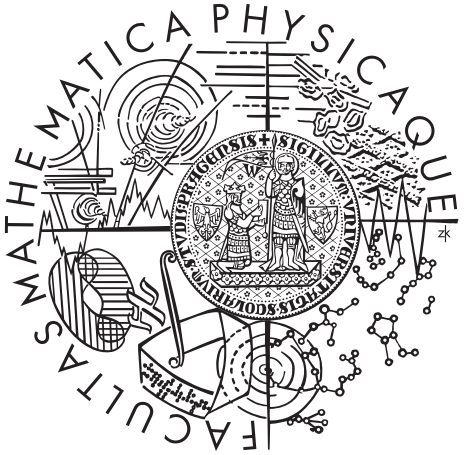
Charles University in Prague
Faculty of Mathematics and Physics

DOCTORAL THESIS



RNDr. Václav Kozmík

Multi-Stage Stochastic Programming with CVaR: Modeling, Algorithms and Robustness

Department of Probability and Mathematical Statistics

Supervisor of the doctoral thesis:	Prof. RNDr. Jitka Dupačová, DrSc.
Study programme:	Mathematics
Specialization:	Econometrics and Operations Research

Prague 2014

I would like to express my sincere gratitude to everybody who supported the development of this thesis. First of all, my thanks belong to my advisor, Prof. Jitka Dupačová, for her restless guidance and help throughout my studies. She was by far the best advisor I could have imagined, and I admire her for her knowledge, patience, great cooperation and care. My deep thanks also belong to Prof. David Morton for his excellent scientific support and hospitality during my stay at University of Texas in Austin. At the same time, I would also like to thank Mrs. Lída Urbánková for making my stay in the United States possible. Moreover, I would like to sincerely thank my consultant, Doc. Miloš Kopa, for his support, scientific help and assistance. Last of all, but not least, I am grateful for the help and support of my family along the way to finishing this thesis.

This work was partly supported by the Czech Science Foundation through grant 402/12/G097 "DYME/Dynamic Models in Economics". This thesis was also partly supported by the grant SVV-2014-260105.

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

Prague, November 18, 2014 RNDr. Václav Kozmík

Název práce: Vícestupňové stochastické programování s CVaR: modely, algoritmy a robustnost

Autor: RNDr. Václav Kozmík

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí disertační práce: Prof. RNDr. Jitka Dupačová, DrSc.

Abstrakt: Předložená práce formuluje tři vícestupňové modely stochastického programování, které jsou založené na míře rizika CVaR, a popisuje jejich vlastnosti včetně časové konzistence. Pro řešení těchto modelů se používá algoritmus stochastického duálního dynamického programování. Při použití vnořené míry rizika s CVaR chybí v současnosti spolehlivý postup na odhad účelové funkce. Náš nový postup, který je založen na technice simulace podle důležitosti, přináší spolehlivé výsledky a umožňuje kontrolu kvality řešení. Postup simulace podle důležitosti je dále zobecněn a lze jej použít pro redukci rozptylu ve všech modelech, které pracují s mírou rizika CVaR. Ke studiu robustnosti využíváme techniku kontaminace a rozšíříme ji pro úlohy s velkým počtem scénářů, pro které není možné nalézt přesné optimální řešení. Navržené postupy jsou ověřeny na numerických příkladech velkého rozsahu, které jsou založeny na jednoduchém vícestupňovém investičním modelu.

Klíčová slova: Vícestupňové stochastické programování, stochastické duální dynamické programování, simulace podle důležitosti, kontaminace, CVaR

Title: Multi-Stage Stochastic Programming with CVaR: Modeling, Algorithms and Robustness

Author: RNDr. Václav Kozmík

Department: Department of Probability and Mathematical Statistics

Supervisor: Prof. RNDr. Jitka Dupačová, DrSc.

Abstract: We formulate a multi-stage stochastic linear program with three different risk measures based on CVaR and discuss their properties, such as time consistency. The stochastic dual dynamic programming algorithm is described and its drawbacks in the risk-averse setting are demonstrated. We present a new approach to evaluating policies in multi-stage risk-averse programs, which aims to eliminate the biggest drawback – lack of a reasonable upper bound estimator. Our approach is based on an importance sampling scheme, which is thoroughly analyzed. A general variance reduction scheme for mean-risk sampling with CVaR is provided. In order to evaluate robustness of the presented models we extend contamination technique to the case of large-scale programs, where a precise solution cannot be obtained. Our computational results are based on a simple multi-stage asset allocation model and confirm usefulness of the presented procedures, as well as give additional insights into the behavior of more complex models.

Keywords: Multi-stage stochastic programming, stochastic dual dynamic programming, importance sampling, contamination, CVaR

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

Stochastic programming evolved from its deterministic counterpart by realizing that the parameters influencing the objective function and constraints are usually uncertain, coming from the real world around us. The first pioneering works of two-stage stochastic programming emerged 60 years ago, see Dantzig [23] and Beale [12]. Since that time, stochastic programming has grown substantially and its applications are widespread, from economy to biology or logistics and engineering. Various ways of managing uncertainty were considered, starting with the straightforward approach to replace random parameters with their expected values, continued with introduction of the two-stage concept. In the two-stage program, we choose a single first-stage decision and evaluate its expected performance in the second-stage program, which contains random parameters. Other significant developments include probabilistic constraints, sometimes called chance constraints, or the introduction of risk measures and dynamics to the stochastic programming. For a thorough introduction to the stochastic programming field we refer to the book by Shapiro et al. [86].

The two-stage concept has been generalized to multi-stage stochastic programming, which allows multiple stages of the decision and data process. Such models capture the dynamics of the underlying random process, and we are allowed to adjust our decisions based on the random parameters observed so far. Our decisions cannot depend on the parameters which are still uncertain and will be resolved in future stages, which means that the decisions fulfill the notion of nonanticipativity. Even though this generalization can be described in a straightforward manner, it brings substantial issues with theoretical properties, random process models and tractability. The applicability of such models often depends on the structure of the problem we are trying to solve. There are many applications where the two-stage models provide sufficient results and the generalizations are not necessary. On the other hand, multi-stage stochastic models have provided valuable improvements in many cases, usually involving some complex time-dependent structure. Such examples can be found, for instance, in finance, energy management or transportation.

There are two common ways to describe uncertainty in the stochastic programming models. The first approach is to collect some historical values or experts' opinion and produce a discrete distribution, which consists of scenarios with assigned probabilities. The second approach is to assume that the random inputs follow some continuous distribution and estimate its parameters from the data or use the experts' opinion to choose the parameters. When a continuous distribution is selected, sampling methods are commonly used to convert it to the discrete version in order to obtain a numerically tractable approximation. For large-scale problems, we are unable to compute precise solutions even for this discrete approximation. This brings us to the question of stability and robustness: we should consider what happens with the optimal solutions and objective values if our assumptions or estimates were imprecise or incorrect, or if our approximations were not accurate. Various questions of stability are discussed for example in Fiacco [39], Bonnans and Shapiro [18] or Römisch [75]. Most of the quantitative techniques rely on various assumptions on the objective function and

constraints or on the assumption of unique optimal solution. Widely applicable results are usually asymptotic, which brings the question of their validity for small size samples. We shall exploit the contamination technique, developed by Dupačová in papers [27, 29, 30], which provides a generally applicable nonasymptotic procedure which fits easily into the standard optimization framework. The first ideas dealing with contamination for multi-stage stochastic linear programs were presented in Dupačová [28] and their application to studying the influence of changes in the structure of multi-stage problems with polyhedral risk measures can be found in Dupačová et al. [35].

The extension of the random input models to the multi-stage case is rather complicated. Single scenarios from the discrete distribution are usually organized into a structure of a scenario tree. The tree structure represents the time dynamics of the model – we start in a single node, root scenario, which has multiple branches connected to the considered events for the next stage. Each of these branches contains a specific scenario of the considered events and is again connected to multiple scenarios of the following stage. This structure is repeated in the same manner up to the final stage, sometimes called horizon. If we relate the decisions with each node of the scenario tree, the condition of nonanticipativity is implicitly fulfilled. In such complex structure, many new questions arise. These include the specification of the branching size, e.g. how many descendant scenarios should be used for each stage, or specification of the corresponding conditional distributions, which have to be estimated with consideration of their time structure. Moreover, many of these techniques produce scenario trees of excessive sizes and procedures to reduce the number of scenarios have to be developed. There are many articles dealing with the issues discussed above, we refer for instance to Dupačová et al. [31, 33], Pflug [63], Pflug and Pichler [65] or Heitsch and Römisch [43].

Due to the complexity of stochastic programs, discussed for example in Shapiro and Nemirovski [85], approximations are often employed. Monte Carlo sampling and scenario approximations have been used even before they got a name of *Sample Average Approximation* in the article of Kleywegt et al. [50]. Approximate solutions depend on the particular set of sampled scenarios and are therefore random in general. Such approximate solutions require statistical validation, which is usually based on doing multiple replications and examining the stability of solutions and objective values, see Bayraksan and Morton [11] for a recent summary of available methods. In the multi-stage setting, modern algorithms often employ sampling techniques even in the case when discrete scenario tree is provided as the model for random inputs. Sampling techniques usually provide significant improvement in terms of computational speed, but validation procedures and convergence issues have to be thoroughly analyzed to provide reasonable guarantees about the solution quality.

Most of the stochastic programming models optimize the expected outcome of the random costs or returns. Resulting decisions are optimal on average, but possible risks are neglected. In many cases, this does not have to be an appropriate goal, as these decisions could produce a very unsatisfactory performance or even lead to bankruptcy under the worst-case scenarios. First developments in modeling risk aversion by using utility functions can be found in Bernoulli [14], or, more formal and precise description, in von Neumann and Morgenstern [59]. Other significant ways to producing more robust solutions include mean-risk

models. These bi-criteria models aim to find an efficient solution with respect to maximization of the mean return and minimization of the risk which is linked to the future uncertainty. Basics of the mean-risk concept using variance and semivariance as a measure of risk were published in the article [57] and book [58] by Harry Markowitz already in the 1950s. In recent years, risk-averse stochastic optimization based on various risk measures has received significant attention. The properties required of *coherent* risk measures, introduced in Artzner et al. [8], are now widely accepted for time-static risk-averse optimization. One of the most popular risk measures, Conditional Value at Risk (CVaR, see Rockafellar and Uryasev [74]), is known to satisfy these properties; for an overview of many others see, for instance, Krokhmal et al. [53].

A number of proposals have been put forward to extend the concept of coherent risk measures to handle multi-stage stochastic optimization. In addition to coherency, time consistency may be desired. There are multiple approaches to time consistency of risk measures and stochastic programs. In the multi-stage case we seek a *policy*, which specifies a decision rule at every stage t for any realization of the stochastic process up to time t. One of the time consistency concepts requires that optimal decisions at time t should not depend on future states of the system, which we already know cannot be realized, conditional on the state of the system at time t. Despite the natural statement of this requirement, there are a variety of risk-averse programs that fail to meet this condition. See Shapiro [82] and Rudloff et al. [77] for such examples, along with further discussions of why stochastic programs which are not time-consistent can produce unsatisfactory policies. On the other hand, it should be noted that some of the time consistency concepts can significantly reduce our choice of available risk measures, up to just risk-neutral case, see Shapiro [84] or Kupper and Schachermayer [54].

Scenario-based stochastic programs can often be reformulated as one large-scale standard optimization program, and such program can be solved directly by solvers like CPLEX [48], Gurobi [42] or COIN-OR [22]. The reformulated programs are usually very large and require long solving times or are unsolvable at all. This motivated the development of algorithms which exploit the special structure of stochastic programs and take advantage of particular properties like convexity. Optimization problems which include integer variables are known to be very hard to solve in general, and they are, of course, even more demanding in the stochastic setting. Most of the recent algorithms employ a technique of building so-called cuts on the feasible space or objective function. These cuts are used to eliminate infeasible or suboptimal decisions, or to approximate the objective function. The basic algorithm – Benders' decomposition, sometimes called L-shaped method, was developed by Benders [13], see also Van Slyke and Wets [89]. There have been many improvements of the basic algorithm, especially the multicut method by Birge and Louveaux [16], regularized decomposition by Ruszczyński [78] and stochastic decomposition by Higle and Sen [45]. These decomposition algorithms usually provide approximate solution and control its quality by computing lower and upper bounds on the true optimal objective value.

The structure of recourse functions in the multi-stage stochastic programs is particularly difficult from the algorithmic perspective. If we transform the multi-stage stochastic program into the dynamic programming recursion, the last stage program can be solved by the algorithms mentioned above. For the preceding

stages, we need to realize that the precise form of the recourse function cannot be obtained, and we need to rely only on its approximation, provided by the cuts. Therefore we are recursively accumulating approximation error, which leads to slower convergence and requires further validation of correctness. The basic multi-stage decomposition algorithm, Nested Benders' decomposition [15], applied to a multi-stage stochastic program requires computational effort that grows exponentially in the number of stages. Other important algorithms designed to solve multi-stage stochastic programs include extensions of stochastic decomposition to the multi-stage case [46, 88], progressive hedging [72] and stochastic dual dynamic programming (SDDP) [62]. SDDP will be used as the main solution technique for multi-stage stochastic programs in this thesis.

SDDP originated in the work of Pereira and Pinto [62], and inspired a number of related algorithms [20, 26, 56, 66], which aim to improve its efficiency. SDDP-style algorithms have computational effort per iteration that grows linearly in the number of stages. To achieve this, SDDP algorithms rely on the assumption of stage-wise independence. That said, SDDP algorithms can also be applied in some special cases of additive interstage dependence, such as when an autoregressive process, or a dynamic linear model, governs the right-hand side vectors [25, 49]. SDDP-style algorithms extend to handle other types of interstage dependency, such as combining the usual finer grain SDDP dependency with a coarser grain state of the system. See Philpott and Matos [67] for SDDP coupled with Markov chain and Rebennack et al. [70] for SDDP coupled with scenario tree. SDDP has been employed successfully in a range of applications, exhibiting good computational tractability on large-scale problem instances; see, e.g., [26, 40, 41, 47, 68, 70].

A standard multi-stage recourse formulation uses an additive form of expected utility. In this case, the usual upper bound estimator in SDDP algorithms is computed by solving subproblems along linear sample paths through the scenario tree, and the resulting computational effort is linear in the product of the number of stages and the number of samples. As we describe below, this type of estimator performs poorly for a model with a nested CVaR risk measure, and this has hampered application of SDDP to such nested risk-averse formulations. We are aware of three approaches that have been proposed in the literature to circumvent this difficulty. First, we can solve a risk-neutral version of the problem instance under some suitable termination criterion and determine the number of iterations needed to reach the solution. Then we run the SDDP algorithm again to solve the risk-averse model under nested CVaR and stop after this fixed number of iterations. Philpott and de Matos [67] report good computational experience with this approach. However, this leaves open the question of whether the same number of iterations is always appropriate for both risk-neutral and risk-averse model instances. Second, we can compute an upper bound estimator via the conditional sampling method of Shapiro [83]. However, the associated computational effort grows exponentially in the number of stages, and as Shapiro [83] discusses, the bound can be loose. Third, a non-statistical deterministic upper bound is proposed in Philpott et al. [69] based on using an inner approximation scheme. This approach is attractive in that it does not have sampling-based error, but as discussed in [69], the upper bound does not scale well as the number of state variables grows.

As mentioned above, multi-stage risk-averse stochastic problems are complex from many perspectives. The first major issue concerns the model itself – we need to design the model structure, decision steps and constraints. Moreover, we have to select appropriate objective function and risk measure, while basic properties, like non-anticipativity, coherence and time consistency, should be considered. With the introduction of risk measures to the objective function, tractability plays even more important role in the context of multi-stage stochastic programming and its applications. For CVaR-type risk measures, one of the biggest issues is policy evaluation, which has been mentioned above: standard solution procedures involving Monte Carlo sampling schemes are usually adjusted for the risk-neutral setting and their sampling schemes can perform poorly in the risk-averse setting. The stability concerns are even more involved, because they usually require additional computation than just a single run of the model. In this thesis, we will discuss the mentioned issues of model selection, policy evaluation and stability considerations in a thorough detail and present some novel approaches to dealing with them.

In Chapter 1, which is based on the article [4], we formulate a multi-stage stochastic linear program with CVaR risk measure in various versions. With the first version, nested CVaR, we follow the same manner as Shapiro [83], largely following his notation. Secondly, we formulate a multi-stage stochastic program with a multi-period CVaR risk measure, following the notion of [64]. The last approach comes from the techniques of multi-objective optimization and consists of a sum of single-period CVaR risk measures, see again [64]. We will discuss differences and similarities of the three models as well as their time consistency properties. SDDP algorithm and procedures for calculation of the upper bound will be described in Chapter 2.

In Chapter 3 we propose, analyze, and computationally demonstrate a new upper bound estimator for SDDP algorithms under a nested CVaR risk measure, based on article [3]. The computational effort required to form our estimator grows linearly in the number of time stages, and it is not limited to models with a modest number of state variables. Moreover, our estimation procedure fits flawlessly in the standard SDDP framework and our bound is significantly tighter than the estimator based on conditional sampling, which further facilities application of natural termination criteria. Such termination criteria are usually based on comparing the difference between an upper bound estimator and the lower bound. That said, our estimation procedure is not turnkey. Rather, it requires specification of functions that can appropriately characterize the tail of the recourse function, as we formalize in our main results and illustrate with an asset allocation model, cf. [2].

The CVaR risk measure computes the expected shortfall below the specified quantile level. As we will demonstrate on simple examples, standard Monte Carlo sampling schemes produce too many scenarios which do not belong to the part of the distribution that determines the CVaR value. An alternative sampling scheme is presented in Chapter 4, as proposed in [5]. This scheme rectifies the imbalance mentioned above and provides better results in terms of a reduced variance. Computational effort required to use our scheme is only slightly higher than the one required for the standard Monte Carlo scheme. The results of this development are further applicable in the improved SDDP upper bound estimator

and lead to additional variance reduction over the standard setup.

As discussed above, many approximations and estimates are involved in multi-stage stochastic programs, and we are usually unable to provide precise solutions. Therefore, it is important to test stability and robustness of our results. However, standard contamination bounds cannot be used directly in large-scale applications. In Chapter 5, which is based on article [4], we extend standard contamination bounds to include only lower bounds and upper bounds of the optimal values, which are available through the SDDP algorithm and its improvements. Numerical results with an asset allocation problem provide sufficiently tight bounds that can be used in practical applications to test stability.

In Chapter 6 we present an illustrative numerical study to demonstrate the power of our multi-stage risk-averse model for a simple asset allocation problem. Our multi-stage model is based on the article [2], and it is an extension of the two-stage results presented in the article [1], where we presented good empirical performance of SAA under the log-normal distribution. Our model is in line with the latest research in asset allocation, focused on dynamic models which allow rebalancing the portfolio multiple times before the final investment horizon is reached; we refer to the book by Dupačová et al. [32] for an introduction to more complex stochastic decision models in finance. Wide range of different investment strategies can be covered by our model, because the risk-aversion settings can be adjusted separately for each stage. We assume that the price ratios observed in stock market follow log-normal distribution. We estimate its parameters from the data and apply our model for various number of stages and risk-aversion settings. The resulting decisions are compared and summarized, following the article [2]. Moreover, with reference to the article [6], we provide an additional comparison of the nested and multi-period CVaR model.

The ideas we present in this thesis are described mostly on the nested CVaR model in the context of SDDP algorithm. We believe that these ideas could be useful in the extensions of SDDP mentioned above and also in other algorithms for multi-stage stochastic programs. When sampling of mean-risk functionals with CVaR is required, many algorithms may benefit from the procedures described below. Moreover, our extension of contamination bounds does not apply only to the models with CVaR or SDDP, but it can be used for any multi-stage stochastic problem where we are able to compute lower and upper bounds. While it is beyond the scope of this thesis, the ideas behind our proposed estimators and sampling schemes may also apply to other models with CVaR-style risk measures, for instance spectral risk measures based on finite combination of CVaR risk measures, cf. Acerbi [7]. We conclude and discuss ideas for future research in the last chapter of this thesis.

1. Multi-stage stochastic models

1.1 Multi-stage stochastic programs

We first formulate a linear multi-stage stochastic program with an expectation in the objective function. We suppose that the model has random parameters in stages $t = 2, \ldots, T$, denoted $\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)$, which are governed by a known, or well-estimated, distribution. The parameters of the first stage, $\boldsymbol{\xi}_1 = (\boldsymbol{c}_1, \boldsymbol{A}_1, \boldsymbol{b}_1)$, are assumed to be known when we make decision $\boldsymbol{x}_1$, but only a probability distribution governing future realizations, $\boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_T$, is known. The realization of $\boldsymbol{\xi}_2$ is known when decisions $\boldsymbol{x}_2$ must be made and so on up to stage T. The term $\boldsymbol{B}_t\boldsymbol{x}_{t-1}$ captures the state of the system. For example, in an asset allocation model, the state could be the total value of the assets in our portfolio. The components $\boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_T$ of $\boldsymbol{\xi}$ and the decisions $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$ are assumed to be random vectors, not necessarily of the same dimension, defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, while $\boldsymbol{\xi}_1$ is deterministic and $\boldsymbol{x}_1$ is a nonrandom vector-valued variable. The sequence of decisions and observations is

$$\boldsymbol{x}_1, \boldsymbol{\xi}_2, \boldsymbol{x}_2(\boldsymbol{x}_1, \boldsymbol{\xi}_2), \ldots, \boldsymbol{x}_T(\boldsymbol{x}_{T-1}, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_T). \tag{1.1}$$

The decision process is *nonanticipative* which means that decisions taken at any stage of the process depend neither on future *realizations* of stochastic data nor on future decisions, whereas the past information as well as the knowledge of the probability distribution of the data process can be exploited. In a mathematical way, let $\mathcal{F}_t \subseteq \mathcal{F}$ be the σ-field generated by the projection $\Pi_t\boldsymbol{\xi} = \boldsymbol{\xi}_{[t]} := (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_t)$ of the stochastic data process $\boldsymbol{\xi}$ that includes data up to stage t, $\mathcal{F}_1 = \{\emptyset, \Omega\}$ is the trivial σ-field. The dependence of the t-th stage decision $\boldsymbol{x}_t$ only on the available information means that $\boldsymbol{x}_t$ is $\mathcal{F}_t$-measurable. Similarly we let $\Pi_t\boldsymbol{x} = \boldsymbol{x}_{[t]} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ denote the sequence of decisions at stages $1, \ldots, t$, $\mathbb{P}$ the probability distribution of $\boldsymbol{\xi}$, $\mathbb{P}_t$ denotes the marginal probability distribution of $\boldsymbol{\xi}_t$, and $\mathbb{P}_t\left[\cdot|\boldsymbol{\xi}_{[t-1]}\right]$, $t = 2, , \ldots, T$, its conditional probability distribution. An illustrative scheme of a scenario tree can be found in Figure 1.1.

The first stage decisions consist of all decisions that have to be selected before further information is revealed whereas the second stage decisions are allowed to adapt to this information, etc. In each of the stages, the decisions are limited by constraints that may depend only on the previous decisions and observations. *Stages do not have to coincide with the observations of the random parameters, they should rather correspond to steps in the decision process.* Besides that, time spans between stages do not have to be equal, which facilitates effective representation of problems with far horizon. The decision process (1.1) has a random outcome, represented by cost $f(\boldsymbol{x}, \boldsymbol{\xi})$, and the basic goal is to find a nonanticipative decision or policy $\boldsymbol{x}(\boldsymbol{\xi})$ which minimizes the expectation $\mathbb{E}_{\mathbb{P}}\left[f(\boldsymbol{x}, \boldsymbol{\xi})\right]$ and satisfies all prescribed constraints. The optimal decision will be denoted $\boldsymbol{x}^*$. If there are more optimal solutions, $\boldsymbol{x}^*$ is used to represent any member of the set of optimal solutions.

