Optimal capacity expansion under uncertainty. *Advances in Applied Probability*, 19:156–176, 1987.

[12] T.P. Filomena and M.A. Lejeune. Stochastic portfolio optimization with proportional transaction costs: Convex reformulations and computational experiments. *Operations Research Letters*, 40:212–217, 2012.

[13] B. Flach, L. Barroso, and M. Pereira. Long-term optimal allocation of hydro generation for a price-maker company in a competitive market: latest developments and a stochastic dual dynamic programming approach. *IET generation, transmission & distribution*, 4:299–314, 2010.

[14] P. Girardeau, V. Leclere, and A.B. Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40:130–145, 2015.

[15] V. Guigues. Robust production management. *Optimization and Engineering*, 10:505–532, 2009.

[16] V. Guigues. SDDP for some interstage dependent risk-averse problems and application to hydro-thermal planning. *Computational Optimization and Applications*, 57:167–203, 2014.

[17] V. Guigues. Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. *SIAM Journal on Optimization*, 26:2468–2494, 2016.

[18] V. Guigues. Inexact cuts for Deterministic and Stochastic Dual Dynamic Programming applied to convex nonlinear optimization problems. *arXiv*, 2017. https://arxiv.org/abs/1707.00812.

[19] V. Guigues, P. Ndiaye, F. Oustry, F. Romanet, and R. Aïd. Robust Mid-Term power generation management. *Optimization*, 58(3):351–371, 2009.

[20] V. Guigues and W. Römisch. Sampling-based decomposition methods for multistage stochastic programs based on extended polyhedral risk measures. *SIAM J. Optim.*, 22:286–312, 2012.

[21] V. Guigues and W. Römisch. SDDP for multistage stochastic linear programs based on spectral risk measures. *Operations Research Letters*, 40:313–318, 2012.

[22] V. Guigues, W. Tekaya, and M. Lejeune. Regularized decomposition methods for deterministic and stochastic convex optimization and application to portfolio selection with direct transaction and market impact costs. *Optimization OnLine*, 2017.

[23] M. Hindsberger and A. B. Philpott. Resa: A method for solving multi-stage stochastic linear programs. *SPIX Stochastic Programming Symposium*, 2001.

[24] G. Infanger and D. Morton. Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming*, 75:241–256, 1996.

[25] S. Jin, S.M. Ryan, J.-P. Watson, and D.L. Woodruff. Modeling and solving a large-scale generation expansion planning problem under uncertainty. *Energy Systems*, 2:209–242, 2011.

[26] V. Kozmik and D.P. Morton. Evaluating policies in risk-averse multi-stage stochastic programming. *Mathematical Programming*, 152:275–300, 2015.

[27] M.I. Kusy and W.T. Ziemba. A bank asset and liability management model. *Operations Research*, 34:356–376, 1986.

[28] Y. Liu, R. Sioshansi, and A.J. Conejo. Multistage stochastic investment planning with multiscale representation of uncertainties and decisions. *IEEE Transactions on Power Systems*, 31:781–791, 2018.

[29] B. Mo, A. Gjelsvik, and A. Grundt. Integrated risk management of hydro power scheduling and contract management. *IEEE Transactions on Power Systems*, 16:216–221, 2001.

[30] J. Mulvey and W.C. Kim. Multistage financial planning models: Integrated stochastic programs and policy simulators. *in "Stochastic Programming. The State of the art. In honor of G. Dantzig. International Series in Operations Research & Management Science"*, pages 257–275, 2011.

[31] J. Mulvey and B. Shetty. Financial planning via multi-stage stochastic optimization. *Computers & Operations Research*, 31:1–20, 2004.

[32] A. Ngai and M. Sherris. Longevity risk management for life and variable annuities: The effectiveness of static hedging using longevity bonds and derivatives. *Insurance: Mathematics and Economics*, 49:100–114, 2011.

[33] M.V.F. Pereira and L.M.V.G Pinto. Stochastic optimization of a multireservoir hydroelectric system: a decomposition approach. *Water Resources Research*, 21:779–792, 1985.

[34] M.V.F. Pereira and L.M.V.G Pinto. Multi-stage stochastic optimization applied to energy planning. *Math. Program.*, 52:359–375, 1991.

[35] G. Pflug and A. Pichler. *MultiStage Stochastic Optimization*. Springer Series in Operations Research and Financial Engineering, 2014.

[36] G. Pflug and W. Römisch. *Modeling, Measuring and Managing Risk*. World Scientific, Singapore, 2007.

[37] A. Philpott and V. de Matos. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of Operational Research*, 218:470–483, 2012.

[38] A. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Oper. Res. Lett.*, 36:450–455, 2008.

[39] A. Philpott and A. Shapiro. A tutorial on stochastic programming. 2007.

[40] W.P. Powell. *Approximate Dynamic Programming*. John Wiley and Sons, 2nd edition, 2011.

[41] A. Ruszczyński and A. Shapiro. *Stochastic Programming, Handbook in Operations Research and Management Science*. Elsevier, 2003.

[42] Zenios S.A. and W.T. Ziemba. *Handbook of Asset and Liability Management*. North Holland, 2007.

[43] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209:63–72, 2011.

[44] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, 2009.

[45] M. Singh and S. Faircloth. The impact of corporate debt on long term investment and firm performance. *Applied Economics*, 37:875–883, 2006.

[46] J. Zou, S. Ahmed, and X.A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming, to appear*.