The basic form of a multi-stage stochastic program is the multi-stage stochastic linear program (MSLP). First, we present it in its nested form, which resembles the backward recursion of stochastic dynamic programming with an additive

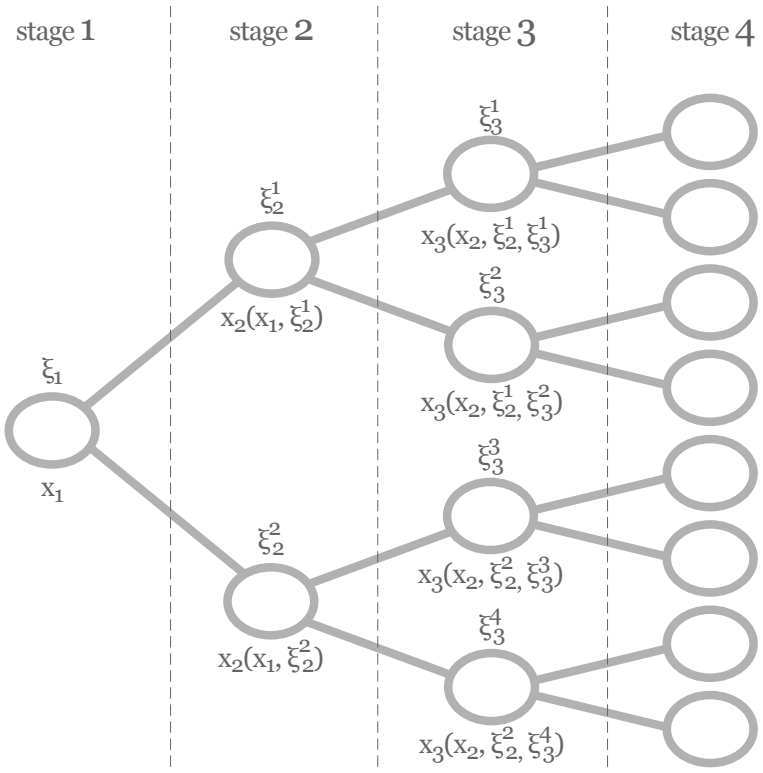


Figure 1.1: Scenario tree with associated decisions

overall cost function:

$$\min_{\boldsymbol{x}_1 \in \mathcal{X}_1} \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \mathbb{E}_{\mathbb{P}} \left[Q_2(\boldsymbol{x}_1, \boldsymbol{\xi}_{[2]}) \right] \text{ with } \mathcal{X}_1 := \{ \boldsymbol{x}_1 | \, \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1, \, \boldsymbol{x}_1 \geq 0 \}, \tag{1.2}$$

and $Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{[t]})$, $t = 2, \ldots, T$, defined recursively as

$$Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{[t]}) = \min_{\boldsymbol{x}_t} \boldsymbol{c}_t(\boldsymbol{\xi}_{[t-1]})^\top \boldsymbol{x}_t + \mathbb{E}_{\mathbb{P}_{t+1}[\cdot | \boldsymbol{\xi}_{[t]}]} \left[Q_{t+1}(\boldsymbol{x}_t, \boldsymbol{\xi}_{[t+1]}) \right] \tag{1.3}$$

subject to constraints $\boldsymbol{x}_t \in \mathcal{X}_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{[t]})$, e.g.

$$\boldsymbol{A}_t(\boldsymbol{\xi}_{[t-1]}) \boldsymbol{x}_t = \boldsymbol{b}_t(\boldsymbol{\xi}_{[t-1]}) - \boldsymbol{B}_t(\boldsymbol{\xi}_{[t-1]}) \boldsymbol{x}_{t-1}, \, \boldsymbol{x}_t \geq 0 \, \text{a.s.},$$

and $Q_{T+1}(\cdot)$ is explicitly given, e.g. $Q_{T+1}(\cdot) \equiv 0$.

Matrices $\boldsymbol{A}_t$ are of a fixed (m_t, n_t) type, and the remaining vectors and matrices are of consistent dimensions. For the first stage, known values of all elements of $\boldsymbol{c}_1, \boldsymbol{A}_1, \boldsymbol{b}_1$ are assumed, and the main decision variable is $\boldsymbol{x}_1$ that corresponds to the first stage. The first stage problem (1.2) has the form of the expectation-type stochastic program with the set of feasible decisions *independent* of $\mathbb{P}$.

One can rewrite (1.2)–(1.3) briefly as

$$\min_{\boldsymbol{x}_1} \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \mathbb{E} \left[\min_{\boldsymbol{x}_2} \boldsymbol{c}_2(\boldsymbol{\xi}_{[1]})^\top \boldsymbol{x}_2 + \mathbb{E} \left[\cdots + \mathbb{E} \left[\min_{\boldsymbol{x}_T} \boldsymbol{c}_T(\boldsymbol{\xi}_{[T-1]})^\top \boldsymbol{x}_T \right] \right] \right] \tag{1.4}$$

with corresponding conditional expectations as in (1.3) and subject to linear constraints $\boldsymbol{x}_t \in \mathcal{X}_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{[t]})$, $t = 1, \ldots, T$, on decision variables. The set of all feasible decisions $\mathcal{X}(\boldsymbol{\xi})$ is given by

$$\left\{ \boldsymbol{x} \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_T} : \boldsymbol{x}_1 \in \mathcal{X}_1, \; \boldsymbol{x}_2 \in \mathcal{X}_2(\boldsymbol{x}_1, \boldsymbol{\xi}_{[2]}), \; \ldots \; , \boldsymbol{x}_T \in \mathcal{X}_T(\boldsymbol{x}_{T-1}, \boldsymbol{\xi}_{[T]}) \right\}.$$

Constraints involving random elements are supposed to hold almost surely and for simplicity we will assume that *all optimal solutions exist*. A common assumption of *relatively complete recourse* requires that for each decision of the stage t, there exists some feasible decision for the stage $t+1$. Moreover, we will suppose that all conditional expectations exist. In the case of *stage-wise independence* the conditional probability distributions boil down to marginal distributions $\mathbb{P}_t$ of $\boldsymbol{\xi}_t$.

An example of a more complicated stochastic program could be an *expectation-based convex stochastic program*. The cost fuction $f(\boldsymbol{x}, \boldsymbol{\xi})$ is supposed to be an inf-compact convex normal integrand whose finite expectation exists and the set of feasible decisions $\mathcal{X}(\boldsymbol{\xi})$ should be closed, convex-valued, nonanticipative and uniformly bounded mapping, i.e. the assumption of relatively complete recourse. The relation to stochastic dynamic programming is further revealed by the fact that the optimal decisions can be obtained by telescoping the T stage problem into t stage ones, as done in Rockafellar and Wets [71]. The following proposition follows from Theorem 1 of [71] where it was formulated for extended real integrand $f(\boldsymbol{x}, \boldsymbol{\xi})$. (Here we assume explicitly formulated nonanticipative constraint mappings as well as existence of expectations and of optimal decisions.)

Theorem 1.1. *Consider the T stage stochastic program*

$$\text{minimize } \mathbb{E}_{\mathbb{P}}\left[f(\boldsymbol{x}, \boldsymbol{\xi})\right] \tag{1.5}$$

subject to constraints $\boldsymbol{x}_t \in \mathcal{X}_t(\boldsymbol{x}_{[t-1]}, \boldsymbol{\xi}_{[t]})$, $t = 1, \ldots, T$.
Put $f_T(\boldsymbol{x}_T, \boldsymbol{x}_{[T-1]}, \boldsymbol{\xi}_{[T]}) := f(\boldsymbol{x}, \boldsymbol{\xi})$ and for $t = 1, \ldots, T-1$ define the t-th stage integrands

$$f_t(\boldsymbol{x}_t, \boldsymbol{x}_{[t-1]}, \boldsymbol{\xi}_{[t]}) = \mathbb{E}_{\mathbb{P}}\left[\min_{\boldsymbol{x}_{[\tau]}}\{f_\tau(\boldsymbol{x}_\tau, \boldsymbol{x}_{[\tau-1]}, \boldsymbol{\xi}_{[\tau]}) : \Pi_t \boldsymbol{\xi}_{[\tau]} = \boldsymbol{\xi}_{[t]}, \Pi_t \boldsymbol{x}_{[\tau]} = \boldsymbol{x}_{[t]}\}\right]$$

for $t \le \tau \le T$ and consider the t stage problems

$$\text{minimize } \mathbb{E}_{\mathbb{P}}\left[f_t(\boldsymbol{x}_t, \boldsymbol{x}_{[t-1]}, \boldsymbol{\xi}_{[t]})\right] \tag{1.6}$$

subject to constraints $\boldsymbol{x}_t \in \mathcal{X}_t(\boldsymbol{x}_{[t-1]}, \boldsymbol{\xi}_{[t]})$. Then all programs (1.6) are solvable and the following property holds true:

If $\boldsymbol{x}^*$ is an optimal solution of (1.5) then its projection $\Pi_t \boldsymbol{x}^*$ solves (1.6) and if $\boldsymbol{x}^*_{[t]}$ solves (1.6), it can be extended to an optimal solution $\boldsymbol{x}^*$ of (1.5) such that $\Pi_t \boldsymbol{x}^* = \boldsymbol{x}^*_{[t]}$.

This theorem forms a bridge between multi-stage stochastic programs solved as a sequence of rolling horizon stochastic programs with a reduced number of stages and the stochastic dynamic programming methodology. For convex multi-stage expectation-based stochastic programs it can be evidently linked with the concept of *dynamic* or *time consistency property* introduced later on, e.g.

> **TC1** [19] The sequence of dynamic optimization problems (1.6) is dynamically consistent if the optimal strategies obtained when solving the original problem remain optimal for all subsequent problems.

When the normal integrand $f(\boldsymbol{x}, \boldsymbol{\xi})$ is separable with respect to stages, one can design an alternative dynamic programming recursion such as in (1.2)–(1.3), or as in Pennanen and Perkkiö [61] where a result similar to Theorem 1.1 can be found for convex multi-stage stochastic programs.

In applications one mostly approximates the true probability distribution $\mathbb{P}$ of $\boldsymbol{\xi}$ by a discrete probability distribution carried by a finite number of atoms (scenarios), say, $\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^K$. They are organized in form of a scenario tree and in principle, the optimal policy can be obtained by solving a large deterministic program. See e.g. the recent book [86] for details and more general cases. Every node of the tree is a root of a scenario subtree which does not contain any branches of other subtrees. Hence, the optimal solutions of a nodal subproblem do not reflect the future information carried by branches of the full tree that, from the point of view of the relevant nodal subproblem, cannot happen in the future. This observation is behind a modified time consistency concept

> **TC2** [82] At each state of the system, optimality of a decision policy should not involve states which cannot happen in the future.

Hence, under modest assumptions, optimal solutions of risk-neutral scenario-based multi-stage stochastic programs possess both of these time consistency properties, whereas there are still open questions concerning time consistency notions for risk-averse multi-stage stochastic programs. In the next sections, we shall discuss the risk-averse formulations and their properties, including time consistency.

1.2 Risk-averse multi-stage stochastic programs

Maximization of expected gains or minimization of expected losses means to get decisions that are optimal on average while possible risks are neglected. This need not be an acceptable goal. The present tendency is to spell out explicitly the concern for risk monitoring and control. There are various types of risk and the choice of a suitable risk definition depends on the context, on the decision maker's attitude, the company goals, etc.

To reflect risks in the stochastic programming formulation, it is necessary to quantify them. Both in theoretical considerations and in applications, rational properties of risk measures are requested. A risk measure is a functional which assigns a real value to the random outcome $f(\boldsymbol{x}, \boldsymbol{\xi})$. Similarly as the risk-neutral expected value criterion, risk measures ρ should not depend on individual realizations of $\boldsymbol{\xi}$, but they depend on decisions and probability distribution $\mathbb{P}$. Moreover, they should also reflect the structure of the filtration $\mathcal{F}_1 \subset \cdots \subset \mathcal{F}_t \cdots \subseteq \mathcal{F}$.

Coherence of ρ (monotonicity, translation invariance, positive homogeneity and subadditivity) cf. [8] is mostly expected. Popular examples of risk measure ρ include Value at Risk (VaR), which is not coherent in general, and the Conditional Value at Risk (CVaR), which belongs to the class of coherent risk measures. Monotonicity with respect to the pointwise partial ordering and subadditivity are straightforward requirements, coming from the principles of risk quantification. Convexity is important to keep a manageable structure of the problem both for computational and theoretical purposes. Polyhedral property, cf. CVaR in Rockafellar and Uryasev [74], or polyhedral risk measures, cf. Eichhorn and

Römisch [38], facilitate the application in scenario-based programs with linear constraints, because we can rely on linear programming techniques.

Whereas there are many suggestions of risk measures for static stochastic programs, which have performed well in numerical experiments and applications, see e.g. [53] and references therein, the situation is much more involved for multi-stage problems. The first idea is to replace the expectation $\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\boldsymbol{\xi})]$ by a suitable risk measure ρ and to keep all constraints including nonanticipativity. Assigning a risk measure ρ to the final outcome $f(\boldsymbol{x},\boldsymbol{\xi})$ does not take into account the information structure given by the filtration. It corresponds to monitoring risk only at the horizon which need not be sufficient. To include risk monitoring in individual stages, one may relate the risk measure to the partial outcomes $f_1(\boldsymbol{x}_1)$, $f_t(\boldsymbol{x}_t,\boldsymbol{x}_{[t-1]},\boldsymbol{\xi}_{[t]})$, $t=2,\ldots,T$. Different risk measures ρ_t can be applied in individual stages. As a result we may construct objective function of the form

$$\min_{\boldsymbol{x}_1} \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \rho_2\left(\min_{\boldsymbol{x}_2} \boldsymbol{c}_2(\boldsymbol{\xi}_{[1]})^\top \boldsymbol{x}_2 + \rho_3\left(\cdots + \rho_T\left(\min_{\boldsymbol{x}_T} \boldsymbol{c}_T(\boldsymbol{\xi}_{[T-1]})^\top \boldsymbol{x}_T\right)\right)\right) \quad (1.7)$$

and use it in the place of (1.4). It is important to agree on acceptable properties of risk measures, usually at least convexity should be satisfied. The full formulation of the risk-averse stochastic program (1.7) has to include the nonanticipativity constraints. Depending on the risk-averse problem and on the applied solution technique, a form of time consistency of optimal solutions is desirable.

Having in mind tractable numerical techniques such as stochastic dual dynamic programming (SDDP) [62] applied to Sample Average Approximation (SAA) of the underlying problem we shall focus on finite discrete probability distributions, and we will study mainly multi-period extensions of conditional value at risk or multi-period polyhedral risk measures. The next section comments on some important definitions.

1.3 Basic definitions

We will model the risk by representing the loss which could be incurred in stages $1,\ldots,T$ by random functions $\boldsymbol{Z}=(Z_1,\ldots,Z_T)$ that will be defined on a suitable linear space $\mathcal{Z}$. The notion of *coherent* risk measures was introduced in Artzner et al. [8] and is now widely accepted in static risk-averse optimization. Following definition, based on Artzner et al. [9] and Eichhorn and Römisch [38], extends this notion to the multi-stage case.

Definition 1.2 (Multi-period risk measures)**.**
A functional ρ on $\mathcal{Z}=\times_{t=1}^{T}\mathcal{L}_p(\Omega,\mathcal{F}_t,\mathbb{P})$ with $p\in[1,\infty]$ is called a multi-period coherent risk measure if it satisfies the following:

1. *$Z_t \geq \tilde{Z}_t$ a.s., $t=1,\ldots,T \Longrightarrow \rho(Z_1,\ldots,Z_T) \geq \rho\left(\tilde{Z}_1,\ldots,\tilde{Z}_T\right)$ (monotonicity);*
2. *for each $r\in\mathbb{R}$: $\rho(Z_1+r,\ldots,Z_T+r)=\rho(Z_1,\ldots,Z_T)+r$ (translation invariance);*
3. *$\rho\left(\mu Z_1+(1-\mu)\tilde{Z}_1,\ldots,\mu Z_T+(1-\mu)\tilde{Z}_T\right) \leq$*
   $\leq \mu\rho(Z_1,\ldots,Z_T)+(1-\mu)\rho\left(\tilde{Z}_1,\ldots,\tilde{Z}_T\right)$ for $\mu\in[0,1]$ (convexity);

4. $\rho(\mu Z_1, \ldots, \mu Z_T) = \mu\rho(Z_1, \ldots, Z_T)$ *for* $\mu \geq 0$ *(positive homogeneity).*

Two special classes of multi-period risk measures have received a lot of attention, polyhedral risk measures and conditional risk mappings. Polyhedral risk measures are defined as the optimal value of a multi-stage stochastic program in the following way, see Eichhorn and Römisch [38].

Definition 1.3 (Multi-period polyhedral risk measures)**.**
A risk measure ρ *on* $\times_{t=1}^{T}\mathcal{L}_p(\Omega, \mathcal{F}_t, \mathbb{P})$ *with* $p \in [1, \infty]$ *is called multi-period polyhedral if there are* $k_t \in \mathbb{N}$, $c_t \in \mathbb{R}^{k_t}$, $t = 1, \ldots, T$, $w_{t,\tau} \in \mathbb{R}^{k_{t-\tau}}$, $t = 1, \ldots, T$, $\tau = 0, \ldots, t-1$, *a polyhedral set* $M_1 \subset R^{k_1}$, *and polyhedral cones* $M_t \subset R^{k_t}$, $t = 2, \ldots, T$, *such that*

$$\begin{aligned} \rho(\boldsymbol{Z}) = \inf\ & \mathbb{E}\left[\sum_{t=1}^{T} c_t^\top Y_t\right] \\ & s.t.\ Y_t \in \mathcal{L}_p(\Omega, \mathcal{F}_t, \mathbb{P}) \quad \forall t \in \{1, \ldots, T\} \\ & \quad Y_t \in M_t\ a.s. \quad \forall t \in \{1, \ldots, T\} \\ & \quad \sum_{\tau=0}^{t-1} w_{t,\tau}^\top Y_{t-\tau} = Z_t\ a.s. \quad \forall t \in \{1, \ldots, T\}. \end{aligned} \tag{1.8}$$

When replacing the expectation of the total outcome of a risk-neutral MSLP by the multi-period polyhedral risk measure it is possible to carry out the minimization with respect to the original decision variable $\boldsymbol{x}$ and minimization in (1.8) simultaneously, see Proposition 4.1 in [38]. Moreover, the scenario form of (1.8) and that of the combined problem is a linear program. Multi-period polyhedral risk measures are not coherent in general, but coherency can be obtained through a special choice of parameters, cf. [38].

The class of conditional risk mappings resembles the conditional expectations in (1.4). This structure is especially convenient for the construction of nested risk measures in order to obtain the time consistency property. Let $\mathcal{F} \subset \mathcal{F}'$ be σ-fields of subsets of Ω and $\mathcal{Z}$ and $\mathcal{Z}'$ be linear spaces of real-valued functions $f(\omega)$, $\omega \in \Omega$ measurable with respect to $\mathcal{F}$ and $\mathcal{F}'$, respectively. Following Ruszczyński and Shapiro [80] we define:

Definition 1.4 (Conditional risk mappings)**.**
We say that mapping $\rho : \mathcal{Z}' \to \mathcal{Z}$ *is a conditional risk mapping if the following properties hold:*

1. *Convexity. If* $\alpha \in [0, 1]$ *and* $X, Y \in \mathcal{Z}'$, *then*

$$\alpha\rho(X) + (1-\alpha)\rho(Y) \succeq \rho(\alpha X + (1-\alpha)Y).$$

2. *Monotonicity. If* $Y \succeq X$, *then* $\rho(Y) \succeq \rho(X)$.

3. *Predictable Translation Equivariance. If* $Y \in \mathcal{Z}$ *and* $X \in \mathcal{Z}'$, *then*

$$\rho(X + Y) = \rho(X) + Y.$$

The inequalities in 1. and 2. are understood component-wise, i.e., $Y \succeq X$ *means that* $Y(\omega) \geq X(\omega)$ *for every* $\omega \in \Omega$.

Note 1.5. *Predictable Translation Equivariance is similar to translation invariance from Definition 1.2, but in a more general setting. General definition of this assumption is still a subject of ongoing discussion and the two mentioned definitions are similar, but not equivalent. Based on our Predictable Translation Equivariance assumption, the addition of a real number in the first stage leads to following equation, see [86]:*

$$\text{for each } r \in \mathbb{R} : \rho\left(Z_1 + r, Z_2, \ldots, Z_T\right) = \rho\left(Z_1, \ldots, Z_T\right) + r,$$

compare with Definition 1.2.

For conditional risk mappings defined above we shall use notation $\rho\left(\cdot|\mathcal{F}\right)$. Using Predictable Translation Equivariance, we can construct *composite risk measures* as follows: Consider conditional risk mappings $\rho_2, \ldots, \rho_T$ and a risk function $\rho : \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_T \to \mathbb{R}$ given by:

$$\rho\left(Z_1, \ldots, Z_T\right) = Z_1 + \rho_2\left(Z_2 + \cdots \rho_{T-1}\left(Z_{T-1} + \rho_T\left(Z_T\right)\right)\right).$$

Using Predictable Translation Equivariance we get

$$\rho_{T-1}\left(Z_{T-1} + \rho_T\left(Z_T\right)\right) = \rho_{T-1} \circ \rho_T\left(Z_{T-1} + Z_T\right).$$

By continuing this process we end up with a composite risk measure $\bar{\rho} := \rho_2 \circ \cdots \circ \rho_T$. It holds

$$\bar{\rho}(Z_1 + \cdots + Z_T) = \rho\left(Z_1, \ldots, Z_T\right). \tag{1.9}$$

Using notation of Definition 1.4 we continue by introducing a concept of dynamic or time-consistent conditional risk mappings [52].

Definition 1.6 (Time-consistent risk mappings)**.**
A conditional risk mapping $\left(\rho_t\left(\cdot|\mathcal{F}_t\right)\right)_{t=1,\ldots,T}$ *is called time-consistent if for all* $1 \leq t_1 \leq t_2 \leq T$ *and* $X, Y \in \mathcal{L}_p\left(\Omega, \mathcal{F}, \mathbb{P}\right)$*:*

$$\rho_{t_2}\left(X|\mathcal{F}_{t_2}\right) \leq \rho_{t_2}\left(Y|\mathcal{F}_{t_2}\right) \Longrightarrow \rho_{t_1}\left(X|\mathcal{F}_{t_1}\right) \leq \rho_{t_1}\left(Y|\mathcal{F}_{t_1}\right).$$

There are various other related consistency concepts for risk mappings, see e.g. [76]; when demanded, they may substantially limit the choice of risk measures up to just the risk-neutral case, see e.g. [84]. In comparison with the time consistency concepts [TC1] and [TC2], which relate to the decisions, Definition 1.6 defines time consistency for the risk mapping itself. In order to evaluate the properties [TC1] and [TC2] we have to specify the subsequent optimization models for *every state of the system*. Without additional assumptions about the model structure, we cannot expect that time consistency of a risk mapping automatically guarantees time consistency of the model.

For scenario-based programs the time consistency property [TC1] holds true whenever it is possible to reformulate the risk-averse multi-stage stochastic problem into the form of a classical risk-neutral stochastic program. This is provided by Theorem 1.1 and holds true for instance in the case of multi-period polyhedral risk measures. It should be noted that such reformulations usually require additional decision variables and are therefore harder to solve than corresponding risk-neutral versions of these models.

1.4 Multi-stage stochastic programs with CVaR-type risk measures

We formulate a multi-stage stochastic linear program with CVaR risk measure in various versions. With the first version, nested CVaR, we follow the same manner as Shapiro [83], largely using his notation. Second, we formulate a multi-stage stochastic program with a multi-period CVaR risk measure and with a sum of CVaR risk measure following the notion of [64]. All models have random parameters in stages $t = 2, \dots, T$, e.g. $\boldsymbol{\xi}_t = \big(\boldsymbol{c}_t(\boldsymbol{\xi}_{[t-1]}), \boldsymbol{A}_t(\boldsymbol{\xi}_{[t-1]}), \boldsymbol{B}_t(\boldsymbol{\xi}_{[t-1]}), \boldsymbol{b}_t(\boldsymbol{\xi}_{[t-1]})\big)$ in (1.3), which are governed by a known conditional distribution. All models can be also formulated in a more general convex form (which is solvable using the SDDP algorithm), but we have chosen the linear versions for easier presentation of our results. For simplicity of notation we will drop the $(\boldsymbol{\xi}_{[t-1]})$ arguments and denote the random parameters only by $\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)$. The parameters of the first stage, $\boldsymbol{\xi}_1 = (\boldsymbol{c}_1, \boldsymbol{A}_1, \boldsymbol{b}_1)$, are assumed to be known. Our models allow specification of different risk aversion coefficients, $\lambda_t \in [0, 1]$, and confidence levels, $\alpha_t \in (0, 1)$, at each stage, $t = 1, \dots, T$.

1.4.1 Nested CVaR model

The nested CVaR model is based on the following composite risk measure [64], as considered in (1.9):

$$\rho^n(\boldsymbol{Z}) = \mathrm{CVaR}_\alpha\left[\cdot|\mathcal{F}_1\right] \circ \cdots \circ \mathrm{CVaR}_\alpha\left[\cdot|\mathcal{F}_{T-1}\right]\left(\sum_{t=1}^{T} Z_t\right).$$

According to [52] this risk measure is time-consistent with respect to the Definition 1.6. In order to provide a formulation of the nested model we introduce the following operator, which forms a weighted sum of conditional expectation and risk associated with random loss Z:

$$\rho_{t,\boldsymbol{\xi}_{[t-1]}}[Z] = (1-\lambda_t)\,\mathbb{E}\left[Z\,\middle|\,\boldsymbol{\xi}_{[t-1]}\right] + \lambda_t\,\mathrm{CVaR}_{\alpha_t}\left[Z\,\middle|\,\boldsymbol{\xi}_{[t-1]}\right]. \tag{1.10}$$

We suppose $\lambda_t \in [0, 1]$; with $\lambda_t = 0$ it covers the risk-neutral problems, whereas $\lambda_t = 1$ puts emphasis on risk control only. The case of $\lambda_t = 0$ for $t < T$ and $\lambda_T \neq 0$ models importance of risk only at the final stage. CVaR penalizes losses in the upper α tail of Z with a typical value of α being 0.05.

We can write the corresponding risk-averse linear multi-stage model with T stages in the following form:

$$\begin{aligned}\min_{\substack{\boldsymbol{A}_1\boldsymbol{x}_1=\boldsymbol{b}_1\\ \boldsymbol{x}_1\geq 0}} \boldsymbol{c}_1^\top\boldsymbol{x}_1 + \rho_{2,\boldsymbol{\xi}_{[1]}}\Bigg[\min_{\substack{\boldsymbol{A}_2\boldsymbol{x}_2=\boldsymbol{b}_2-\boldsymbol{B}_2\boldsymbol{x}_1\\ \boldsymbol{x}_2\geq 0}} \boldsymbol{c}_2^\top\boldsymbol{x}_2 + \cdots \\ \cdots + \rho_{T,\boldsymbol{\xi}_{[T-1]}}\left[\min_{\substack{\boldsymbol{A}_T\boldsymbol{x}_T=\boldsymbol{b}_T-\boldsymbol{B}_T\boldsymbol{x}_{T-1}\\ \boldsymbol{x}_T\geq 0}} \boldsymbol{c}_T^\top\boldsymbol{x}_T\right]\Bigg].\end{aligned} \tag{1.11}$$

We assume model (1.11) is feasible, has relatively complete recourse, and has a finite optimal value.

Our model, with the nested risk measure, allows a dynamic programming formulation to be developed, as is described in [83]. Using in (1.10) the definition of conditional value at risk from [74],

$$\mathrm{CVaR}_{\alpha}\left[Z\right] = \min_{u}\left(u + \frac{1}{\alpha}\mathbb{E}\left[Z-u\right]_{+}\right), \tag{1.12}$$

where $\left[\,\cdot\,\right]_{+} \equiv \max\{\,\cdot\,, 0\}$, we can rewrite (1.11) as

$$\begin{aligned} \min_{\boldsymbol{x}_1, u_1}\ & \boldsymbol{c}_1^{\top}\boldsymbol{x}_1 + \lambda_2 u_1 + \mathcal{Q}_2(\boldsymbol{x}_1, u_1, \boldsymbol{\xi}_{[1]}) \\ \text{s.t. }\ & \boldsymbol{A}_1\boldsymbol{x}_1 = \boldsymbol{b}_1 \\ & \boldsymbol{x}_1 \geq 0. \end{aligned} \tag{1.13}$$

The recourse value $Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{[t]})$ at stage $t = 2, \ldots, T$ is given by:

$$\begin{aligned} Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{[t]}) = \min_{\boldsymbol{x}_t, u_t}\ & \boldsymbol{c}_t^{\top}\boldsymbol{x}_t + \lambda_{t+1}u_t + \mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t]}) \\ \text{s.t. }\ & \boldsymbol{A}_t\boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t\boldsymbol{x}_{t-1} \\ & \boldsymbol{x}_t \geq 0, \end{aligned} \tag{1.14}$$

where

$$\begin{aligned} &\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t]}) = \\ &\quad = \mathbb{E}_{\mathbb{P}_{t+1}\left[\cdot|\boldsymbol{\xi}_{[t]}\right]}\left[(1-\lambda_{t+1})\,Q_{t+1}(\boldsymbol{x}_t, \boldsymbol{\xi}_{[t+1]}) + \frac{\lambda_{t+1}}{\alpha_{t+1}}\left[Q_{t+1}(\boldsymbol{x}_t, \boldsymbol{\xi}_{[t+1]}) - u_t\right]_{+}\right]. \end{aligned} \tag{1.15}$$

We take $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$ so that the objective function of model (1.14) reduces to $\boldsymbol{c}_T^{\top}\boldsymbol{x}_T$ when $t = T$; compare with (1.2)–(1.3).

The interpretation of the objective function is not straightforward, but it can be viewed as the real cost we would be willing to pay at the first stage instead of incurring the sequence of random costs $Z_1, \ldots, Z_T$; cf. [79].

The nested model is formulated in the framework of conditional risk mappings, and this formulation is time-consistent with respect to both [TC1] and [TC2], cf. [82]. However, due to its nesting structure, it cannot be represented as a polyhedral risk measure. We need the value of future recourse function $\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t]})$ from stage $t+1$ to calculate the value $Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{[t]})$ at time t. This is in contradiction with the definition of polyhedral risk measures, which allows only historical values to be used.

1.4.2 Multi-period CVaR model

The multi-period CVaR model is based on the following risk measure (see [64]):

$$\rho^{m}\left(\boldsymbol{Z}\right) = \sum_{t=2}^{T}\mu_t\mathbb{E}\left[\mathrm{CVaR}_{\alpha_t}\left[Z_t|\mathcal{F}_{t-1}\right]\right]. \tag{1.16}$$

with $\sum_{t=2}^{T}\mu_t = 1$, $\mu_t \geq 0\,\forall t$. The multi-period CVaR risk measure is time-consistent with respect to the Definition 1.6, according to the Theorem 3.3.11 of [52].

Using this risk measure and the mean-risk operator (1.10) we obtain a multi-period CVaR model:

$$\begin{aligned}
\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T} \; & \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \mu_2 \rho_{2,\boldsymbol{\xi}_{[1]}} \left[\boldsymbol{c}_2^\top \boldsymbol{x}_2\right] + \cdots + \mu_T \mathbb{E}\left[\rho_{T,\boldsymbol{\xi}_{[T-1]}}\left[\boldsymbol{c}_T^\top \boldsymbol{x}_T\right]\right] \\
\text{s.t. } & \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1 \\
& \boldsymbol{A}_2 \boldsymbol{x}_2 = \boldsymbol{b}_2 - \boldsymbol{B}_2 \boldsymbol{x}_1 \\
& \quad \vdots \\
& \boldsymbol{A}_T \boldsymbol{x}_T = \boldsymbol{b}_T - \boldsymbol{B}_T \boldsymbol{x}_{T-1} \\
& \boldsymbol{x}_t \geq 0, \; \boldsymbol{x}_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_t, \mathbb{P}\right), \quad t = 1, \ldots, T.
\end{aligned} \tag{1.17}$$

We assume model (1.17) is feasible, has relatively complete recourse, and has a finite optimal value. While $\rho_{2,\boldsymbol{\xi}_{[1]}}$ is deterministic, $\rho_{t,\boldsymbol{\xi}_{[t-1]}}$, $t = 3, \ldots, T$ are random variables and expectation is applied to get a meaningful model. The difference between this model and the nested CVaR model (1.11) is that here we apply expectation instead of the risk measure nesting. We also give a reformulation which uses the definition (1.12) of conditional value at risk and auxiliary variables q_t to express its nonlinear term:

$$\begin{aligned}
\min_{\boldsymbol{x}_t, u_t, q_t \forall t} \; & \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \sum_{t=1}^{T-1} \mu_{t+1} \mathbb{E}\left[\lambda_{t+1} u_t\right] + \sum_{t=2}^{T} \mu_t \mathbb{E}\left[\left(1 - \lambda_t\right) \boldsymbol{c}_t^\top \boldsymbol{x}_t + \frac{1}{\alpha_t} \lambda_t q_t\right] \\
\text{s.t. } & \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1 \\
& \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1}, \quad t = 2, \ldots, T \\
& q_t \geq \boldsymbol{c}_t^\top \boldsymbol{x}_t - u_{t-1}, \quad t = 2, \ldots, T \\
& q_t \geq 0, \quad t = 2, \ldots, T \\
& u_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_t, \mathbb{P}\right), \quad t = 1, \ldots, T-1 \\
& q_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_t, \mathbb{P}\right), \quad t = 2, \ldots, T \\
& \boldsymbol{x}_t \geq 0, \; \boldsymbol{x}_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_t, \mathbb{P}\right), \quad t = 1, \ldots, T.
\end{aligned} \tag{1.18}$$

As is shown in the book [64] the multi-period risk measure (1.16) is polyhedral. The multi-period CVaR model is therefore time-consistent with respect to [TC1]. Moreover, with reference to the dynamic programming equations developed below, multi-period CVaR model is time-consistent with respect to [TC2]. Other concepts of time consistency with this risk measure are discussed in an example in [9] and also in [52].

Note 1.7. *According to the Proposition 3.36 of [64], there is an inequality comparing the values of multi-period and nested* CVaR *risk measures. Under the assumption of $\alpha = \alpha_t \; \forall t$ we have that: $\rho^n\left(\boldsymbol{Z}\right) \leq \alpha^{-(T-2)} \rho^m\left(\boldsymbol{Z}\right)$. However, this bound can be loose for programs with large values of T. Besides that, it cannot be applied to the optimal values of our optimization problems, since they combine the value of the risk measure with the mean return.*

Similarly as in the case with nested CVaR model we develop dynamic programming equations. Contrary to the nested model, CVaR is now evaluated in the stochastic program which determines the recourse value. In consequence, notation of the recourse value $Q_t(\boldsymbol{x}_{t-1}, u_{t-1}, \boldsymbol{\xi}_{[t]})$ includes auxiliary variables u_t.

Using the interchangeability principle (see Theorem 14.60 of Rockafellar and Wets [73]) we get:

$$\begin{aligned} \min_{\boldsymbol{x}_1, u_1} \; & \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \mu_2 \lambda_2 u_1 + \mathcal{Q}_2(\boldsymbol{x}_1, u_1, \boldsymbol{\xi}_{[1]}) \\ \text{s.t.} \; & \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1 \\ & \boldsymbol{x}_1 \geq 0 \end{aligned} \tag{1.19}$$

with the recourse value $Q_t(\boldsymbol{x}_{t-1}, u_{t-1}, \boldsymbol{\xi}_{[t]})$ at stage $t = 2, \ldots, T$ given by:

$$\begin{aligned} & Q_t(\boldsymbol{x}_{t-1}, u_{t-1}, \boldsymbol{\xi}_{[t]}) = \\ & \quad = \min_{\boldsymbol{x}_t, u_t, q_t} \; \mu_t \left(1 - \lambda_t\right) \boldsymbol{c}_t^\top \boldsymbol{x}_t + \mu_{t+1} \lambda_{t+1} u_t + \mu_t \frac{1}{\alpha_t} \lambda_t q_t + \mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t]}) \\ & \qquad \text{s.t.} \; \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1} \\ & \qquad\quad q_t \geq \boldsymbol{c}_t^\top \boldsymbol{x}_t - u_{t-1} \\ & \qquad\quad q_t \geq 0 \\ & \qquad\quad \boldsymbol{x}_t \geq 0, \end{aligned} \tag{1.20}$$

where:

$$\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t]}) = \mathbb{E}_{\mathbb{P}_{t+1}\left[\cdot | \boldsymbol{\xi}_{[t]}\right]} \left[Q_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t+1]}) \right]. \tag{1.21}$$

We take $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$.

1.4.3 Sum of CVaR model

The weighted sum of CVaR model is based on the following risk measure (see [38]):

$$\rho^s(\boldsymbol{Z}) = \sum_{t=2}^{T} \mu_t \, \mathrm{CVaR}_{\alpha_t}[Z_t]$$

with $\sum_{t=2}^{T} \mu_t = 1$, $\mu_t \geq 0 \, \forall t$.

It can be shown that sum of CVaR is not a time-consistent risk measure with respect to the Definition 1.6, see [9, 77]. The sum of CVaR model does not include nesting of the recourse values. It can be deduced from the scalarization technique of the multiobjective optimization. Using (1.10) it reads

$$\begin{aligned} \min_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T} \; & \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \mu_2 \rho_{2, \boldsymbol{\xi}_{[1]}} \left[\boldsymbol{c}_2^\top \boldsymbol{x}_2 \right] + \cdots + \mu_T \rho_{T, \boldsymbol{\xi}_{[1]}} \left[\boldsymbol{c}_T^\top \boldsymbol{x}_T \right] \\ \text{s.t.} \; & \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1 \\ & \boldsymbol{A}_2 \boldsymbol{x}_2 = \boldsymbol{b}_2 - \boldsymbol{B}_2 \boldsymbol{x}_1 \\ & \quad \vdots \\ & \boldsymbol{A}_T \boldsymbol{x}_T = \boldsymbol{b}_T - \boldsymbol{B}_T \boldsymbol{x}_{T-1} \\ & \boldsymbol{x}_t \geq 0, \; \boldsymbol{x}_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_t, \mathbb{P}\right), \quad t = 1, \ldots, T. \end{aligned} \tag{1.22}$$

We assume again that model (1.22) is feasible, has relatively complete recourse, and has a finite optimal value. Please note that no nesting of the CVaR values is present and that we always condition the operator ρ with the first stage information $\boldsymbol{\xi}_{[1]}$, i.e. $\rho_{t, \boldsymbol{\xi}_{[1]}}$ is deterministic $\forall t = 2, \ldots, T$.

Using mean-risk operator (1.10) and auxiliary variables q_t to express the non-linear term in (1.12) we can rewrite the model as the following multi-stage stochastic linear program:

$$\begin{aligned}
\min_{\boldsymbol{x}_t, u_t, q_t \forall t} \; & \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \sum_{t=1}^{T-1} \mu_{t+1}\lambda_{t+1} u_t + \sum_{t=2}^{T} \mu_t \mathbb{E}\left[(1-\lambda_t)\, \boldsymbol{c}_t^\top \boldsymbol{x}_t + \frac{1}{\alpha_t}\lambda_t q_t \right] \\
\text{s.t. } & \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1 \\
& \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1}, \quad t = 2, \ldots, T \\
& q_t \geq \boldsymbol{c}_t^\top \boldsymbol{x}_t - u_{t-1}, \quad t = 2, \ldots, T \\
& q_t \geq 0, \quad t = 2, \ldots, T \\
& u_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_1, \mathbb{P}\right), \quad t = 1, \ldots, T-1 \\
& q_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_t, \mathbb{P}\right), \quad t = 2, \ldots, T \\
& \boldsymbol{x}_t \geq 0, \; \boldsymbol{x}_t \in \mathcal{L}_p\left(\Omega, \mathcal{F}_t, \mathbb{P}\right), \quad t = 1, \ldots, T.
\end{aligned} \tag{1.23}$$

It can be seen that the risk measure $\rho^s\left(\boldsymbol{Z}\right)$ used in this linear program satisfies requirements of Definition 1.3 and is therefore polyhedral and the corresponding optimization model is time-consistent under the Definition [TC1]. On the other hand, all variables u_t are decided in the first stage, and the model is therefore not time-consistent according to the Definition [TC2].

We again develop dynamic programming equations using the interchangeability principle (see Theorem 14.60 of Rockafellar and Wets [73]):

$$\begin{aligned}
\min_{\boldsymbol{x}_1, u_1, \ldots u_{T-1}} \; & \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \sum_{t=1}^{T-1} \mu_{t+1}\lambda_{t+1} u_t + \mathcal{Q}_2(\boldsymbol{x}_1, u_1, \ldots, u_{T-1}, \boldsymbol{\xi}_{[1]}) \\
\text{s.t. } & \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1 \\
& \boldsymbol{x}_1 \geq 0
\end{aligned} \tag{1.24}$$

with recourse value $Q_t(\boldsymbol{x}_{t-1}, u_{t-1}, \ldots, u_{T-1}, \boldsymbol{\xi}_{[t]})$ at stage $t = 2, \ldots, T$, given by:

$$\begin{aligned}
Q_t(\ldots) = \min_{\boldsymbol{x}_t, q_t} \; & \mu_t\left(1-\lambda_t\right) \boldsymbol{c}_t^\top \boldsymbol{x}_t + \mu_t \frac{1}{\alpha_t}\lambda_t q_t + \mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \ldots, u_{T-1}, \boldsymbol{\xi}_{[t]}) \\
\text{s.t. } & \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1} \\
& q_t \geq \boldsymbol{c}_t^\top \boldsymbol{x}_t - u_{t-1} \\
& q_t \geq 0 \\
& \boldsymbol{x}_t \geq 0,
\end{aligned} \tag{1.25}$$

where:

$$\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \ldots, u_{T-1}, \boldsymbol{\xi}_{[t]}) = \mathbb{E}_{\mathbb{P}_{t+1}\left[\cdot | \boldsymbol{\xi}_{[t]}\right]}\left[Q_{t+1}(\boldsymbol{x}_t, u_t, \ldots, u_{T-1}, \boldsymbol{\xi}_{[t+1]}) \right]. \tag{1.26}$$

We take $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$.

Other multi-period polyhedral risk measures and their comparison can be found in [38]. The final decision regarding which of the multi-period risk measures to choose depends on the solved problem.

1.5 Challenges in the risk-averse models

Risk-neutral models usually employ utility functions to represent the value of money for an individual, see Neumann and Morgenstern [59] for the formal definition of an additive utility. In contrast to a multi-stage formulation rooted in expected utility, our multi-stage models with CVaR have additional decision variables, u_t, which estimate the value-at-risk level. After introducing the auxiliary variables, the problem seems to be converted to a simpler case, involving only expectations of an additive utility. This impression may lead to the false conclusion that traditional algorithms can be applied. The nested nonlinearity arising from the positive-part function precludes this, as we illustrate in the next example.

Example 1.8. *Suppose we incur random costs Z_2 in the second stage and Z_3 in the third stage. Then under an additive utility with contribution $U_t(\cdot)$ in stage t, we have:*

$$\mathbb{E}\left[U_2(Z_2) + \mathbb{E}\left[U_3(Z_3)\,\middle|\,\boldsymbol{\xi}_{[2]}\right]\right] = \mathbb{E}\left[U_2(Z_2)\right] + \mathbb{E}\left[U_3(Z_3)\right].$$

However, this additive form does not hold under CVaR*. Using Predictable Translation Equivariance from Definition 1.4 we can write the composite risk measure as:*

$$\text{CVaR}_\alpha\left[Z_2 + \text{CVaR}_\alpha\left[Z_3\,\middle|\,\boldsymbol{\xi}_{[2]}\right]\right] = \text{CVaR}_\alpha\left[\text{CVaR}_\alpha\left[Z_2 + Z_3\,\middle|\,\boldsymbol{\xi}_{[2]}\right]\right],$$

but the composite risk measure does not lend itself to further simplification. Subadditivity of CVaR *yields*

$$\text{CVaR}_\alpha\left[Z_2 + \text{CVaR}_\alpha\left[Z_3\,\middle|\,\boldsymbol{\xi}_{[2]}\right]\right] \leq \text{CVaR}_\alpha\left[Z_2\right] + \text{CVaR}_\alpha\left[\text{CVaR}_\alpha\left[Z_3\,\middle|\,\boldsymbol{\xi}_{[2]}\right]\right],$$

which only bounds the risk measure and, even then, the composite measure still requires evaluation.

It is for the reasons illustrated in Example 1.8 that Philpott and de Matos [67] and Shapiro [83] point to the lack of a good upper bound estimator for model (1.11) when the problem has more than a very small number of stages. The natural conditional sampling estimator, discussed in [67, 83], has computational effort that grows exponentially in the number of stages. The following example points to a second issue associated with estimating CVaR.

Example 1.9. *Consider the following estimator of* $\text{CVaR}_\alpha\left[Z\right]$*, where $Z^1, Z^2, \ldots, Z^M$ are independent and identically distributed (i.i.d.) from the distribution of Z:*

$$\min_u \left(u + \frac{1}{\alpha M}\sum_{j=1}^{M}\left[Z^j - u\right]_+ \right).$$

If $\alpha = 0.05$ only about 5% of the samples contribute nonzero values to this estimator of CVaR*.*

The inefficiency pointed to in Example 1.9 compounds the computational challenges of forming a conditional sampling estimator of CVaR in the multi-stage setting. When forming an estimator of our risk measure from equation (1.10), this inefficiency means that, say, 95% of the samples are devoted to only estimating expected cost and the remaining 5% contribute to estimating both CVaR and expected cost. In Section 3.1 we propose an approach to upper bound estimation in the context of SDDP that rectifies this imbalance and has computational requirements that grow gracefully with the number of stages.

1.6 Stage independence

To apply SDDP algorithm, we have to assume $\boldsymbol{\xi}_t$, $t = 2, \dots, T$, to be stage-wise independent. It may seem that such assumption is overly restrictive, but this is not the case. See discussion in the introduction and further description in Chapter 2 to find very general models that can be covered under the assumption of stage independence. Moreover, for tractability and storage reasons, fully general models cannot be solved in practice for large number of stages.

The assumption of stage-wise independence further simplifies formulations of the presented models. In the nested CVaR model the function $\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t]})$ from equation (1.14) now takes the form $\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t)$. The dynamic programming equations are given by:

$$\begin{aligned} \min_{\boldsymbol{x}_1, u_1} \ & \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \lambda_2 u_1 + \mathcal{Q}_2(\boldsymbol{x}_1, u_1) \\ \text{s.t.} \ & \boldsymbol{A}_1 \boldsymbol{x}_1 = \boldsymbol{b}_1 \\ & \boldsymbol{x}_1 \geq 0 \end{aligned} \tag{1.27}$$

with the recourse value $Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$ at stage $t = 2, \dots T$ given by:

$$\begin{aligned} Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) = \min_{\boldsymbol{x}_t, u_t} \ & \boldsymbol{c}_t^\top \boldsymbol{x}_t + \lambda_{t+1} u_t + \mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t) \\ \text{s.t.} \ & \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1} \\ & \boldsymbol{x}_t \geq 0, \end{aligned} \tag{1.28}$$

where

$$\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t) = \mathbb{E}\left[(1 - \lambda_{t+1}) \, Q_{t+1}(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}) + \frac{\lambda_{t+1}}{\alpha_{t+1}} \left[Q_{t+1}(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}) - u_t \right]_+ \right]. \tag{1.29}$$

Similar development applies to the function $\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{[t]})$ from equation (1.21) and the function $\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \dots, u_{T-1}, \boldsymbol{\xi}_{[t]})$ from equation (1.26) which will be written as $\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t, \dots, u_{T-1})$ in the stage independent case.

1.7 Asset allocation model

1.7.1 Nested model

Our procedures will be demonstrated mostly on a simple asset allocation model, cf. [2]. We would like to emphasize that our primary purpose is to provide computational results for our techniques as opposed to building a high-fidelity model for practical use. At stage t the decisions $\boldsymbol{x}_t$ denote the allocations (in units of a multiple of a base currency, say USD), and $\boldsymbol{r}_t$ denotes the gross return per stage; i.e., the ratio of the price at stage t to that in stage $t - 1$. These represent the only random parameters in the model. Without transaction costs, model (1.28) specializes to:

$$Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) = \min_{\boldsymbol{x}_t, u_t} \quad -\mathbf{1}^\top \boldsymbol{x}_t + \lambda_{t+1} u_t + \mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t) \tag{1.30a}$$

$$\text{s.t.} \quad \mathbf{1}^\top \boldsymbol{x}_t = \boldsymbol{r}_t^\top \boldsymbol{x}_{t-1} \tag{1.30b}$$

$$\boldsymbol{x}_t \geq 0, \tag{1.30c}$$

with recourse fuction given by (1.29).

In the first stage the initial capital of the right-hand side of (1.30b) is assumed to be equal to 1 and because $-\mathbf{1}^\top \boldsymbol{x}_1$ is then identically -1, we drop this constant from the objective function. First stage optimal solution is given by a following program, compare again with (1.27):

$$\begin{aligned} \min_{\boldsymbol{x}_1, u_1} \ & \lambda_2 u_1 + \mathcal{Q}_2(\boldsymbol{x}_1, u_1) \\ \text{s.t.} \ & \mathbf{1}^\top \boldsymbol{x}_1 = 1 \\ & \boldsymbol{x}_1 \geq 0. \end{aligned} \tag{1.31}$$

We also include an extended version of the asset allocation model, which contains transaction costs. We consider the case in which transaction costs are proportional to the value of the assets sold or bought, and in particular that the fee is $f_t = 0.3\%$ of the transaction value. The first stage program remains in the form of (1.31), but we must modify the rebalancing equation between stage $t-1$ and stage t to include the transaction costs of $f_t \mathbf{1}^\top |\boldsymbol{x}_t - \boldsymbol{x}_{t-1}|$, where the $|\cdot|$ function applies component-wise. Linearizing we obtain the following special case of model (1.28):

$$\begin{aligned} Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) = \min_{\boldsymbol{x}_t, \boldsymbol{o}_t, u_t} \ & -\mathbf{1}^\top \boldsymbol{x}_t + \lambda_{t+1} u_t + \mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t) \\ \text{s.t.} \ & \mathbf{1}^\top \boldsymbol{x}_t + f_t \mathbf{1}^\top \boldsymbol{o}_t = \boldsymbol{r}_t^\top \boldsymbol{x}_{t-1} \\ & \boldsymbol{o}_t - \boldsymbol{x}_t \geq -\boldsymbol{x}_{t-1} \\ & \boldsymbol{o}_t + \boldsymbol{x}_t \geq \boldsymbol{x}_{t-1} \\ & \boldsymbol{x}_t \geq 0, \end{aligned} \tag{1.32}$$

with recourse fuction given by (1.29) and first-stage program given by (1.31).

1.7.2 Multi-period model

In addition, we include computational results for a second model, which is based on multi-period CVaR risk measure. The decisions, random parameters and notation remain the same as in the case of a nested model. We also suppose the same behavior of transaction costs, relative to the volume of traded assets. Under the assumption of stage independence, first stage of model (1.19) specializes to:

$$\begin{aligned} \min_{\boldsymbol{x}_1, u_1} \ & \lambda_2 u_1 + \mathcal{Q}_2(\boldsymbol{x}_1, u_1) \\ \text{s.t.} \ & \mathbf{1}^\top \boldsymbol{x}_1 = 1 \\ & \boldsymbol{x}_1 \geq 0 \end{aligned} \tag{1.33}$$

with the recourse value $Q_t(\boldsymbol{x}_{t-1}, u_{t-1}, \boldsymbol{\xi}_t)$ at stage $t = 2, \ldots, T$ given by:

$$
\begin{aligned}
&Q_t(\boldsymbol{x}_{t-1}, u_{t-1}, \boldsymbol{\xi}_t) = \\
&\quad = \min_{\boldsymbol{x}_t, \boldsymbol{o}_t, u_t, q_t} \; (1-\lambda_t)\, \boldsymbol{c}_t^\top \boldsymbol{x}_t + \lambda_{t+1} u_t + \frac{1}{\alpha_t} \lambda_t q_t + \mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t) \\
&\qquad \text{s.t. } \mathbf{1}^\top \boldsymbol{x}_t + f_t \mathbf{1}^\top \boldsymbol{o}_t = \boldsymbol{r}_t^\top \boldsymbol{x}_{t-1} \\
&\qquad\quad \boldsymbol{o}_t - \boldsymbol{x}_t \geq -\boldsymbol{x}_{t-1} \\
&\qquad\quad \boldsymbol{o}_t + \boldsymbol{x}_t \geq \boldsymbol{x}_{t-1} \\
&\qquad\quad q_t \geq \boldsymbol{c}_t^\top \boldsymbol{x}_t - u_{t-1} \\
&\qquad\quad q_t \geq 0 \\
&\qquad\quad \boldsymbol{x}_t \geq 0,
\end{aligned}
\tag{1.34}
$$

where:

$$
\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t) = \mathbb{E}\left[Q_{t+1}(\boldsymbol{x}_t, u_t, \boldsymbol{\xi}_{t+1}) \right]. \tag{1.35}
$$

We take $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$.

1.7.3 Numerical experiments

We have considered three different data sets in our allocation model. First set consists of the stock market indices DJA, NDX, NYA, and OEX and captures price ratios observed month-to-month from September 1985 until September 2011. The set consists of 311 observations and their summary can be found in the Table 1.1. Two other data sets are based on price data of the most important assets traded on the Prague Stock Exchange. For monthly data, it means the total of 33 observations from January 2009 till February 2012, the summary can be found in the Table 1.2. Last data set is based on weekly data from November 2007 to March 2012 with 229 observations. The week-to-week price ratios were adjusted to include stock dividends, and their basic characteristics are reported in the Table 1.3. We have fitted a multidimensional correlated log-normal distribution to all of the data sets to obtain a continuous distribution, which has been used independently for all stages of the considered problems. Illustrative numerical study using these data sets can be found in Chapter 6.

All numerical experiments were performed with empirical scenario trees constructed by Algorithm 2.1. While sampling from the log-normal distribution, we have used the polar method [51] for sampling of the underlying normal distributions. The L'Ecuyer random generator [55] was used to generate the required uniform random variables. We implemented the SDDP algorithm in C++ software, using CPLEX [48] and COIN-OR [22] to solve the required linear programs and Armadillo [60] library for matrix computations.

asset	**mean**	**std. deviation**
DJA	0.9949	0.0459
NDX	0.9967	0.0948
NYA	0.9954	0.0469
OEX	0.9986	0.0701

Table 1.1: US assets data summary

asset	mean	std. deviation
AAA	1.0290	0.1235
CETV	0.9984	0.2469
ČEZ	0.9990	0.0647
ERSTE GROUP BANK	1.0172	0.1673
KOMERČNÍ BANKA	1.0110	0.1157
ORCO	1.0085	0.2200
PEGAS NONWOVENS	1.0221	0.0863
PHILIP MORRIS ČR	1.0213	0.0719
TELEFÓNICA C.R.	0.9993	0.0595
UNIPETROL	1.0079	0.0843
VIENNA INSURANCE GROUP	1.0074	0.1100

Table 1.2: Czech monthly data summary

asset	mean	std. deviation
AAA	0.9980	0.0716
CETV	0.9929	0.0995
ČEZ	0.9994	0.0406
ERSTE GROUP BANK	0.9983	0.0795
KOMERČNÍ BANKA	1.0018	0.0543
ORCO	0.9899	0.0938
PEGAS NONWOVENS	0.9995	0.0398
PHILIP MORRIS ČR	1.0035	0.0368
TELEFÓNICA C.R.	1.0004	0.0266
UNIPETROL	0.9986	0.0506

Table 1.3: Czech weekly data summary

2. SDDP algorithm

2.1 Stochastic dual dynamic programming

We use stochastic dual dynamic programming to approximately solve models presented in Section 1.4. SDDP algorithm can be used to solve any convex multi-stage stochastic program, but we will restrict our description to the model (1.11) for simplicity. Similar development applies to the models (1.17) and (1.22). We assume model (1.11) is feasible, has relatively complete recourse, and has a finite optimal value. SDDP does not operate directly on model (1.11). Instead, we first form a sample average approximation (SAA) of model (1.11), and SDDP approximately solves that SAA. Thus in our context SDDP forms estimators by sampling within an empirical scenario tree. Algorithm 2.1 describes how we form the sampling-based scenario tree for the SAA. Then in the remainder of this thesis we restrict attention to solving that SAA via SDDP. See Shapiro [81] for a discussion of asymptotics of SAA for multi-stage problems, Philpott and Guan [66] for convergence properties of SDDP, and Chiralaksanakul and Morton [21] or Bayraksan and Morton [10] for procedures to assess the quality of SDDP-based policies.

To apply SDDP, we have to assume $\boldsymbol{\xi}_t$, $t = 2, \ldots, T$, to be stage-wise independent, which means that the distribution of $\boldsymbol{\xi}_t$ does not depend on the previous realizations of $\boldsymbol{\xi}_{[t-1]}$. We further assume that for each stage $t = 2, \ldots, T$ there is a known (possibly continuous) distribution P_t of $\boldsymbol{\xi}_t$ and that we have a procedure to sample i.i.d. observations from this distribution. Using this procedure we obtain a single empirical distribution for each stage, denoted $\hat{P}_t$, $t = 2, \ldots, T$, and the associated empirical scenario tree is interstage independent. The scenarios generated by this procedure are equally probable, but this is not required to apply SDDP.

Algorithm 2.1 (Sampling under interstage independence)**.**

1. *Let* $\boldsymbol{\xi}_1$ *denote the deterministic first stage parameters.*

2. *Sample* D_2 *i.i.d. observations* $\boldsymbol{\xi}_2^1, \ldots, \boldsymbol{\xi}_2^{D_2}$ *from* P_2*. These are the descendants of the first stage scenario (node)* $\{\boldsymbol{\xi}_1\}$*.*

3. *Sample* D_3 *i.i.d. observations* $\boldsymbol{\xi}_3^1, \ldots, \boldsymbol{\xi}_3^{D_3}$ *from* P_3*, independent of those formed in stage* 2*. Let these denote the same set of descendant nodes for each of the* $N_2 = D_2$ *nodes* $\{\boldsymbol{\xi}_1\} \times \left\{\boldsymbol{\xi}_2^1, \ldots, \boldsymbol{\xi}_2^{D_2}\right\}$*.*

   $\vdots$

t. *Sample* D_t *i.i.d. observations* $\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{D_t}$ *from* P_t*, independent of those formed in stages* $2, \ldots, t-1$*. Let these denote the same set of descendant nodes for each of the* $N_{t-1} = \prod_{i=2}^{t-1} D_i$ *nodes* $\{\boldsymbol{\xi}_1\} \times \left\{\boldsymbol{\xi}_2^1, \ldots, \boldsymbol{\xi}_2^{D_2}\right\} \times \cdots \times \left\{\boldsymbol{\xi}_{t-1}^1, \ldots, \boldsymbol{\xi}_{t-1}^{D_{t-1}}\right\}$*.*

   $\vdots$

T. *Sample* D_T *i.i.d. observations* $\boldsymbol{\xi}_T^1, \ldots, \boldsymbol{\xi}_T^{D_T}$ *from* P_T*, independent of those formed in stages* $2, \ldots, T-1$*. Let these denote the same set of descendant*

nodes for each of the $N_{T-1} = \prod_{i=2}^{T-1} D_i$ *nodes* $\{\boldsymbol{\xi}_1\} \times \left\{\boldsymbol{\xi}_2^1, \ldots, \boldsymbol{\xi}_2^{D_2}\right\} \times \cdots \times \left\{\boldsymbol{\xi}_{T-1}^1, \ldots, \boldsymbol{\xi}_{T-1}^{D_{T-1}}\right\}$.

Distribution $\hat{P}_t$ has D_t realizations and stage t has N_t scenarios, where $N_t = \prod_{i=2}^{t} D_i$. We construct the sampling-based scenario tree for SDDP by using the same set of D_t observations at stage t to form the descendant nodes of all N_{t-1} scenarios at stage $t-1$, where $N_{t-1} = \prod_{i=2}^{t-1} D_i$. The SDDP algorithm does not apply, for example, to a scenario tree in which we use a separate, independent set of i.i.d. observations $\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{D_t}$ for each of the stage $t-1$ nodes, because the resulting empirical scenario tree would not be stage-wise independent. Procedures for construction of trees with general dependency structures are much more involved than Algorithm 2.1. However, fully general forms of interstage dependency lead to inherent computational intractability as even the memory requirements to store a general sampled scenario tree grow exponentially in the number of stages. Therefore, we do not focus on such procedures and estimation of possible conditional distributions, but rather point to several tractable dependency structures in the introduction. Tractable dependency structures are typically rooted in some form of independent increments between stages; e.g., autoregressive models, moving-average models, and dynamic linear models [90].

We let $\hat{\Omega}_t$ denote the stage t sample space, where $|\hat{\Omega}_t| = N_t$. We use $j_t \in \hat{\Omega}_t$ to denote a stage t sample point, which we call a stage t scenario. We define the mapping $a(j_t) : \hat{\Omega}_t \to \hat{\Omega}_{t-1}$, which specifies the unique stage $t-1$ ancestor for the stage t scenario j_t. Similarly, we use $\Delta(j_t) : \hat{\Omega}_t \to 2^{\hat{\Omega}_{t+1}}$ to denote the set of descendant nodes for j_t, where $|\Delta(j_t)| = D_{t+1}$. The empirical scenario tree therefore has stage t realizations denoted $\boldsymbol{\xi}_t^{j_t}$, $j_t \in \hat{\Omega}_t$. At the last stage, we have $\boldsymbol{\xi}_T^{j_T}$, $j_T \in \hat{\Omega}_T$, and each stage T scenario corresponds to a full path of observations through each stage of the scenario tree. That is, given j_T, we recursively have $j_{t-1} = a(j_t)$ for $t = T, T-1, \ldots, 2$. For this reason and for notational simplicity, when possible, we suppress the stage T subscript and denote $j_T \in \hat{\Omega}_T$ by $j \in \hat{\Omega}$.

We briefly describe SDDP to give sufficient context for our results. For further details on SDDP, see [62, 83]. The simplest SDDP algorithm applies to the risk-neutral version of our model, which means setting $\lambda_t = 0$ for $t = 1, \ldots, T$ in equation (1.10) and model (1.11) or equivalently in (1.27)–(1.29). We denote the recourse value for the risk-neutral version of our model by $Q_t^N(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$, which for $t = 2, \ldots, T$, is given by:

$$\begin{aligned} Q_t^N(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) = \min_{\boldsymbol{x}_t} \; & \boldsymbol{c}_t^\top \boldsymbol{x}_t + \mathcal{Q}_{t+1}^N(\boldsymbol{x}_t) \\ \text{s.t.} \; & \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1} \\ & \boldsymbol{x}_t \geq 0, \end{aligned} \tag{2.1}$$

where

$$\mathcal{Q}_{t+1}^N(\boldsymbol{x}_t) = \mathbb{E}\left[Q_{t+1}^N(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1})\right], \tag{2.2}$$

and where $\mathcal{Q}_{T+1}^N(\cdot) \equiv 0$. The risk-neutral formulation is completed via model (1.27) with $\lambda_2 = 0$ and $\mathcal{Q}_2(\boldsymbol{x}_1, u_1)$ replaced by $\mathcal{Q}_2^N(\boldsymbol{x}_1)$.

During a typical iteration of SDDP, cuts have been accumulated at each stage. These represent a piecewise linear outer approximation of the expected future cost function, $\mathcal{Q}_{t+1}^N(\boldsymbol{x}_t)$. On a forward pass we sample a number of linear paths

through the tree by sampling from a uniform distribution of $j \in \hat{\Omega}$. As we solve a sequence of master programs (which we specify below) along these forward paths, the cuts that have been accumulated so far are used to form decisions at each stage. Solutions found along a forward path in this way form a policy, which does not anticipate the future. In fact, the solutions can be found at a node on a sample path via the stage t master program, even before we sample the random parameters at stage $t+1$. The sample mean of the costs incurred along all the forward sampled paths through the tree forms an estimator of the expected cost of the current policy determined by the master programs.

In the backward pass of the algorithm, we add cuts to the collection defining the current approximation of the expected future cost function at each stage. We do this by solving subproblems at the descendant nodes of each node in the linear paths from the forward pass, except in the final stage, T. The cuts collected at any node in stage t apply to all the nodes in that stage, and hence we maintain a single set of cuts for each stage. We let C_t denote the number of cuts accumulated so far in stage t. This reduction is possible because of our interstage independence assumption. Every cut, based on a decision $\boldsymbol{x}_t^j$, is represented by the average value of descendant recourse functions $\hat{\mathcal{Q}}_{t+1}^j$ and average of their subgradients $\mathbf{g}_{t+1}^j$. An illustrative scheme of a single iteration of the SDDP algorithm can be found in Figure 2.1.

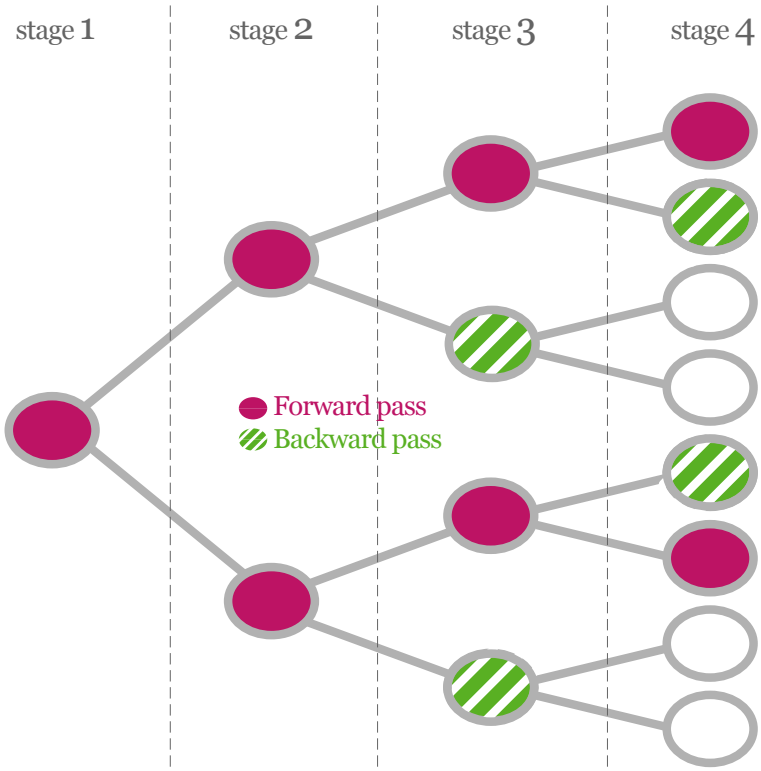


Figure 2.1: SDDP algorithm scheme

Model (2.3) acts as a master program for its stage $t+1$ descendant scenarios and acts as a subproblem for its stage $t-1$ ancestor:

$$\hat{Q}_t = \min_{\boldsymbol{x}_t, \theta_t} \quad \boldsymbol{c}_t^\top \boldsymbol{x}_t + \theta_t \tag{2.3a}$$

$$\text{s.t.} \quad \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1} \quad : \boldsymbol{\pi}_t \tag{2.3b}$$

$$\theta_t \geq \hat{\mathcal{Q}}_{t+1}^j + \left(\mathbf{g}_{t+1}^j\right)^\top \left(\boldsymbol{x}_t - \boldsymbol{x}_t^j\right), \quad j = 1, \ldots, C_t \tag{2.3c}$$

$$\boldsymbol{x}_t \geq 0. \tag{2.3d}$$

Decision variable θ_t in the objective function (2.3a), coupled with cut constraints in (2.3c), forms the outer linearization of the recourse function $\mathcal{Q}_{t+1}^N(\boldsymbol{x}_t)$ from model (2.1) and equation (2.2). The structural and nonnegativity constraints in (2.3b) and (2.3d) simply repeat the same constraints from model (2.1). In the final stage T, we omit the cut constraints and the θ_T variable. While we could append an "N" superscript on terms like $\hat{Q}_t$, $\hat{\mathcal{Q}}_{t+1}^j$, $\mathbf{g}_{t+1}^j$, etc., we suppress this index for notational simplicity.

As we indicate in constraint (2.3b), we use $\boldsymbol{\pi}_t$ to denote the dual vector associated with the structural constraints. Let j_t denote a stage t scenario from a sampled forward path. With $\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{a(j_t)}$ and with $\boldsymbol{\xi}_t = \boldsymbol{\xi}_t^{j_t}$ in model (2.3), we refer to that model as sub(j_t). Given model sub(j_t) and its solution $\boldsymbol{x}_t$, we form one new cut constraint at stage t for each backward pass of the SDDP algorithm as follows. We form and solve sub(j_{t+1}), where $j_{t+1} \in \Delta(j_t)$ indexes all descendant nodes of j_t. This yields optimal values $\hat{Q}_{t+1}^{j_{t+1}}(\boldsymbol{x}_t)$ and dual solutions $\boldsymbol{\pi}_{t+1}^{j_{t+1}}$ for $j_{t+1} \in \Delta(j_t)$. We then form

$$\boldsymbol{g}_{t+1}^{j_{t+1}} = -\left(\boldsymbol{B}_{t+1}^{j_{t+1}}\right)^\top \boldsymbol{\pi}_{t+1}^{j_{t+1}}, \tag{2.4}$$

where $\boldsymbol{g}_{t+1}^{j_{t+1}} = \boldsymbol{g}_{t+1}^{j_{t+1}}(\boldsymbol{x}_t)$ is a subgradient of $\hat{Q}_{t+1}^{j_{t+1}} = \hat{Q}_{t+1}^{j_{t+1}}(\boldsymbol{x}_t)$. The cut is then obtained by averaging over the descendants:

$$\hat{\mathcal{Q}}_{t+1} = \frac{1}{D_{t+1}} \sum_{j_{t+1} \in \Delta(j_t)} \hat{Q}_{t+1}^{j_{t+1}} \tag{2.5}$$

$$\mathbf{g}_{t+1} = \frac{1}{D_{t+1}} \sum_{j_{t+1} \in \Delta(j_t)} \boldsymbol{g}_{t+1}^{j_{t+1}}. \tag{2.6}$$

As we indicate above, both $\hat{Q}_{t+1}$ and $\boldsymbol{g}_{t+1}$ depend on $\boldsymbol{x}_t$, but we suppress this dependency for notational simplicity. We also suppress the j_t index on $\hat{\mathcal{Q}}_{t+1}$ and $\mathbf{g}_{t+1}$, because we append a new cut to the stage t collection of cuts and do not associate it with a particular stage t subproblem. Note that from the manner in which we express constraint (2.3c), it may appear as if we must keep track of the solution, $\boldsymbol{x}_t^j$, at which we form the cut, but this is not the case. Rather we store the term,

$$\hat{\mathcal{Q}}_{t+1}^j - \left(\mathbf{g}_{t+1}^j\right)^\top \boldsymbol{x}_t^j,$$

as a scalar intercept term for each cut. For simplicity in stating the SDDP algorithm below, we assume we have known lower bounds L_t on the recourse functions.

Algorithm 2.2 (Stochastic dual dynamic programming)**.**

1. *Let iteration $k = 1$ and append lower bounding cuts $\theta_t \geq L_t$, $t = 1, \ldots, T-1$.*

2. *Solve the stage 1 master program (2.3), $t = 1$, and obtain $\boldsymbol{x}_1^k, \theta_1^k$. Let $\underline{z}_k = \boldsymbol{c}_1^\top \boldsymbol{x}_1^k + \theta_1^k$.*

3. *Forward pass: sample i.i.d. paths from $\hat{\Omega}$ and index them by S^k.*

   For all $j \in S^k$ {
   For $t = 2, \ldots, T$ {
   Form and solve $\mathrm{sub}(j_t)$ *given by (2.3) to obtain $\left(\boldsymbol{x}_t^{j_t}\right)^k$;*
   }
   }

   Form the upper bound estimator:

$$\overline{z}_k = \boldsymbol{c}_1^\top \boldsymbol{x}_1^k + \frac{1}{|S^k|} \sum_{j \in S^k} \sum_{t=2}^{T} (\boldsymbol{c}_t^{j_t})^\top \left(\boldsymbol{x}_t^{j_t}\right)^k. \tag{2.7}$$

4. *If a stopping criterion, given $\overline{z}_k$ and $\underline{z}_k$, is satisfied then stop and output first stage solution $\boldsymbol{x}_1 = \boldsymbol{x}_1^k$ and lower bound $\underline{z} = \underline{z}_k$.*

5. *Backward pass:*

   For $t = T-1, \ldots, 1$ {
   For all $j \in S^k$ {
   For all descendant nodes $j_{t+1} \in \Delta(j_t)$ {
   Form and solve $\mathrm{sub}(j_{t+1})$ *given by (2.3) to obtain $\hat{Q}_{t+1}^{j_{t+1}}$ and $\boldsymbol{\pi}_{t+1}^{j_{t+1}}$;*
   Calculate $\boldsymbol{g}_{t+1}^{j_{t+1}}$ using formula (2.4);
   }
   Calculate optimal value $\hat{\mathcal{Q}}_{t+1}$ using equation (2.5);
   Calculate cut gradient $\mathfrak{g}_{t+1}$ using equation (2.6);
   Append the resulting cut to the collection (2.3c) for stage t;
   }
   }

6. *Let $k = k + 1$ and goto step 2.*

See Bayraksan and Morton [11] and Homem-de-Mello et al. [47] for stopping rules that can be employed in step 4.

2.2 Risk-averse approach

The SDDP algorithm of Section 2.1 must be modified to handle the risk-averse model (1.11). The auxiliary variables u_t now play a role both in computing the cuts and in determining the policy from the master programs. In the modified SDDP algorithm we select the VaR level, u_t, along with our stage t decisions, $\boldsymbol{x}_t$, and then solve the subproblems at the descendant nodes. The VaR level influences

the value of the recourse function estimate and therefore is included in the cuts, in the same way as any other decision variable. Extending the development from the previous section, the stage t subproblem in the risk-averse case is given by:

$$\begin{aligned}\hat{Q}_t = \min_{\boldsymbol{x}_t, u_t, \theta_t} \; & \boldsymbol{c}_t^\top \boldsymbol{x}_t + \lambda_{t+1} u_t + \theta_t \\ \text{s.t. } & \boldsymbol{A}_t \boldsymbol{x}_t = \boldsymbol{b}_t - \boldsymbol{B}_t \boldsymbol{x}_{t-1} \quad : \boldsymbol{\pi}_t \\ & \theta_t \geq \hat{\mathcal{Q}}_{t+1}^j + \left(\boldsymbol{\mathfrak{g}}_{t+1}^j\right)^\top \left[\begin{pmatrix} \boldsymbol{x}_t \\ u_t \end{pmatrix} - \begin{pmatrix} \boldsymbol{x}_t^j \\ u_t^j \end{pmatrix} \right], \quad j = 1, \ldots, C_t \\ & \boldsymbol{x}_t \geq 0.\end{aligned} \tag{2.8}$$

While $\hat{Q}_{t+1} = \hat{Q}_{t+1}(\boldsymbol{x}_t)$, we now have $\hat{\mathcal{Q}}_{t+1} = \hat{\mathcal{Q}}_{t+1}(\boldsymbol{x}_t, u_t)$ and $\boldsymbol{\mathfrak{g}}_{t+1} = \boldsymbol{\mathfrak{g}}_{t+1}(\boldsymbol{x}_t, u_t)$ as a function of both the stage t decision and the VaR level. The subgradient of $\hat{Q}_{t+1}(\boldsymbol{x}_t)$ is still computed by equation (2.4). However, the function value $\hat{\mathcal{Q}}_{t+1}(\boldsymbol{x}_t, u_t)$ and its subgradient have to be adjusted to reflect the differences between the risk-neutral and risk-averse cases. Shapiro et al. [87] describe an approach in which the VaR level does not appear explicitly in model (2.8). However, as we describe below, our upper bound estimators require explicit VaR levels.

As in Section 2.1, we let j_t denote a stage t scenario from a sample path. We let sub(j_t) denote model (2.8) when we set $\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{a(j_t)}$ and $\boldsymbol{\xi}_t = \boldsymbol{\xi}_t^{j_t}$. Given sub$(j_t)$ and its solution $(\boldsymbol{x}_t, u_t)$, we form a new cut constraint at stage t as follows. We form and solve sub(j_{t+1}), where $j_{t+1} \in \Delta(j_t)$ indexes all descendant nodes of j_t. This yields optimal values $\hat{Q}_{t+1}^{j_{t+1}}$ and dual solutions $\boldsymbol{\pi}_{t+1}^{j_{t+1}}$, along with subgradients $\boldsymbol{g}_{t+1}^{j_{t+1}}$ via equation (2.4), for $j_{t+1} \in \Delta(j_t)$. The sample mean variant of equation (1.29) then yields:

$$\hat{\mathcal{Q}}_{t+1} = \frac{1}{D_{t+1}} \sum_{j_{t+1} \in \Delta(j_t)} \left[(1 - \lambda_{t+1}) \, \hat{Q}_{t+1}^{j_{t+1}} + \frac{\lambda_{t+1}}{\alpha_{t+1}} \left[\hat{Q}_{t+1}^{j_{t+1}} - u_t \right]_+ \right]. \tag{2.9}$$

To compute a subgradient of $\hat{\mathcal{Q}}_{t+1}(\boldsymbol{x}_t, u_t)$ we must employ the chain rule for subdifferentials to deal with the positive-part operator. Following [83] this leads to

$$\boldsymbol{\mathfrak{g}}_{t+1} = \frac{1}{D_{t+1}} \begin{pmatrix} (1 - \lambda_{t+1}) \sum_{j_{t+1} \in \Delta(j_t)} \boldsymbol{g}_{t+1}^{j_{t+1}} + \frac{\lambda_{t+1}}{\alpha_{t+1}} \sum_{j_{t+1} \in J_{t+1}} \boldsymbol{g}_{t+1}^{j_{t+1}} \\ -\frac{\lambda_{t+1}}{\alpha_{t+1}} |J_{t+1}| \end{pmatrix}, \tag{2.10}$$

where the index set

$$J_{t+1} = \left\{ j_{t+1} : \hat{Q}_{t+1}^{j_{t+1}} > u_t, \; j_{t+1} \in \Delta(j_t) \right\}.$$

In modifying the SDDP algorithm for the risk-averse formulation, equations (2.9) and (2.10) replace equations (2.5) and (2.6) in the backward pass of step 5 of Algorithm 2.2 to provide the piecewise linear outer approximation of $\mathcal{Q}_{t+1}(\boldsymbol{x}_t, u_t)$. One issue that remains concerns evaluation of an upper bound; i.e., an analog of estimator (2.7) for the risk-averse setting. As we illustrate in Example 1.8, we cannot expect an analogous additive estimator to be appropriate for the risk-averse setting. Example 1.8 suggests that to compute the conditional risk measure, we should start at the last stage and recurse back to the first stage to obtain an estimator of the risk measure evaluated at a policy. This differs significantly from the

risk-neutral case, where the costs incurred at any stage can be estimated just by averaging costs at sampled nodes. Starting at the final stage, T, our cost under scenario j_T is $(\boldsymbol{c}_T^{j_T})^\top \boldsymbol{x}_T^{j_T}$. For the stage $T-1$ ancestor scenario $j_{T-1} = a(j_T)$ we must calculate

$$(\boldsymbol{c}_{T-1}^{j_{T-1}})^\top \boldsymbol{x}_{T-1}^{j_{T-1}} + \lambda_T u_{T-1}^{j_{T-1}} + \mathcal{Q}_T(\boldsymbol{x}_{T-1}^{j_{T-1}}, u_{T-1}^{j_{T-1}}).$$

The question that remains is how to estimate $\mathcal{Q}_T(\boldsymbol{x}_{T-1}^{j_{T-1}}, u_{T-1}^{j_{T-1}})$. We maintain a parallel with the estimator in the risk-neutral version of the SDDP algorithm in the sense that we estimate this term using the value of one descendant scenario along the corresponding forward path in step 3 of Algorithm 2.2. This means that based on equation (1.29) we estimate $\mathcal{Q}_T(\boldsymbol{x}_{T-1}^{j_{T-1}}, u_{T-1}^{j_{T-1}})$ by

$$(1-\lambda_T)\,(\boldsymbol{c}_T^{j_T})^\top \boldsymbol{x}_T^{j_T} + \frac{\lambda_T}{\alpha_T}\left[(\boldsymbol{c}_T^{j_T})^\top \boldsymbol{x}_T^{j_T} - u_{T-1}^{j_{T-1}}\right]_+.$$

Removing the expectation operator in equation (1.29), the associated recursion of the objective function in model (1.28) and equation (1.29) yields the following recursive estimator of $\mathcal{Q}_t(\boldsymbol{x}_{t-1}^{j_{t-1}}, u_{t-1}^{j_{t-1}})$ for $t = 2, \ldots, T$:

$$\begin{aligned}\hat{\mathfrak{v}}_t(\boldsymbol{\xi}_{t-1}^{j_{t-1}}) = (1-\lambda_t)\left((\boldsymbol{c}_t^{j_t})^\top \boldsymbol{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}(\boldsymbol{\xi}_t^{j_t})\right) + \\ \lambda_t u_{t-1}^{j_{t-1}} + \frac{\lambda_t}{\alpha_t}\left[(\boldsymbol{c}_t^{j_t})^\top \boldsymbol{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}(\boldsymbol{\xi}_t^{j_t}) - u_{t-1}^{j_{t-1}}\right]_+,\end{aligned} \tag{2.11}$$

where $\hat{\mathfrak{v}}_{T+1}(\boldsymbol{\xi}_T^{j_T}) \equiv 0$. We denote the estimator for sample path j by

$$\hat{\mathfrak{v}}(\boldsymbol{\xi}^j) = \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \hat{\mathfrak{v}}_2. \tag{2.12}$$

Because the first stage parameters, $\boldsymbol{\xi}_1^{j_1}$, are deterministic, we simply write $\hat{\mathfrak{v}}_2 = \hat{\mathfrak{v}}_2(\boldsymbol{\xi}_1^{j_1})$, dropping its argument.

Having selected scenario j and solved all node subproblems (2.8) associated with realizations $\boldsymbol{\xi}_1^{j_1}, \ldots, \boldsymbol{\xi}_T^{j_T}$ along the sample path, we form the estimator recursively as follows. We start at the stage $T-1$ node, compute $\hat{\mathfrak{v}}_T(\boldsymbol{\xi}_{T-1}^{j_{T-1}})$, substitute it into formula (2.11) for $t = T-1$ to obtain $\hat{\mathfrak{v}}_{T-1}(\boldsymbol{\xi}_{T-2}^{j_{T-2}})$ and so on until we obtain $\hat{\mathfrak{v}}_2$ and hence can compute the value of $\hat{\mathfrak{v}}(\boldsymbol{\xi}^j)$ via equation (2.12). Then if $\boldsymbol{\xi}^j$, $j = 1, \ldots, M$, are i.i.d. sample paths, sampled from the scenario tree's empirical distribution as in step 3 in Algorithm 2.2, the corresponding upper bound estimator is given by:

$$U^{\mathbf{n}} = \frac{1}{M}\sum_{j=1}^{M} \hat{\mathfrak{v}}(\boldsymbol{\xi}^j). \tag{2.13}$$

We use the "$\mathbf{n}$" superscript to indicate that we use naive Monte Carlo sampling here, and to distinguish it from estimators we develop below.

We can attempt to use estimator (2.13), which is the natural analog of (2.7), to solve the risk-averse problem. Unfortunately, this estimator has large variance. The main shortcoming of this estimator lies in the imbalance in sampled scenarios we point to in Example 1.9 coupled with the policy now specifying an approximation of the value at risk level via u_{t-1}. If the descendant node has value less than u_{t-1} then the positive-part term in equation (2.11) is zero. When the opposite occurs, the difference between the node value and u_{t-1} is multiplied by

α_t^{-1}, which can lead to a large value of the estimator because a typical value of α_t^{-1} is 20. When $\hat{\mathfrak{v}}_t(\boldsymbol{\xi}_{t-1}^{j_{t-1}})$ is large, this increases the likelihood that preceding values are also large and hence multiplied by $\alpha_{t-1}^{-1}, \alpha_{t-2}^{-1}, \ldots$ many more times in the backward recursion. This leads to a highly variable estimator which is of little practical use, particularly when T is not small.

To overcome the issues we have just discussed, Shapiro [83] describes an estimator which uses more nodes to estimate the recourse value. This estimator for three-stage problems is obtained by sampling and solving subproblems associated with i.i.d. realizations $\boldsymbol{\xi}_2^1, \ldots, \boldsymbol{\xi}_2^{M_2}$ from the second stage, and for each of these solving subproblems associated with i.i.d. realizations $\boldsymbol{\xi}_3^1, \ldots, \boldsymbol{\xi}_3^{M_3}$ from the third stage in order to estimate the future risk measure. This requires solving subproblems at a total of $M_2 M_3$ nodes. More generally under this approach, given a stage $T-1$ scenario $\boldsymbol{\xi}_{T-1}^{j_{T-1}}$, we estimate the recourse function value by:

$$\hat{\mathfrak{v}}_T^{\mathbf{e}}(\boldsymbol{\xi}_{T-1}^{j_{T-1}}) = \frac{1}{M_T} \sum_{j_T=1}^{M_T} \left[(1-\lambda_T) \left(\boldsymbol{c}_T^{j_T}\right)^\top \boldsymbol{x}_T^{j_T} + \lambda_T u_{T-1}^{j_{T-1}} + \frac{\lambda_T}{\alpha_T} \left[\left(\boldsymbol{c}_T^{j_T}\right)^\top \boldsymbol{x}_T^{j_T} - u_{T-1}^{j_{T-1}} \right]_+ \right].$$

For stages $t = 2, \ldots, T-1$ we have:

$$\begin{aligned} \hat{\mathfrak{v}}_t^{\mathbf{e}}(\boldsymbol{\xi}_{t-1}^{j_{t-1}}) = \frac{1}{M_t} \sum_{j_t=1}^{M_t} \Big[& (1-\lambda_t) \left(\left(\boldsymbol{c}_t^{j_t}\right)^\top \boldsymbol{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}^{\mathbf{e}}(\boldsymbol{\xi}_t^{j_t}) \right) \\ & + \lambda_t u_{t-1}^{j_{t-1}} + \frac{\lambda_t}{\alpha_t} \left[\left(\boldsymbol{c}_t^{j_t}\right)^\top \boldsymbol{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}^{\mathbf{e}}(\boldsymbol{\xi}_t^{j_t}) - u_{t-1}^{j_{t-1}} \right]_+ \Big]. \end{aligned} \tag{2.14}$$

And finally for the upper bound estimator we compute:

$$U^{\mathbf{e}} = \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \hat{\mathfrak{v}}_2^{\mathbf{e}}. \tag{2.15}$$

Shapiro [83] discusses two significant problems with the upper bound estimator (2.15). First, the estimator requires solving an exponential number, $\prod_{t=2}^T M_t$, of subproblems in the number of stages (thus the "$\mathbf{e}$" superscript) and hence is impractical unless T is small. Second, as we examine further in Section 3.2, even when we can afford to compute the bound provided by (2.15), the bound is not very tight. For these reasons, estimator (2.15) is not typically used in practice.

Philpott and de Matos [67] mention another approach. They avoid computing an upper bound for the risk-averse model by first solving the risk-neutral version of the problem, in which one can compute reliable upper bound estimators and hence employ a reasonable termination criterion. When the SDDP algorithm stops, we fix the number of iterations needed to satisfy the termination criterion. We then form the risk-averse model and run the SDDP algorithm, without evaluating an upper bound estimator. The solution and corresponding lower bound obtained after that fixed number of iterations are considered the algorithm's output. However, this approach has some pitfalls. It is unclear that the number of iterations for the risk-averse model should be the same as in the risk-neutral case, because the shape of the cost-to-go functions differs. This approach gives us no guarantees on the quality of the solution and requires that we run the SDDP algorithm twice.

To our knowledge, the most effective procedure currently available to compute an upper bound is proposed by Philpott et al. [69]. They develop an inner

approximation scheme that provides a candidate policy and a deterministic upper bound on the policy's value, using a convex combination of feasible policies. This bound provides significantly better results than estimator (2.15), and it does not have sampling error. However, as Philpott et al. [69] discuss, its main drawback is that the computational effort increases rapidly in the dimension of the decision variables. Applicability of the type of Monte Carlo estimators we propose tends to scale more gracefully with dimension. We further discuss the approach of [69] in Section 3.2.

When we restrict attention to statistical upper bound estimators, we have three possible approaches at this point. The two upper bound estimators (2.13) and (2.15) are available in the risk-averse case, and there is also a heuristic based on solving the risk-neutral model to determine the stopping iteration. In our view, all three of these approaches are unsatisfactory. We are either forced to use loose upper bounds that lead to very weak guarantees on solution quality and scale poorly. Or, we are forced to use the third approach which provides no guarantees on solution quality, even if reasonable empirical performance has been reported in the literature. In the next chapter we propose a new upper bound estimator to overcome these difficulties. Our estimator scales better with the number of stages and can yield greater precision than the previous approaches.

3. Policy evaluation with CVaR

3.1 Improved upper bound estimation

To overcome the shortcomings of the upper bound estimators (2.13) and (2.15) let us first focus on the main issue causing the estimators to be poor: A relatively small fraction of the sampled scenario-tree nodes contributes to estimating CVaR, for reasons we illustrate in Example 1.9. To sample in a better manner we assume that for every stage, $t = 2, \ldots, T$, we can cheaply evaluate a real-valued *approximation function*, $a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$, which estimates the recourse value of our decisions $\boldsymbol{x}_{t-1}$ after the random parameters $\boldsymbol{\xi}_t$ have been observed.

The functions a_t play a central role in our proposal for sampling descendant nodes. Rather than solving linear programs at a large number of descendant nodes, as is done in estimator (2.15), we instead evaluate a_t at these nodes and then sort the nodes based on their values. This guides sampling of the nodes to estimate CVaR. Having such a function a_t indicates that once we observe the random outcome for stage $t + 1$, we have some means of distinguishing "good" and "bad" decisions at stage t without knowledge of subsequent random events in stages $t + 2, \ldots, T$. Sometimes this is possible via an approximation of the recourse value associated with the system state. For example, when dealing with some asset allocation models, we may use current wealth to define a_t.

Algorithm 2.1 forms an empirical scenario tree with equally-weighted scenarios and discrete empirical distributions $\hat{P}_t$, $t = 2, \ldots, T$. The probability mass function (pmf) governing the conditional probability of the descendant nodes from any stage $t - 1$ node is given by:

$$g_t(\boldsymbol{\xi}_t) = \frac{1}{D_t}\mathbb{I}\left[\boldsymbol{\xi}_t \in \left\{\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{D_t}\right\}\right], \tag{3.1}$$

where the indicator function $\mathbb{I}[\cdot]$ takes value 1 if its argument is true and 0 otherwise.

We will need to compute value at risk levels, which will be defined as $(1 - \alpha)$ quantiles of the underlying distributions. Consider random loss Z and following definition of VaR at level α:

$$\mathrm{VaR}_\alpha[Z] = \inf\{u \in \mathbb{R} : \mathbb{P}[Z > u] \leq \alpha\} \tag{3.2}$$

We propose a sampling scheme based on importance sampling. The scheme depends on the current state of the system, giving rise to a new pmf, which we denote $h_t(\boldsymbol{\xi}_t|\boldsymbol{x}_{t-1})$. This pmf is tailored specifically for use with CVaR. Alternative pmfs would be needed to apply the proposed ideas to other risk measures. An example of possible extension would be spectral risk measures, based on a finite combination of CVaR risk measures. Given the current state of the system we can compute the value at risk for our approximation function, $u_a = \mathrm{VaR}_{\alpha_t}[a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)]$, and partition the nodes corresponding to $\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{D_t}$ into two groups by comparing their approximate value to u_a. We note that the value u_a is a function of the current decision, state of the system and confidence level, but we drop its arguments

for simplicity. In particular, the importance sampling pmf is:

$$h_t(\boldsymbol{\xi}_t|\boldsymbol{x}_{t-1}) = \begin{cases} \frac{1}{2}\frac{1}{\lfloor \alpha_t D_t \rfloor}\mathbb{I}\left[\boldsymbol{\xi}_t \in \left\{\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{D_t}\right\}\right], & \text{if } a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) \geq u_a \\ \frac{1}{2}\frac{1}{D_t - \lfloor \alpha_t D_t \rfloor}\mathbb{I}\left[\boldsymbol{\xi}_t \in \left\{\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{D_t}\right\}\right], & \text{if } a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) < u_a, \end{cases} \tag{3.3}$$

where the $\lfloor \cdot \rfloor$ operator rounds down to the nearest integer. The factor of $\frac{1}{2}$ in the pmf $h_t(\boldsymbol{\xi}_t|\boldsymbol{x}_{t-1})$ modifies the probability masses so that the probabilities of drawing sample observation above and below $u_a = \text{VaR}_{\alpha_t}\left[a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)\right]$ are equal. Here, we choose $\frac{1}{2}$ for simplicity, but in general a good choice of this factor could be tailored to the values of the confidence levels, α_t, and risk aversion coefficients, λ_t. We will further analyze this issue in Chapter 4.

In accordance with importance sampling schemes, we can compute the required expectation under our new measure via

$$\mathbb{E}_{g_t}\left[Z\right] = \mathbb{E}_{h_t}\left[Z\,\frac{g_t}{h_t}\right],$$

for any random variable Z for which the expectations exist. If the expectation is taken across the distributions for all T stages we denote the analogous operators by $\mathbb{E}_g\left[\cdot\right]$ and $\mathbb{E}_h\left[\cdot\right]$.

If we omit the rounding operations in equation (3.3), we have that the likelihood ratio satisfies:

$$\frac{g_t}{h_t} \approx \begin{cases} 2\alpha_t, & \text{if } a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) \geq u_a \\ 2(1-\alpha_t), & \text{if } a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) < u_a. \end{cases}$$

We can form an estimator similar to (2.13), except that we employ our importance sampling distributions, h_t, in place of the empirical distributions, g_t, in the forward pass of SDDP when selecting the sample paths. In particular, given a single sample path from stage 1 to stage T, $\boldsymbol{\xi}^j$, we form estimator (2.12), which uses recursion (2.11) and preserves the good scalability of the estimator with the number of stages. We carry this out for a set of samples drawn using the new measure h_t to select the sample paths.

Thus we have weights for each stage

$$w_t(\boldsymbol{\xi}_t|\boldsymbol{x}_{t-1}) = \frac{g_t(\boldsymbol{\xi}_t)}{h_t(\boldsymbol{\xi}_t|\boldsymbol{x}_{t-1})},$$

which yields weights along a sample path

$$w(\boldsymbol{\xi}^j) = \prod_{t=2}^{T} w_t(\boldsymbol{\xi}_t^{j_t}|\boldsymbol{x}_{t-1}),$$

and an estimator of the form

$$\frac{1}{M}\sum_{j=1}^{M} w(\boldsymbol{\xi}^j)\hat{\mathfrak{v}}(\boldsymbol{\xi}^j).$$

This estimator is a weighted sum of the upper bounds (2.12) for the sampled scenarios. Normalizing the weights reduces the variability of the estimator (see Hesterberg [44]) and yields:

$$U^{\mathbf{i}} = \frac{1}{\sum_{j=1}^{M} w(\boldsymbol{\xi}^j)} \sum_{j=1}^{M} w(\boldsymbol{\xi}^j)\hat{\mathfrak{v}}(\boldsymbol{\xi}^j), \tag{3.4}$$

where "**i**" indicates that the estimator uses importance sampling. We summarize the development so far in the following proposition.

Proposition 3.1. *Assume model (1.11) has finite optimal value, relatively complete recourse and interstage independence. Let φ denote the optimal value of model (1.11) under the empirical distributions, $\hat{P}_t$, $t = 2, \ldots, T$, generated by i.i.d. sampling. Let $\boldsymbol{\xi}$ denote a sample path selected under the empirical distribution, and let $\hat{\mathfrak{v}}(\boldsymbol{\xi})$ be defined by (2.12) for that sample path. Then $\mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right] \geq \varphi$. Furthermore if $\boldsymbol{\xi}^j$, $j = 1, \ldots, M$, are i.i.d. and generated by the pmfs (3.3) and $U^{\mathbf{i}}$ is defined by (3.4) then $U^{\mathbf{i}} \to \mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right]$, w.p.1, as $M \to \infty$.*

Proof. The optimal value of model (1.11) as reformulated in model (1.27) yields φ. Along sample path $\boldsymbol{\xi}$, under the assumption of relatively complete recourse, the cuts in subproblems (2.8) generate a feasible policy in the space of the $(\boldsymbol{x}_t, u_t)$ variables. Removing the expectation operator in equation (1.29), the associated recursion of the objective function in model (1.28) and equation (1.29) coincides with the recursion in equation (2.11). Taking expectations yields $\mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right] \geq \varphi$.

Since the expected value of importance sampling weights is equal to one, we have by the law of large numbers that

$$\lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} w(\boldsymbol{\xi}^j) = 1, \text{ w.p.1} \tag{3.5}$$

For $\boldsymbol{\xi}$ generated by the empirical pmfs (3.1) and for each $\boldsymbol{\xi}^j$, generated by the pmfs (3.3), we have

$$\mathbb{E}_h\left[w(\boldsymbol{\xi}^j)\hat{\mathfrak{v}}(\boldsymbol{\xi}^j)\right] = \mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right].$$

Thus by the law of large numbers we have

$$\lim_{M\to\infty} \frac{1}{M} \sum_{j=1}^{M} w(\boldsymbol{\xi}^j)\hat{\mathfrak{v}}(\boldsymbol{\xi}^j) = \mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right], \text{ w.p.1} \tag{3.6}$$

From the definition of $U^{\mathbf{i}}$ from (3.4) we have

$$U^{\mathbf{i}} = \frac{1}{M^{-1}\sum_{j=1}^{M} w(\boldsymbol{\xi}^j)} \frac{1}{M} \sum_{j=1}^{M} w(\boldsymbol{\xi}^j)\hat{\mathfrak{v}}(\boldsymbol{\xi}^j).$$

Using a converging-together result with equations (3.5) and (3.6) we then have

$$U^{\mathbf{i}} \to \mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right], \text{ w.p.1},$$

as $M \to \infty$. □

In the sense made precise in Proposition 3.1, estimator (3.4) provides an asymptotic upper bound on the optimal value of model (1.11). The naive estimator $U^{\mathbf{n}}$ of (2.13) is an unbiased and consistent estimator of $\mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right]$. However, if the functions a_t provide a good approximation, in the sense that they order the state of the system in the same way as the recourse function, we anticipate that $U^{\mathbf{i}}$ will have smaller variance than $U^{\mathbf{n}}$. That said, we view estimator (3.4) as an intermediate step to an improved estimator. Under an additional assumption, the estimator can be improved significantly. We now consider a stricter assumption, with the simplified notation $Q_t = Q_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$ and $a_t = a_t(\boldsymbol{x}_{t-1},\, \boldsymbol{\xi}_t)$.

Assumption 3.2. *For every stage $t = 2, \ldots, T$ and decision $\boldsymbol{x}_{t-1}$ the approximation function a_t satisfies:*

$$Q_t \geq \mathrm{VaR}_{\alpha_t}\left[Q_t\right] \;\; \textit{if and only if} \;\; a_t \geq \mathrm{VaR}_{\alpha_t}\left[a_t\right].$$

Assumption 3.2 requires that our approximation function can fully describe which observations belong to the tail of the recourse function, which is used for the evaluation of CVaR. Under this assumption we can strengthen the estimator through a reformulation. Given a sample path $\boldsymbol{\xi}$ we modify the recursive estimator (2.11) for $t = 2, \ldots, T$ as:

$$\hat{\mathfrak{v}}^{\mathbf{a}}_t(\boldsymbol{\xi}^{j_{t-1}}_{t-1}) \;\; = \;\; (1-\lambda_t)\left((\boldsymbol{c}^{j_t}_t)^\top \boldsymbol{x}^{j_t}_t + \hat{\mathfrak{v}}^{\mathbf{a}}_{t+1}(\boldsymbol{\xi}^{j_t}_t)\right) \tag{3.7a}$$

$$+\lambda_t u^{j_{t-1}}_{t-1} + \mathbb{I}[a_t \geq \mathrm{VaR}_{\alpha_t}\left[a_t\right]]\frac{\lambda_t}{\alpha_t}\left[(\boldsymbol{c}^{j_t}_t)^\top \boldsymbol{x}^{j_t}_t + \hat{\mathfrak{v}}^{\mathbf{a}}_{t+1}(\boldsymbol{\xi}^{j_t}_t) - u^{j_{t-1}}_{t-1}\right]_+, \tag{3.7b}$$

where $\hat{\mathfrak{v}}^{\mathbf{a}}_{T+1}(\boldsymbol{\xi}^{j_T}_T) \equiv 0$, and we let

$$\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi}) = \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \hat{\mathfrak{v}}^{\mathbf{a}}_2. \tag{3.8}$$

Like the estimators $U^{\mathbf{n}}$ and $U^{\mathbf{i}}$, which are based on (2.11) and (2.12), we note that the estimator we propose next, based on equations (3.7) and (3.8), requires explicit estimation of the VaR-level by the u_{t-1} decision variables. With $\boldsymbol{\xi}^j$, $j = 1, \ldots, M$, i.i.d. from the pmfs (3.3) we form the upper bound estimator:

$$U^{\mathbf{a}} = \frac{1}{\sum_{j=1}^M w(\boldsymbol{\xi}^j)} \sum_{j=1}^M w(\boldsymbol{\xi}^j)\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi}^j). \tag{3.9}$$

Proposition 3.3. *Assume the hypotheses of Proposition 3.1. Let $\boldsymbol{\xi}$ denote a sample path selected under the empirical distribution, let $\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})$ be defined by (3.8) for that sample path, and let Assumption 3.2 hold. Then $\mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})\right] \geq \varphi$. If $\boldsymbol{\xi}^j$, $j = 1, \ldots, M$, are i.i.d. and generated by the pmfs (3.3) and $U^{\mathbf{a}}$ is defined by (3.9) then $U^{\mathbf{a}} \to \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})\right]$, w.p.1, as $M \to \infty$. Furthermore if subproblems (2.8) induce the same policy for both $\hat{\mathfrak{v}}(\boldsymbol{\xi})$ and $\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})$ then $\mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right] \geq \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})\right]$.*

Proof. Let $(\boldsymbol{x}_1, u_1), \ldots, (\boldsymbol{x}_{T-1}, u_{T-1}), \boldsymbol{x}_T$ be the feasible sequence to models (1.27) and (1.28) for $t = 2, \ldots, T$, specified by (2.8) along sample path $\boldsymbol{\xi}$. The result $\mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right] \geq \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})\right]$ holds because $\mathbb{I}[a_t \geq \mathrm{VaR}_{\alpha_t}\left[a_t\right]]$ can preclude some positive terms in the recursion (3.7) that are included in (2.11).

The terms in (3.7b) are used to estimate CVaR. Thus to establish the proof of the proposition it suffices to show:

$$\begin{aligned}\mathrm{VaR}_{\alpha_t}[Q_t] + \frac{1}{\alpha_t}\mathbb{E}\left[[Q_t - \mathrm{VaR}_{\alpha_t}[Q_t]]_+\right] \leq \\ \leq u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[a_t \geq \mathrm{VaR}_{\alpha_t}[a_t]]\,[Q_t - u_{t-1}]_+\right]\end{aligned}$$

because the rest of the proof then follows in the same fashion as that of Proposition 3.1.

First consider the case in which $u_{t-1} \geq \mathrm{VaR}_{\alpha_t}[Q_t]$. We have:

$$\begin{aligned}&\mathrm{VaR}_{\alpha_t}[Q_t] + \frac{1}{\alpha_t}\mathbb{E}\left[[Q_t - \mathrm{VaR}_{\alpha_t}[Q_t]]_+\right] \\ &\quad\leq u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[[Q_t - u_{t-1}]_+\right] \\ &\quad= u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]\,[Q_t - u_{t-1}]_+\right] \\ &\quad= u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[a_t \geq \mathrm{VaR}_{\alpha_t}[a_t]]\,[Q_t - u_{t-1}]_+\right],\end{aligned}$$

where the inequality follows from CVaR's definition as the optimal value of a minimization problem, the first equality holds because the indicator has no effect when $u_{t-1} \geq \mathrm{VaR}_{\alpha_t}[Q_t]$, and the last equality follows from Assumption 3.2.

We will need following inequality to continue the proof: $\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]] \geq \alpha_t$. In the continuous case, we clearly have $\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]] = \alpha_t$ and the inequality holds. In the discrete case, $\mathrm{VaR}_{\alpha_t}[Q_t]$ has to be an atom of the distribution, otherwise there exists $\epsilon > 0$ such that $\mathbb{P}[Q_t > \mathrm{VaR}_{\alpha_t}[Q_t] - \epsilon] \leq \alpha_t$, which is in contradiction with the definition of VaR (3.2). Now if $\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]] < \alpha_t$, then again there exists $\epsilon > 0$ such that $\mathbb{P}[Q_t > \mathrm{VaR}_{\alpha_t}[Q_t] - \epsilon] \leq \alpha_t$, contradiction with the fact that $\mathrm{VaR}_{\alpha_t}[Q_t]$ fulfills the definition (3.2). Therefore $\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]] \geq \alpha_t$ holds in both cases.

For the case when $u_{t-1} < \mathrm{VaR}_{\alpha_t}[Q_t]$ we first drop the positive part operator, because it is handled by the indicator, and write:

$$\begin{aligned}&\mathrm{VaR}_{\alpha_t}[Q_t] + \frac{1}{\alpha_t}\mathbb{E}\left[[Q_t - \mathrm{VaR}_{\alpha_t}[Q_t]]_+\right] \\ &\quad= \mathrm{VaR}_{\alpha_t}[Q_t] + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]\,(Q_t - u_{t-1} + u_{t-1} - \mathrm{VaR}_{\alpha_t}[Q_t])\right] \\ &\quad= \left(1 - \frac{\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]}{\alpha_t}\right)\mathrm{VaR}_{\alpha_t}[Q_t] + \left(\frac{\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]}{\alpha_t}\right)u_{t-1} \\ &\qquad+ \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]\,(Q_t - u_{t-1})\right] \\ &\quad\leq \left(1 - \frac{\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]}{\alpha_t}\right)u_{t-1} + \left(\frac{\mathbb{P}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]}{\alpha_t}\right)u_{t-1} \\ &\qquad+ \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]\,(Q_t - u_{t-1})\right] \\ &\quad= u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t]]\,[Q_t - u_{t-1}]_+\right] \\ &\quad= u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[a_t \geq \mathrm{VaR}_{\alpha_t}[a_t]]\,[Q_t - u_{t-1}]_+\right],\end{aligned}$$

where the inequality holds because $\mathbb{P}\left[Q_t \geq \text{VaR}_{\alpha_t}\left[Q_t\right]\right] \geq \alpha_t$ means that the term before $\text{VaR}_{\alpha_t}\left[Q_t\right]$ is negative and we have that $u_{t-1} < \text{VaR}_{\alpha_t}\left[Q_t\right]$. This completes the proof as the desired result holds in both cases. □

As Proposition 3.3 indicates, $U^{\mathbf{a}}$ provides an asymptotic upper bound estimator for the optimal value of model (1.11). It also provides a tighter upper bound in expectation than estimators $U^{\mathbf{n}}$ and $U^{\mathbf{i}}$. We also anticipate that estimator $U^{\mathbf{a}}$ will have smaller variance than $U^{\mathbf{i}}$ provided the "induce the same policy" hypothesis holds. Note the same-policy hypothesis is not needed for the consistency result of this proposition. As we discuss in Section 2.2, when a sample path is such that the positive-part term in (2.11) does not equal to zero, it is multiplied by $\alpha_t^{-1} = 20$ (say). This increases the likelihood that as we recur backward in time, we obtain large values, repeatedly multiplied by α_{t-1}^{-1}, α_{t-2}^{-1}, etc. This repeated multiplication should occur for some samples, but it can also occur when it should not. The indicator function in $U^{\mathbf{a}}$ helps to avoid this issue and hence tends to reduce variance.

Under Assumption 3.2, the approximation function, a_t, characterizes the recourse value in that it fully classifies whether a realization is in the upper α tail of the recourse values. We now weaken Assumption 3.2 to incorporate the notion of what we call a *margin function*, $m_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$, in order to address a broader class of stochastic programs. Under Assumption 3.4, given below, the margin function is sufficient to classify a realization as *not* being in the upper α tail of the recourse values. It accomplishes this by effectively lowering the threshold that approximates the upper tail and has the effect of increasing the number of descendant scenarios that contribute to the positive-part CVaR term.

Assumption 3.4. *For every stage $t = 2, \ldots, T$ and decision $\boldsymbol{x}_{t-1}$ we have real-valued functions $a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$ and $m_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$ which satisfy:*

$$\textit{if } \; a_t < m_t \; \textit{ then } \; Q_t < \text{VaR}_{\alpha_t}\left[Q_t\right].$$

Given a sample path $\boldsymbol{\xi}$ we modify the recursive estimators (2.11) and (3.7) for $t = 2, \ldots, T$ as:

$$\begin{aligned}\hat{\mathfrak{v}}_t^{\mathbf{m}}(\boldsymbol{\xi}_{t-1}^{j_{t-1}}) = {} & (1-\lambda_t)\left((\boldsymbol{c}_t^{j_t})^\top \boldsymbol{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}^{\mathbf{m}}(\boldsymbol{\xi}_t^{j_t})\right) \\ & + \lambda_t u_{t-1}^{j_{t-1}} + \mathbb{I}[a_t \geq m_t]\frac{\lambda_t}{\alpha_t}\left[(\boldsymbol{c}_t^{j_t})^\top \boldsymbol{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}^{\mathbf{m}}(\boldsymbol{\xi}_t^{j_t}) - u_{t-1}^{j_{t-1}}\right]_+ ,\end{aligned} \tag{3.10}$$

where $\hat{\mathfrak{v}}_{T+1}^{\mathbf{m}}(\boldsymbol{\xi}_T^{j_T}) \equiv 0$, and we let

$$\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi}) = \boldsymbol{c}_1^\top \boldsymbol{x}_1 + \hat{\mathfrak{v}}_2^{\mathbf{m}}. \tag{3.11}$$

With $\boldsymbol{\xi}^j$, $j = 1, \ldots, M$, i.i.d. and from the pmfs (3.3), which use functions a_t, we form the upper bound estimator:

$$U^{\mathbf{m}} = \frac{1}{\sum_{j=1}^M w(\boldsymbol{\xi}^j)} \sum_{j=1}^M w(\boldsymbol{\xi}^j)\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi}^j). \tag{3.12}$$

Again, note that we do not modify the importance sampling procedure here to use the margin value. The sampling scheme still relies on the $\text{VaR}_{\alpha_t}\left[a_t\right]$ level

of the approximation function via the pmfs (3.3). However, the estimator based on (3.11) is more generally applicable than the estimator based on (3.8) because we can drop Assumption 3.2 and instead require only the weaker implication of Assumption 3.4.

Proposition 3.5. *Assume the hypotheses of Proposition 3.1. Let $\boldsymbol{\xi}$ denote a sample path selected under the empirical distribution, let $\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})$ be defined by (3.11) for that sample path, and let Assumption 3.4 hold. Then $\mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})\right] \geq \varphi$. If $\boldsymbol{\xi}^j$, $j = 1, \ldots, M$, are i.i.d. and generated by the pmfs (3.3) and $U^{\mathbf{m}}$ is defined by (3.12) then $U^{\mathbf{m}} \to \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})\right]$, w.p.1, as $M \to \infty$. Furthermore if subproblems (2.8) induce the same policy for both $\hat{\mathfrak{v}}(\boldsymbol{\xi})$ and $\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})$ then $\mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right] \geq \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})\right]$. Finally, if Assumption 3.2 also holds and subproblems (2.8) induce the same policy for both $\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})$ and $\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})$ then $\mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})\right] \geq \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})\right]$.*

Proof. We have:

$$\begin{aligned}
&\mathrm{VaR}_{\alpha_t}\left[Q_t\right] + \frac{1}{\alpha_t}\mathbb{E}\left[\left[Q_t - \mathrm{VaR}_{\alpha_t}\left[Q_t\right]\right]_+\right] \\
&\leq u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[Q_t \geq \mathrm{VaR}_{\alpha_t}\left[Q_t\right]]\left[Q_t - u_{t-1}\right]_+\right] \\
&\leq u_{t-1} + \frac{1}{\alpha_t}\mathbb{E}\left[\mathbb{I}[a_t \geq m_t]\left[Q_t - u_{t-1}\right]_+\right]
\end{aligned}$$

where the first inequality comes from following the steps of the proof of Proposition 3.3 (in both of the cases considered) and the second inequality follows from Assumption 3.4. From this we have $\mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})\right] \geq \varphi$, and the consistency result for $U^{\mathbf{m}}$ follows in the same manner as in the proof of Proposition 3.1. Inequality $\mathbb{E}_g\left[\hat{\mathfrak{v}}(\boldsymbol{\xi})\right] \geq \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})\right]$ holds because $\mathbb{I}[a_t \geq m_t]$ can preclude some positive terms in the recursion (3.10) that are included in (2.11). Finally, $\mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{m}}(\boldsymbol{\xi})\right] \geq \mathbb{E}_g\left[\hat{\mathfrak{v}}^{\mathbf{a}}(\boldsymbol{\xi})\right]$ holds because under Assumptions 3.2 and 3.4 the indicator $\mathbb{I}[a_t \geq m_t]$ allows inclusion of some positive terms that the indicator $\mathbb{I}[a_t \geq \mathrm{VaR}_{\alpha_t}\left[a_t\right]]$ does not. □

In order to ensure that $U^{\mathbf{a}}$ is a valid upper bound estimator we require that we have an approximation function that can fully order states of the system in the sense of Assumption 3.2, and this limits applicability of the estimator in some cases. Assumption 3.4 weakens considerably this requirement, and widens the applicability of estimator $U^{\mathbf{m}}$. While $U^{\mathbf{m}}$ again provides an asymptotic upper bound estimator for the optimal value of model (1.11), the price we pay is that it is weaker than $U^{\mathbf{a}}$ as Proposition 3.5 indicates.

For the types of approximation and margin functions (a_t and m_t) that we envision, our importance-sampling estimators ($U^{\mathbf{i}}$, $U^{\mathbf{a}}$, and $U^{\mathbf{m}}$) require modest additional computation relative to estimator $U^{\mathbf{n}}$, which uses samples from the empirical pmfs (3.1). In particular with D_t denoting the number of stage t descendant nodes formed in the sampling procedure, the bulk of the additional computation requires evaluating a_t and m_t at each of these D_t nodes and determining $u_a = \mathrm{VaR}_{\alpha_t}\left[a_t\right]$, which can be done by sorting with effort $O(D_t \log D_t)$ or in linear time in D_t (see [17]). This effort is small compared to solving linear programs for modest values of D_t, particularly recalling that in SDDP's backward pass we must solve linear programs at all D_t nodes to compute a cut. Furthermore, in Sections 2.1–3.1, we have simplified the presentation by using an SDDP tree with equally-likely realizations. However, our ideas generalize in a straightforward manner to handle general discrete distributions.

3.2 Performance of upper bound estimators

We present computational results for applying SDDP with the upper bound estimators described in Sections 2.2 and 3.1 to two variants of asset allocation model under nested CVaR risk measure. We present results for our four new upper bound estimators: (i) $U^{\mathbf{n}}$ from equation (2.13); (ii) $U^{\mathbf{i}}$ from equation (3.4); (iii) $U^{\mathbf{a}}$ from equation (3.9); and, (iv) $U^{\mathbf{m}}$ from equation (3.12). We compare their performance with that of the existing upper bound estimator from the literature: $U^{\mathbf{e}}$ from equation (2.15). The two asset allocation models (1.30) and (1.32) differ only in whether we include transaction costs or not. As we detail below, estimator $U^{\mathbf{a}}$ cannot be used in the case with transaction costs, but estimator $U^{\mathbf{m}}$ remains applicable even in this more complex model.

The assets in our allocation model (1.30) consist of the stock market indices DJA, NDX, NYA, and OEX. We used monthly data for these indices from September 1985 until September 2011 to fit the multivariate log-normal distribution to the price ratios observed month-to-month. The data summary can be found in the Table 1.1. The confidence level was set to $\alpha_t = 5\%$ and the risk coefficients were set to $\lambda_t = \frac{t-1}{T}$, $t = 2, \ldots, T$ so that risk aversion increases in later stages. Table 3.1 shows the sizes of the empirical scenario trees formed by Algorithm 2.1 for our test problem instances.

stages (T)	**descendants per node (D_t)**	**total scenarios ($\lvert\hat{\Omega}\rvert$)**
2	50,000	50,000
3	1,000	1,000,000
4	100	1,000,000
5	50	6,250,000
10	50	$\approx 10^{15}$
15	50	$\approx 10^{24}$

Table 3.1: Sizes of empirical scenario trees for test problem instances

The approximation function, $a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$, that we use for the importance sampling estimators $U^{\mathbf{i}}$ and $U^{\mathbf{a}}$ is simply our current wealth, which is determined by the previous stage decisions and the current price:

$$a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) = -\boldsymbol{r}_t^\top \boldsymbol{x}_{t-1}.$$

Note that this function meets the requirements of Assumption 3.2, because when we have no transaction costs, the specific allocations in the vector $\boldsymbol{x}_{t-1}$ can be rebalanced with no loss. This is captured mathematically in model (1.30) in that $a_t = -\boldsymbol{r}_t^\top \boldsymbol{x}_{t-1}$ acts as the sole state variable; i.e., the equality constraint of (1.30b) could be used to replace the first term in the objective function of (1.30a). Hence, we see Q_t also depends solely on a_t and that dependence is monotonic in a_t due to monotonicity of CVaR.

Our primary purpose is to compare the upper bound estimators that we have developed. For this reason we ran the SDDP algorithm with each of the upper bound estimators until the algorithm reached nearly the same optimal value as estimated by the first stage master program's objective function; i.e., the lower bound $\underline{z}$ from step 2 of Algorithm 2.2 for the risk-averse model. Specifically, SDDP was terminated when $\underline{z}$ did not improve by more than 10^{-6} over 10 iterations.

All lower bounds obtained across the four runs were equal. A total of 100 iterations of SDDP sufficed to accomplish this for problem instances with $T = 2, \ldots, 5$ and a total of 200 iterations sufficed for the larger instances with $T = 10$ and 15. For estimators $U^{\mathbf{n}}$, $U^{\mathbf{i}}$, and $U^{\mathbf{a}}$ on problem instances with $T = 2, 3, 4$, and 5 we used respective sample sizes of $M = 1001, 501, 334$, and 251. In this way, forming the estimators required solving around 1000 linear programming subproblems in each case. For $T = 10$ and 15 we used $M = 1112$ and 3572 so that forming the estimator required solving about 10,000 and 50,000 linear programs, respectively. For the estimator $U^{\mathbf{e}}$ we must specify a sample size M_t for each stage: For $T = 2$ we used $M_2 = 1000$. For $T = 3$ we used $M_2 = M_3 = 32$ because this means forming the estimator requires solving $32^2 \approx 1000$ linear programs and this allows for a fair comparison with the single-path estimators $U^{\mathbf{n}}$, $U^{\mathbf{i}}$, and $U^{\mathbf{a}}$. With similar reasoning for $T = 4$ we used $M_t = 11\,\forall t$, and for $T = 5$ we used $M_t = 6\,\forall t$, for the largest value of T for which we compute $U^{\mathbf{e}}$, $T = 10$, we used $M_t = 3\,\forall t$.

T	$\underline{z}$	$U^{\mathbf{n}}$ (s.d.)	$U^{\mathbf{i}}$ (s.d.)	$U^{\mathbf{a}}$ (s.d.)	$U^{\mathbf{e}}$ (s.d.)
2	-0.9518	-0.9515 (0.0020)	-0.9517 (0.0012)	-0.9517 (0.0011)	-0.9518 (0.0019)
3	-1.8674	-1.8300 (0.0145)	-1.8285 (0.0108)	-1.8656 (0.0060)	-1.8013 (0.0302)
4	-2.7811	-2.4041 (0.1472)	-2.3931 (0.1128)	-2.7764 (0.0126)	-2.6027 (0.0883)
5	-3.6794	-3.4608 (0.1031)	-3.4963 (0.1008)	-3.6731 (0.0303)	-2.9031 (0.5207)
10	-7.6394	9.3×10^4 (1.4×10^4)	9.0×10^4 (8.7×10^4)	-7.5465 (0.2562)	1.5×10^7 (1.3×10^6)
15	-11.5188	NA	NA	-11.0148 (0.6658)	NA

Table 3.2: Comparison of four upper bound estimators, including the point estimates and their standard deviations (s.d.) for the model with no transaction costs.

Table 3.2 shows results for four estimators for the asset allocation model without transaction costs. These results were computed using the sample sizes that we indicate above, except that we formed 100 i.i.d. replicates of the estimators. For a particular problem instance, all 100 replicates used the same single run of 100 or 200 iterations of SDDP. Each cell in Table 3.2 reports the mean and standard deviation of the 100 replicates of the estimator. The table also shows the lower bound $\underline{z}$ for the models obtained as we describe above. The estimators perform similarly for the two-stage problem instance, but the advantages of the proposed estimator, $U^{\mathbf{a}}$, are revealed as the number of stages grows. Note that estimators $U^{\mathbf{n}}$, $U^{\mathbf{i}}$ and $U^{\mathbf{e}}$ degrade at $T = 10$ for reasons we discuss above involving recursive multiplication by $\alpha_t^{-1} = 20$ along some sample paths. Due to this degradation we do not report results for these estimators for $T = 15$. We suspect it is for this same reason that the benefit of the importance sampling scheme is only fully realized when we include the indicator functions shown in equation (3.7); compare the performance of $U^{\mathbf{i}}$ and $U^{\mathbf{a}}$ in the table. For $T = 2, \ldots, 5$ the variance reduction of $U^{\mathbf{a}}$ relative to $U^{\mathbf{e}}$ grows from roughly 3 to 25 to 50 to 300. The smaller standard deviations of $U^{\mathbf{a}}$ could facilitate its use in a sensible stopping rule.

For our second set of problem instances the model incorporates transaction costs, see (1.32). This allows us to show how to implement our upper bound estimation procedure in a more complex model. The transaction costs are proportional to the value of the assets sold or bought, and the fee is $f_t = 0.3\%$ of the transaction value.

We again use the approximation function $a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) = -\boldsymbol{r}_t^\top \boldsymbol{x}_{t-1}$. With

nonzero transaction costs the conditions of Assumption 3.2 are no longer satisfied: Suppose in the third stage it is optimal to invest all money in stock A. Arriving at this point with the second stage portfolio consisting only of stock A is convenient because we need not rebalance and incur a transaction cost. A portfolio of less worth, in the sense of $\boldsymbol{r}_t^\top \boldsymbol{x}_{t-1}$, consisting only of stock A may be preferred to another portfolio with larger value of $\boldsymbol{r}_t^\top \boldsymbol{x}_{t-1}$ but consisting of other stocks. Fortunately, we can still compare some portfolios. Consider the worst case scenario in which we must rebalance the entire portfolio; i.e., sell all our assets and buy some other assets at stage t. This would increase the loss by a factor of $\frac{1+f_t}{1-f_t}$. However, this portfolio is still better than any other portfolio whose total loss exceeds this portfolio's loss adjusted by the same factor. This leads us to the margin function given by:

$$m_t = \left(\frac{1+f_t}{1-f_t}\right) \mathrm{VaR}_{\alpha_t}\left[a_t\right].$$

Functions a_t and m_t satisfy Assumption 3.4, and we can apply the upper bound estimator $U^{\mathbf{m}}$. This construction, of course, increases the bias of the estimator as we indicate in Proposition 3.5. However, if the transaction costs are modest compared to market volatility, we may expect our estimator to provide reasonable results.

Table 3.3 reports results in the same manner as Table 3.2, now comparing $U^{\mathbf{m}}$ and $U^{\mathbf{e}}$. The value $\underline{z}$ is computed in the same way we describe above. From Table 3.2 we see that the point estimate $U^{\mathbf{a}}$ as a percentage of $\underline{z}$ drops from 99.8% to 98.8% to 95.6% for $T = 5, 10$, and 15, respectively. The analogous values for $U^{\mathbf{m}}$ from Table 3.3 are weaker as expected, dropping from 99.6% to 98.7% to 90.5%. Note that these same values for $U^{\mathbf{e}}$ for $T = 5$ are 78.9% without transaction costs and 75.3% with transaction costs. For $T = 2, 3, 4, 5$ the variance reduction of $U^{\mathbf{m}}$ over $U^{\mathbf{e}}$ grows from roughly 3 to 20 to 40 to 400, again indicating that our proposed upper bound estimator is superior to the previously available estimator.

$\boldsymbol{T}$	$\underline{\boldsymbol{z}}$	$\boldsymbol{U^{\mathbf{m}}}$ **(s.d.)**	$\boldsymbol{U^{\mathbf{e}}}$ **(s.d.)**
2	-0.9518	-0.9517 (0.0011)	-0.9518 (0.0019)
3	-1.8668	-1.8642 (0.0060)	-1.8043 (0.0282)
4	-2.7697	-2.7555 (0.0138)	-2.5878 (0.0858)
5	-3.6653	-3.6508 (0.0306)	-2.7582 (0.6197)
10	-7.5579	-7.4562 (0.2339)	5.2×10^6 (7.8×10^5)
15	-11.3379	-10.2662 (0.8511)	NA

Table 3.3: Comparison of the estimators for the model with transaction costs

To assess the required computational effort, for the model instances without transaction costs for $T = 5$, 10, and 15 stages we ran 100 replications of estimators $U^{\mathbf{n}}$ and $U^{\mathbf{a}}$. For each estimator we used a sample size of $M = 1000$ and the subproblems in each stage had 1000 cuts. The computations were performed using a single thread on an Intel 2.53 GHz Core2 Duo with 4 GB of RAM and CPLEX version 12.2. The average computation time for estimator $U^{\mathbf{n}}$ grew from 8.7 seconds to 31.6 sec. to 67.4 sec. for the respective model instances with $T = 5$, 10, and 15 stages. The computation times, again averaged over 100 replications

of the estimators, for $U^{\mathbf{a}}$ grew from 6.8 sec. to 30.0 sec. to 66.5 sec. for the same instances. (The standard deviations associated with the run times are at most 1% of the average.) Like estimator $U^{\mathbf{n}}$, the computational effort we require to compute $U^{\mathbf{a}}$ scales well with the number of stages. We note that the estimator $U^{\mathbf{a}}$ does have additional computational overhead, relative to $U^{\mathbf{n}}$, that grew from an average 0.3 sec. to 1.2 sec. to 2.9 seconds for $T = 5$, 10, and 15 stages. However, for these particular problem instances the linear programming master programs happen to solve slightly more quickly for estimator $U^{\mathbf{a}}$ compared to $U^{\mathbf{n}}$. We obtained very similar scaling results with T for the estimator $U^{\mathbf{m}}$ applied to problem instances with transaction costs.

Although we stop short of a computational comparison, we discuss similarities and differences between our estimators ($U^{\mathbf{a}}$ and $U^{\mathbf{m}}$) and the inner approximation bound of Philpott et al. [69]. The computational work in [69] is on a hydro-thermal scheduling model that maintains an inventory of water (or energy) in four aggregate reservoirs. The analogous dimension for our first set of computational examples is one (current wealth), and the dimension is five for our second set of problem instances. Our largest models have 15 stages while those of Philpott et al. have 24 stages. The problems to which the respective bounds are applied are quite different, but our point estimate with the largest gap is about 10% (see $T = 15$ row of Table 3.3), and the gap reported in Table 3 of [69] with 10,000 cuts is of similar magnitude. With 10,000 cuts the inner approximation of [69] would require solving $150,000$ linear programs for our 15-stage models, while we report results for our estimators solving $50,000$ linear programs. We suspect that our approach will scale well with dimension, although we have yet to investigate this computationally. We note that our estimators require specification of functions that appropriately characterize the tail of the recourse function while the inner approximation of Philpott et al. does not have this requirement. A final distinction is that, like the SDDP algorithm for the risk-neutral model, our upper bound estimators are for the policy dictated by the cuts, i.e., the policy associated with the outer approximation. Philpott et al. do not provide an upper bound for this cut-based policy. Rather, their bound is for a policy associated with a set of points employed in developing the inner approximation.

4. Variance reduction with CVaR

4.1 Mean-risk estimators

We will consider random variable Z representing losses and the definition of conditional value at risk (1.12). CVaR penalizes losses in the upper α tail of Z with a typical value of α being 0.05. Under this setup, we consider a mean-risk functional, which forms a convex sum of expectation and risk associated with random loss Z. We will use a single-period version of (1.10) for our analysis:

$$\mathcal{Q}_\alpha[Z] = (1-\lambda)\,\mathbb{E}[Z] + \lambda\,\mathrm{CVaR}_\alpha[Z], \tag{4.1}$$

with $\lambda \in [0,1]$. In the following text, we will consider the confidence level α as fixed and omit it from our notation. When it is clear from the context, we write just $\mathcal{Q}$ or $\mathcal{Q}_Z$ instead of $\mathcal{Q}_\alpha[Z]$ for notational simplicity. CVaR is generally harder to estimate than the expectation operator, as we have illustrated in Section 1.5.

As we have pointed out in Example 1.9, standard Monte Carlo estimators may be inefficient when applied to functionals of the form (4.1). We have tackled this issue in Section 3.1 by using the importance sampling technique with a fixed choice of the importance sampling weights. Benefits of this technique are not restricted to the SDDP algorithm, our importance sampling scheme may be used in all procedures which employ sampling of CVaR value. Therefore, we will focus on the choice of importance sampling weights in a more general setup.

We suppose that a probability density function (pdf) of Z exists and denote it by g. Consider scenarios $Z^1, Z^2, \ldots, Z^M$ independent and identically distributed (i.i.d.) from the distribution given by g. Using equation (1.12), the standard Monte Carlo estimator of the functional $\mathcal{Q}$ takes the form:

$$\hat{\mathcal{Q}}^s = \min_u \left(\frac{1}{M} \sum_{j=1}^{M} (1-\lambda)\, Z^j + \lambda \left(u + \frac{1}{\alpha} \left[Z^j - u \right]_+ \right) \right). \tag{4.2}$$

Following sampling scheme is based on importance sampling with a new pdf, which we denote h. Given the original distribution of Z represented by the pdf g, we are able to compute the value at risk $u_Z = \mathrm{VaR}_\alpha[Z]$, which represents the $1-\alpha$ quantile, and partition the support of the distribution into two groups by comparing the value of the random variable with u_Z. Alternatively, the value u_Z could be estimated by sampling or computed from the given discrete distribution. In particular, the proposed importance sampling pdf is, with $\beta \in (0,1)$:

$$h(z) = \begin{cases} \dfrac{\beta}{\alpha} g(z), & \text{if } z \geq u_Z \\ \dfrac{1-\beta}{1-\alpha} g(z), & \text{if } z < u_Z, \end{cases} \tag{4.3}$$

compare with (3.3). The idea of pdf h is a modification of the probability masses so that we are more likely to draw sample observations above $u_Z = \mathrm{VaR}_\alpha[Z]$. We will try to find a suitable choice of the factor β, which should be based on the values of the confidence level α and risk aversion coefficient λ.

In accordance with importance sampling schemes, we can compute the required expectation under our new measure via

$$\mathbb{E}_g[Z] = \mathbb{E}_h\left[Z\,\frac{g}{h}\right], \tag{4.4}$$

for any random variable Z for which the expectations exist.

Consider i.i.d. scenarios $Z^1, Z^2, \ldots, Z^M$ from the distribution given by h. The importance sampling estimator of the functional $\mathcal{Q}$ takes the form:

$$\hat{\mathcal{Q}}^i = \min_u \frac{1}{M}\sum_{j=1}^{M} \frac{g(Z^j)}{h(Z^j)}\left((1-\lambda)\,Z^j + \lambda\left(u + \frac{1}{\alpha}\left[Z^j - u\right]_+\right)\right). \tag{4.5}$$

It follows from equation (4.3) that the likelihood ratio satisfies:

$$\frac{g}{h} = \begin{cases} \dfrac{\alpha}{\beta}, & \text{if } Z \geq u_Z \\ \dfrac{1-\alpha}{1-\beta}, & \text{if } Z < u_Z. \end{cases} \tag{4.6}$$

The terms $w^j = \frac{g(Z^j)}{h(Z^j)}$ can be viewed as weights in a weighted sum of estimators for the sampled scenarios. Normalizing the weights reduces the variability of the estimator (see Hesterberg [44]) and yields:

$$\hat{\mathcal{Q}^m} = \min_u \frac{1}{\sum_{j=1}^{M} w^j}\sum_{j=1}^{M} w^j\left((1-\lambda)\,Z^j + \lambda\left(u + \frac{1}{\alpha}\left[Z^j - u\right]_+\right)\right). \tag{4.7}$$

4.2 Variance reduction

Using the CVaR definition (1.12), the objective value (4.1) can be computed as:

$$\mathcal{Q} = \min_u \mathbb{E}_g\left[\left((1-\lambda)\,Z + \lambda\left(u + \frac{1}{\alpha}\left[Z - u\right]_+\right)\right)\right]. \tag{4.8}$$

It can be shown that the optimal value for the u in the problem above is u_Z, value at risk at the level α, see [74]. Hence:

$$\mathcal{Q} = \mathbb{E}_g\left[(1-\lambda)\,Z + \lambda\left(u_Z + \frac{1}{\alpha}\left[Z - u_Z\right]_+\right)\right]. \tag{4.9}$$

Now, instead of the standard pdf, we will use the importance sampling under the measure h. Using the equation (4.4) we have:

$$\mathcal{Q} = \mathbb{E}_h\left[\frac{g}{h}\left((1-\lambda)\,Z + \lambda\left(u_Z + \frac{1}{\alpha}\left[Z - u_Z\right]_+\right)\right)\right]. \tag{4.10}$$

For further development we define:

$$Q^s = (1-\lambda)\,Z + \lambda\left(u_Z + \frac{1}{\alpha}\left[Z - u_Z\right]_+\right)$$
$$Q^i = \frac{g}{h}\left((1-\lambda)\,Z + \lambda\left(u_Z + \frac{1}{\alpha}\left[Z - u_Z\right]_+\right)\right).$$

It clearly holds $\mathbb{E}_h[Q^i] = \mathbb{E}_g[Q^s]$, but our aim would be on decreasing the variance, e.g. finding suitable parameter β, so that $\mathrm{var}_h[Q^i] < \mathrm{var}_g[Q^s]$. In the following text, we will be working with several random variables. Therefore all symbols defined so far would be appended with the index of the random variable's name. We first show how variance of the operator Q^i behaves when we add a constant.

Proposition 4.1. *Let X, Y be random variables, $Y = X + \mu$, $\mu \in \mathbb{R}$, g_X and g_Y the corresponding pdfs. Suppose that their importance sampling versions h_X and h_Y are defined using the same value of parameter β. Then*

$$\mathrm{var}_{h_Y}\left[Q_Y^i\right] = \mathrm{var}_{h_X}\left[Q_X^i\right] + 2\mu\mathbb{E}\left[\left(\frac{g_X}{h_X} - 1\right)Q_X^s\right] + \mu^2\frac{(\alpha-\beta)^2}{\beta(1-\beta)}. \tag{4.11}$$

Proof. From the basic properties of expectation and CVaR we have $\mathbb{E}[Y] = \mathbb{E}[X] + \mu$ and $\mathrm{CVaR}_\alpha[Y] = \mathrm{CVaR}_\alpha[X] + \mu$. Therefore:

$$\begin{aligned}\mathbb{E}_{h_Y}\left[Q_Y^i\right] &= (1-\lambda)\,\mathbb{E}[Y] + \lambda\,\mathrm{CVaR}_\alpha[Y] = (1-\lambda)\,\mathbb{E}[X] + \lambda\,\mathrm{CVaR}_\alpha[X] + \mu \\ &= \mathbb{E}_{h_X}\left[Q_X^i\right] + \mu\end{aligned}$$

Clearly, $\mathrm{VaR}_\alpha[Y] = \mathrm{VaR}_\alpha[X] + \mu$ and $Y \geq u_Y \iff X \geq u_X$. Moreover, $Q_Y^s = Q_X^s + \mu$ and from equation (4.6) we have $\frac{g_Y}{h_Y} = \frac{g_X}{h_X}$. Altogether:

$$\begin{aligned}\mathbb{E}_{h_Y}\left[\left(Q_Y^i\right)^2\right] &= \mathbb{E}_{h_Y}\left[\frac{g_Y^2}{h_Y^2}\left(Q_Y^s\right)^2\right] = \mathbb{E}\left[\frac{g_Y}{h_Y}\left(Q_Y^s\right)^2\right] = \mathbb{E}\left[\frac{g_X}{h_X}\left(Q_Y^s\right)^2\right] \\ &= \mathbb{E}\left[\frac{g_X}{h_X}\left(\left(Q_X^s\right)^2 + 2\mu Q_X^s + \mu^2\right)\right] \\ &= \mathbb{E}_{h_X}\left[\left(Q_X^i\right)^2\right] + 2\mu\mathbb{E}_{h_X}\left[\frac{g_X}{h_X}Q_X^i\right] + \mu^2\mathbb{E}\left[\frac{g_X}{h_X}\right] \\ &= \mathbb{E}_{h_X}\left[\left(Q_X^i\right)^2\right] + 2\mu\mathbb{E}_{h_X}\left[\frac{g_X}{h_X}Q_X^i\right] + \mu^2\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\end{aligned}$$

$$\begin{aligned}\mathrm{var}_{h_Y}\left[Q_Y^i\right] &= \mathbb{E}_{h_Y}\left[\left(Q_Y^i\right)^2\right] - \left(\mathbb{E}_{h_Y}\left[Q_Y^i\right]\right)^2 \\ &= \mathbb{E}_{h_X}\left[\left(Q_X^i\right)^2\right] + 2\mu\mathbb{E}_{h_X}\left[\frac{g_X}{h_X}Q_X^i\right] + \mu^2\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right) - \\ &\quad - \left(\left(\mathbb{E}_{h_X}\left[Q_X^i\right]\right)^2 + 2\mu\mathbb{E}_{h_X}\left[Q_X^i\right] + \mu^2\right) \\ &= \mathrm{var}_{h_X}\left[Q_X^i\right] + 2\mu\mathbb{E}_{h_X}\left[\left(\frac{g_X}{h_X} - 1\right)Q_X^i\right] + \mu^2\frac{(\alpha-\beta)^2}{\beta(1-\beta)} \\ &= \mathrm{var}_{h_X}\left[Q_X^i\right] + 2\mu\mathbb{E}\left[\left(\frac{g_X}{h_X} - 1\right)Q_X^s\right] + \mu^2\frac{(\alpha-\beta)^2}{\beta(1-\beta)}\end{aligned}$$

□

Next, we will focus on scale transformations.

Proposition 4.2. *Let X, Y be random variables, $Y = \sigma X$, $\sigma > 0$, g_X and g_Y the corresponding pdfs. Suppose that their importance sampling versions h_X and h_Y are defined using the same value of parameter β. Then* $\mathrm{var}_{h_Y}[Q_Y^i] = \sigma^2\,\mathrm{var}_{h_X}[Q_X^i]$.

Proof. Similarly as in previous proof $\mathbb{E}[Y] = \sigma\mathbb{E}[X]$ and $\mathrm{CVaR}_\alpha[Y] = \sigma\,\mathrm{CVaR}_\alpha[X]$. Therefore $\mathbb{E}_{h_Y}[Q_Y^i] = \sigma\mathbb{E}_{h_X}[Q_X^i]$. Clearly, $\mathrm{VaR}_\alpha[Y] = \sigma\,\mathrm{VaR}_\alpha[X]$ and $Y \geq u_Y \iff X \geq u_X$. Moreover, from equation (4.6) we have $\frac{g_Y}{h_Y} = \frac{g_X}{h_X}$ and therefore $Q_Y^i = \sigma Q_X^i$.

$$\begin{aligned}\mathrm{var}_{h_Y}\left[Q_Y^i\right] &= \mathbb{E}_{h_Y}\left[Q_Y^i - \mathbb{E}_{h_Y}\left[Q_Y^i\right]\right]^2 = \mathbb{E}_{h_X}\left[\sigma Q_X^i - \sigma\mathbb{E}_{h_X}\left[Q_X^i\right]\right]^2 \\ &= \sigma^2\mathbb{E}_{h_X}\left[Q_X^i - \mathbb{E}_{h_X}\left[Q_X^i\right]\right]^2 = \sigma^2\,\mathrm{var}_{h_X}\left[Q_X^i\right].\end{aligned}$$

□

We will denote the probability density function of standardized normal distribution $\phi(x)$ and $\Phi(x)$ its distribution function. Recall that for $\mathcal{N}(0,1)$ we have $u_Z = \Phi^{-1}(1-\alpha)$.

Proposition 4.3. *Let $Y \sim \mathcal{N}(\mu, \sigma^2)$ be a random variable. Denote $X \sim \mathcal{N}(\frac{\mu}{\sigma}, 1)$ and $Z \sim \mathcal{N}(0,1)$. In order to minimize the variance $\mathrm{var}_{h_Y}[Q_Y^i]$, the optimal value of the importance sampling parameter β can be obtained by solving the quadratic equation:*

$$\frac{\partial}{\partial\beta}\left(\mathrm{var}_{h_X}\left[Q_X^i\right]\right) = 0, \tag{4.12}$$

with

$$\begin{aligned}&\frac{\partial}{\partial\beta}\left(\mathrm{var}_{h_X}\left[Q_X^i\right]\right) = \\ &= \frac{1-\alpha}{(1-\beta)^2}(1-\lambda)^2(1-\alpha-u_Z\phi(u_Z)) - \frac{\alpha}{\beta^2}(1-\lambda)^2(\alpha+u_Z\phi(u_Z)) \\ &\quad - \frac{\lambda^2}{\alpha\beta^2}\left(\alpha - u_Z\phi(u_Z) + u_Z^2\alpha\right) + \lambda^2 u_Z^2\left(\frac{(1-\alpha)^2}{(1-\beta)^2} - \frac{\alpha^2}{\beta^2}\right) \\ &\quad - 2\frac{\lambda(1-\lambda)\alpha}{\beta^2} + 2\lambda u_Z(1-\lambda)\phi(u_Z)\left(-\frac{\alpha}{\beta^2} - \frac{1-\alpha}{(1-\beta)^2}\right) \\ &\quad - 2\frac{\lambda^2}{\beta^2}u_Z(\phi(u_Z) - \alpha u_Z) + 2\frac{\mu}{\sigma}(1-\lambda)\phi(u_Z)\left(-\frac{\alpha}{\beta^2} - \frac{1-\alpha}{(1-\beta)^2}\right) \\ &\quad - 2\frac{\mu}{\sigma}\frac{\lambda}{\beta^2}(\phi(u_Z) - \alpha u_Z) + \frac{2\alpha^2\beta - 2\alpha\beta^2 - \alpha^2 + \beta^2}{\beta^2(1-\beta)^2}\left(2\frac{\mu}{\sigma}\lambda u_Z + \frac{\mu^2}{\sigma^2}\right).\end{aligned} \tag{4.13}$$

Proof. Following are known integrals with normal pdfs:

$$\begin{aligned}\int z\phi(z)\mathrm{d}z &= -\phi(z) + C, \\ \int z^2\phi(z)\mathrm{d}z &= \Phi(z) - z\phi(z) + C.\end{aligned}$$

We will write f instead od f_Z and g instead of g_Z for simplicity and expand the basic formula for variance:

$$\mathrm{var}_h\left[Q_Z^i\right] = \mathbb{E}_h\left[\left(Q_Z^i\right)^2\right] - \left(\mathbb{E}_h\left[Q_Z^i\right]\right)^2,$$

$$\begin{aligned}\mathbb{E}_h\left[\left(Q_Z^i\right)^2\right] =& \underbrace{\mathbb{E}_h\left[\frac{g^2}{h^2}\left(1-\lambda\right)^2 Z^2\right]}_{(1)} + \underbrace{\mathbb{E}_h\left[\frac{g^2}{h^2}\frac{\lambda^2}{\alpha^2}\left[Z-u_Z\right]_+^2\right]}_{(2)} + \underbrace{\mathbb{E}_h\left[\frac{g^2}{h^2}\lambda^2 u_Z^2\right]}_{(3)} + \\ &+ \underbrace{2\mathbb{E}_h\left[\frac{g^2}{h^2}\left(1-\lambda\right)\frac{\lambda}{\alpha}Z\left[Z-u_Z\right]_+\right]}_{(4)} + \underbrace{2\mathbb{E}_h\left[\frac{g^2}{h^2}\left(1-\lambda\right)\lambda u_Z Z\right]}_{(5)} + \\ &+ \underbrace{2\mathbb{E}_h\left[\frac{g^2}{h^2}\frac{\lambda}{\alpha}\lambda u_Z\left[Z-u_Z\right]_+\right]}_{(6)}.\end{aligned} \tag{4.14}$$

Next, we calculate all terms from the previous equations:

$$\begin{aligned}\mathbb{E}_h\left[\left(Q_Z^i\right)\right] &= \int_{u_Z}^{\infty}(z-u_Z)\frac{\alpha}{\beta}\frac{\lambda}{\alpha}\frac{\beta}{\alpha}\phi(z)\mathrm{d}z + \lambda u_Z \\ &= \frac{\lambda}{\alpha}\left(\int_{u_Z}^{\infty} z\phi(z)\mathrm{d}z - u_Z\int_{u_Z}^{\infty}\phi(z)\mathrm{d}z\right) + \lambda u_Z \\ &= \frac{\lambda}{\alpha}\left(\phi(u_Z) - u_Z(1-\Phi(u_Z)\right) + \lambda u_Z = \frac{\lambda}{\alpha}\phi(u_Z).\end{aligned}$$

$$\begin{aligned}(1)\ \mathbb{E}_h&\left[\frac{g^2}{h^2}\left(1-\lambda\right)^2 Z^2\right] = \\ &= \int_{-\infty}^{u_Z} z^2\frac{(1-\alpha)^2}{(1-\beta)^2}\left(1-\lambda\right)^2\frac{1-\beta}{1-\alpha}\phi(z)\mathrm{d}z + \int_{u_Z}^{\infty} z^2\frac{\alpha^2}{\beta^2}\left(1-\lambda\right)^2\frac{\beta}{\alpha}\phi(z)\mathrm{d}z \\ &= \frac{1-\alpha}{1-\beta}\left(1-\lambda\right)^2\left(\Phi(u_Z) - u_Z\phi(u_Z)\right) + \frac{\alpha}{\beta}\left(1-\lambda\right)^2\left(1-\Phi(u_Z) + u_Z\phi(u_Z)\right) \\ &= \frac{1-\alpha}{1-\beta}\left(1-\lambda\right)^2\left(1-\alpha-u_Z\phi(u_Z)\right) + \frac{\alpha}{\beta}\left(1-\lambda\right)^2\left(\alpha+u_Z\phi(u_Z)\right)\end{aligned}$$

$$\begin{aligned}(2)\ \mathbb{E}_h&\left[\frac{g^2}{h^2}\frac{\lambda^2}{\alpha^2}\left[Z-u_Z\right]_+^2\right] = \\ &= \int_{u_Z}^{\infty}(z-u_Z)^2\frac{\alpha^2}{\beta^2}\frac{\lambda^2}{\alpha^2}\phi(z)\frac{\beta}{\alpha}\mathrm{d}z \\ &= \frac{\alpha}{\beta}\frac{\lambda^2}{\alpha^2}\int_{u_Z}^{\infty}(z^2 - 2u_Z z + u_Z^2)\phi(z)\mathrm{d}z \\ &= \frac{1}{\beta}\frac{\lambda^2}{\alpha}\left(1-\Phi(u_Z) + u_Z\phi(u_Z) - 2u_Z\phi(u_Z) + u_Z^2 - u_Z^2\Phi(u_Z)\right) \\ &= \frac{\lambda^2}{\alpha\beta}\left(\alpha - u_Z\phi(u_Z) + u_Z^2\alpha\right)\end{aligned}$$

$$\begin{aligned}(3)\ \mathbb{E}_h&\left[\frac{g^2}{h^2}\lambda^2 u_Z^2\right] = \\ &= \int_{-\infty}^{u_Z}\frac{(1-\alpha)^2}{(1-\beta)^2}\lambda^2 u_Z^2\frac{1-\beta}{1-\alpha}\phi(z)\mathrm{d}z + \int_{u_Z}^{\infty}\frac{\alpha^2}{\beta^2}\lambda^2 u_Z^2\frac{\beta}{\alpha}\phi(z)\mathrm{d}z \\ &= \frac{1-\alpha}{1-\beta}(1-\alpha)\lambda^2 u_Z^2 + \frac{\alpha}{\beta}\alpha\lambda^2 u_Z^2 = \lambda^2 u_Z^2\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\end{aligned}$$

$$
\begin{aligned}
(4)\ & 2\mathbb{E}_h\left[\frac{g^2}{h^2}(1-\lambda)\frac{\lambda}{\alpha}Z\left[Z-u_Z\right]_+\right] = \\
&= 2\int_{u_Z}^{\infty} z(z-u_Z)\frac{\alpha^2}{\beta^2}(1-\lambda)\frac{\lambda}{\alpha}\frac{\beta}{\alpha}\phi(z)\mathrm{d}z \\
&= 2\frac{\alpha}{\beta}(1-\lambda)\frac{\lambda}{\alpha}\left(\int_{u_Z}^{\infty} z^2\phi(z)\mathrm{d}z - u_Z\int_{u_Z}^{\infty} z\phi(z)\mathrm{d}z\right) \\
&= 2\frac{\lambda(1-\lambda)}{\beta}\left(1-\Phi(u_Z)+u_Z\phi(u_Z)-u_Z\phi(u_Z)\right) \\
&= 2\frac{\lambda(1-\lambda)\alpha}{\beta}
\end{aligned}
$$

$$
\begin{aligned}
(5)\ & 2\mathbb{E}_h\left[\frac{g^2}{h^2}(1-\lambda)\lambda u_Z Z\right] = \\
&= 2\int_{-\infty}^{u_Z}\frac{(1-\alpha)^2}{(1-\beta)^2}\lambda u_Z(1-\lambda)\frac{1-\beta}{1-\alpha}z\phi(z)\mathrm{d}z + 2\int_{u_Z}^{\infty}\frac{\alpha^2}{\beta^2}\lambda u_Z(1-\lambda)\frac{\beta}{\alpha}z\phi(z)\mathrm{d}z \\
&= 2\lambda u_Z(1-\lambda)\phi(u_Z)\left(\frac{\alpha}{\beta}-\frac{1-\alpha}{1-\beta}\right)
\end{aligned}
$$

$$
\begin{aligned}
(6)\ & 2\mathbb{E}_h\left[\frac{g^2}{h^2}\frac{\lambda}{\alpha}\lambda u_Z\left[Z-u_Z\right]_+\right] = \\
&= 2\int_{u_Z}^{\infty}\frac{\alpha^2}{\beta^2}\frac{\lambda}{\alpha}\lambda u_Z\frac{\beta}{\alpha}(z-u_Z)\phi(z)\mathrm{d}z \\
&= 2\frac{\alpha}{\beta}\frac{\lambda}{\alpha}\lambda u_Z\left(\phi(u_Z)-\alpha u_Z\right) \\
&= 2\frac{\lambda^2}{\beta}u_Z\left(\phi(u_Z)-\alpha u_Z\right)
\end{aligned}
$$

Substituting all above terms into (4.14):

$$
\begin{aligned}
\mathrm{var}_h\left[Q_Z^i\right] =& \frac{1-\alpha}{1-\beta}(1-\lambda)^2(1-\alpha-u_Z\phi(u_Z)) + \frac{\alpha}{\beta}(1-\lambda)^2(\alpha+u_Z\phi(u_Z)) \\
&+ \frac{\lambda^2}{\alpha\beta}\left(\alpha-u_Z\phi(u_Z)+u_Z^2\alpha\right) + \lambda^2u_Z^2\left(\frac{(1-\alpha)^2}{1-\beta}+\frac{\alpha^2}{\beta}\right) \\
&+ 2\frac{\lambda(1-\lambda)\alpha}{\beta} + 2\lambda u_Z(1-\lambda)\phi(u_Z)\left(\frac{\alpha}{\beta}-\frac{1-\alpha}{1-\beta}\right) \\
&+ 2\frac{\lambda^2}{\beta}u_Z\left(\phi(u_Z)-\alpha u_Z\right) - \left(\frac{\lambda}{\alpha}\phi(u_Z)\right)^2.
\end{aligned}
\tag{4.15}
$$

Consider a random variable $X = Z + \frac{\mu}{\sigma}$. Then by using Proposition 4.1:

$$
\mathrm{var}_{h_X}\left[Q_X^i\right] = \mathrm{var}_{h_Z}\left[Q_Z^i\right] + 2\frac{\mu}{\sigma}\mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)Q_Z^s\right] + \frac{\mu^2}{\sigma^2}\frac{(\alpha-\beta)^2}{\beta(1-\beta)} \tag{4.16}
$$

Since $\mathbb{E}[Z] = 0$:

$$\begin{aligned}\mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)Q_Z^s\right] &= \mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)\left((1-\lambda)Z+\lambda\left(u_Z+\frac{1}{\alpha}[Z-u_Z]_+\right)\right)\right] \\ &= (1-\lambda)\underbrace{\mathbb{E}\left[\frac{g_Z}{h_Z}Z\right]}_{(a)}+\lambda u_Z\underbrace{\mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)\right]}_{(b)}+ \\ &\quad+\frac{\lambda}{\alpha}\underbrace{\mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)[Z-u_Z]_+\right]}_{(c)},\end{aligned} \tag{4.17}$$

where

$$(a)\ \mathbb{E}\left[\frac{g_Z}{h_Z}Z\right] = \int_{-\infty}^{u_Z}\frac{1-\alpha}{1-\beta}z\phi(z)\mathrm{d}z+\int_{u_Z}^{\infty}\frac{\alpha}{\beta}z\phi(z)\mathrm{d}z = \\ = \phi(u_Z)\left(\frac{\alpha}{\beta}-\frac{1-\alpha}{1-\beta}\right),$$

$$(b)\ \mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)\right] = \frac{(1-\alpha)^2}{1-\beta}+\frac{\alpha^2}{\beta}-1 = \frac{(\alpha-\beta)^2}{\beta(1-\beta)},$$

$$(c)\ \mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)[Z-u_Z]_+\right] = \int_{u_Z}^{\infty}\left(\frac{\alpha}{\beta}-1\right)(z-u_Z)\phi(z)\mathrm{d}z = \\ = \left(\frac{\alpha}{\beta}-1\right)(\phi(u_Z)-\alpha u_Z),$$

Substituting all above terms into (4.17):

$$\begin{aligned}\mathbb{E}\left[\left(\frac{g_Z}{h_Z}-1\right)Q_Z^s\right] &= (1-\lambda)\phi(u_Z)\left(\frac{\alpha}{\beta}-\frac{1-\alpha}{1-\beta}\right)+\lambda u_Z\frac{(\alpha-\beta)^2}{\beta(1-\beta)}+ \\ &\quad+\frac{\lambda}{\alpha}\left(\frac{\alpha}{\beta}-1\right)(\phi(u_Z)-\alpha u_Z)\end{aligned} \tag{4.18}$$

Combining (4.15), (4.16) and (4.18):

$$\begin{aligned}\mathrm{var}_{h_X}\left[Q_X^i\right] =& \frac{1-\alpha}{1-\beta}(1-\lambda)^2(1-\alpha-u_Z\phi(u_Z))+\frac{\alpha}{\beta}(1-\lambda)^2(\alpha+u_Z\phi(u_Z)) \\ &+\frac{\lambda^2}{\alpha\beta}\left(\alpha-u_Z\phi(u_Z)+u_Z^2\alpha\right)+\lambda^2u_Z^2\left(\frac{(1-\alpha)^2}{1-\beta}+\frac{\alpha^2}{\beta}\right) \\ &+2\frac{\lambda(1-\lambda)\alpha}{\beta}+2\lambda u_Z(1-\lambda)\phi(u_Z)\left(\frac{\alpha}{\beta}-\frac{1-\alpha}{1-\beta}\right) \\ &+2\frac{\lambda^2}{\beta}u_Z(\phi(u_Z)-\alpha u_Z)-\left(\frac{\lambda}{\alpha}\phi(u_Z)\right)^2 \\ &+2\frac{\mu}{\sigma}(1-\lambda)\phi(u_Z)\left(\frac{\alpha}{\beta}-\frac{1-\alpha}{1-\beta}\right)+2\frac{\mu}{\sigma}\frac{\lambda}{\alpha}\left(\frac{\alpha}{\beta}-1\right)(\phi(u_Z)-\alpha u_Z) \\ &+\frac{(\alpha-\beta)^2}{\beta(1-\beta)}\left(2\frac{\mu}{\sigma}\lambda u_Z+\frac{\mu^2}{\sigma^2}\right).\end{aligned} \tag{4.19}$$

We want to find a minimum with respect to the parameter β and therefore we differentiate:

$$\begin{aligned}
&\frac{\partial}{\partial\beta}\left(\mathrm{var}_{h_X}\left[Q_X^i\right]\right) = \\
&= \frac{1-\alpha}{(1-\beta)^2}(1-\lambda)^2(1-\alpha-u_Z\phi(u_Z)) - \frac{\alpha}{\beta^2}(1-\lambda)^2(\alpha+u_Z\phi(u_Z)) \\
&\quad - \frac{\lambda^2}{\alpha\beta^2}\left(\alpha - u_Z\phi(u_Z) + u_Z^2\alpha\right) + \lambda^2 u_Z^2\left(\frac{(1-\alpha)^2}{(1-\beta)^2} - \frac{\alpha^2}{\beta^2}\right) \\
&\quad - 2\frac{\lambda(1-\lambda)\alpha}{\beta^2} + 2\lambda u_Z(1-\lambda)\phi(u_Z)\left(-\frac{\alpha}{\beta^2} - \frac{1-\alpha}{(1-\beta)^2}\right) \\
&\quad - 2\frac{\lambda^2}{\beta^2}u_Z(\phi(u_Z) - \alpha u_Z) + 2\frac{\mu}{\sigma}(1-\lambda)\phi(u_Z)\left(-\frac{\alpha}{\beta^2} - \frac{1-\alpha}{(1-\beta)^2}\right) \\
&\quad - 2\frac{\mu}{\sigma}\frac{\lambda}{\beta^2}(\phi(u_Z) - \alpha u_Z) + \frac{2\alpha^2\beta - 2\alpha\beta^2 - \alpha^2 + \beta^2}{\beta^2(1-\beta)^2}\left(2\frac{\mu}{\sigma}\lambda u_Z + \frac{\mu^2}{\sigma^2}\right).
\end{aligned}$$

Consider now $Y = \sigma X$, $Y \sim \mathcal{N}(\mu, \sigma^2)$. Using Proposition 4.2 we have $\mathrm{var}_{h_Y}\left[Q_Y^i\right] = \sigma^2 \mathrm{var}_{h_X}\left[Q_X^i\right]$. Therefore by solving the quadratic equation

$$\frac{\partial}{\partial\beta}\left(\mathrm{var}_{h_X}\left[Q_X^i\right]\right) = 0,$$

we obtain the optimal selection of β with respect to the choice of parameters α and λ under the assumption of normal distribution.

□

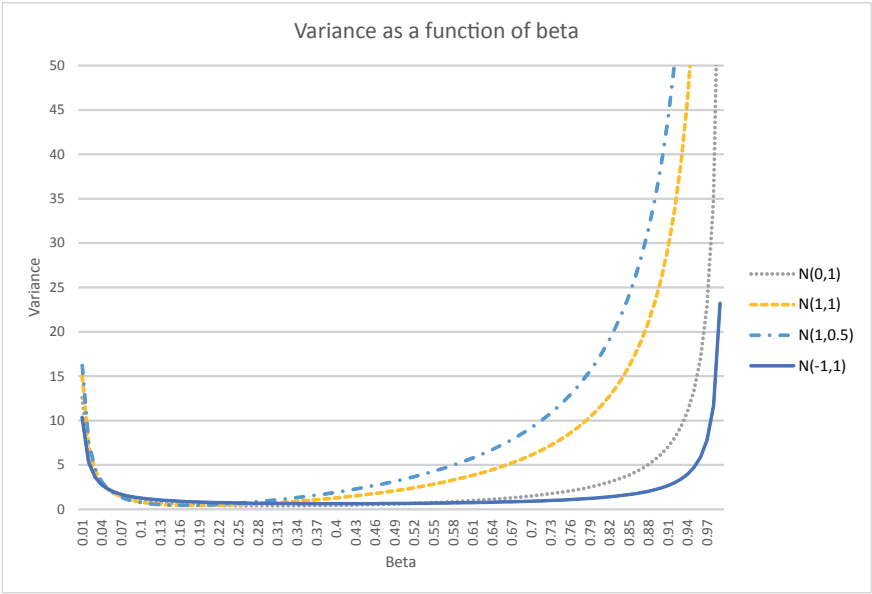


Figure 4.1: Variance as a function of β with $\alpha = 5\%$ and $\lambda = 0.5$

We can see that the importance sampling variance clearly depends on our choice of parameter β and the problem setup – confidence level α and the risk aversion coefficient λ. As expected, in the case of normal distribution the variance as a function of beta is a U-shaped function with minimums around the center

of the interval $(0, 1)$ and huge values near the boundaries of this interval. An example of this function is shown in Figure 4.1. When increasing the mean of the normal distribution, variance of our importance sampling estimator tends to grow faster with increasing β. The opposite holds for the variance parameter of the normal distribution, higher variance slows down the growth rate of variance with respect to β. The optimal choice of the parameter β for different choices of λ is depicted in Figure 4.2.

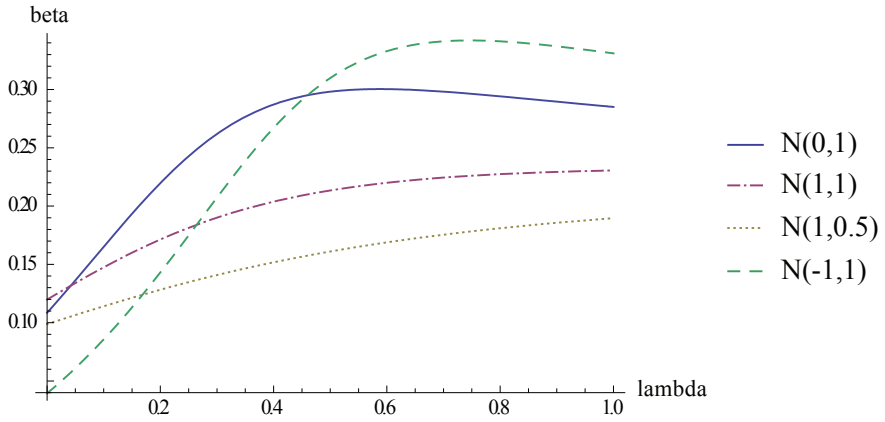


Figure 4.2: Optimal β as a function of λ with $\alpha = 5\%$

In the case of other distributions, one can proceed in the same way as we did with normal distribution. If the integrals are too hard to compute analytically, numerical solutions or sampling methods could be used. Following algorithm presents a simple approach to obtain a suitable value for parameter β for general distributions.

Algorithm 4.4 (Sampling algorithm to determine β)**.**
Let α and λ be specified from the model. Suppose we are able to draw i.i.d. samples from the distribution of Z, which could be continuous or discrete.

1. *Calculate the quantile $u_Z = \mathrm{VaR}_{1-\alpha}[Z]$. For discrete distributions, this can be done in linear time with respect to the number of scenarios, see [17]. For continuous distributions, use the analytical value or an estimate produced by sampling with the same procedure as for discrete distributions.*

2. *Set up a mesh of possible β values, e. g. $\mathcal{B} = \{0.01, 0.02, \ldots, 0.99\}$. Choose a number of iterations J, for instance $J = 10^6$.*

   For all $\beta \in \mathcal{B}$ {
   For $j = 1, \ldots, J$ {
   Sample uniformly a random number $r \in (0, 1)$;
   If $r < \beta$ {
   Sample Z^j from the distribution of Z until $Z^j \geq u_Z$;
   Set $Q^j = \lambda u_Z + \frac{\lambda}{\alpha}(Z^j - u_Z) + (1 - \lambda) Z^j$;
   Set $w^j = \frac{\alpha}{\beta}$;
   } else {
   Sample Z^j from the distribution of Z until $Z^j < u_Z$;
   Set $Q^j = \lambda u_Z + (1 - \lambda) Z^j$;

Set $w^j = \frac{1-\alpha}{1-\beta}$;

}

}

Calculate sampling mean: $\overline{\mathcal{Q}}_\beta = \frac{1}{\sum_{j=1}^J w^j} \sum_{j=1}^J w^j Q^j$;

Calculate sampling variance: $\mathrm{var}_\beta = \frac{1}{\sum_{j=1}^J w^j} \sum_{j=1}^J \left(w^j Q^j - \overline{\mathcal{Q}}_\beta\right)^2$;

}

3. *Choose the suitable* β^* *as* $\beta^* = \arg\min_{\beta \in \mathcal{B}} \mathrm{var}_\beta$.

We have chosen log-normal distribution to present the results of Algorithm 4.4. Since the distribution is not symmetrical, it is important to specify which tails are going to be connected with the greatest loss. We used three different setups of mean and variance and supposed that the heavier tails of the distribution are representing the greatest loss. Compared to the normal distribution, we can see a different shape of the variance function for certain parameters, see Figure 4.3.

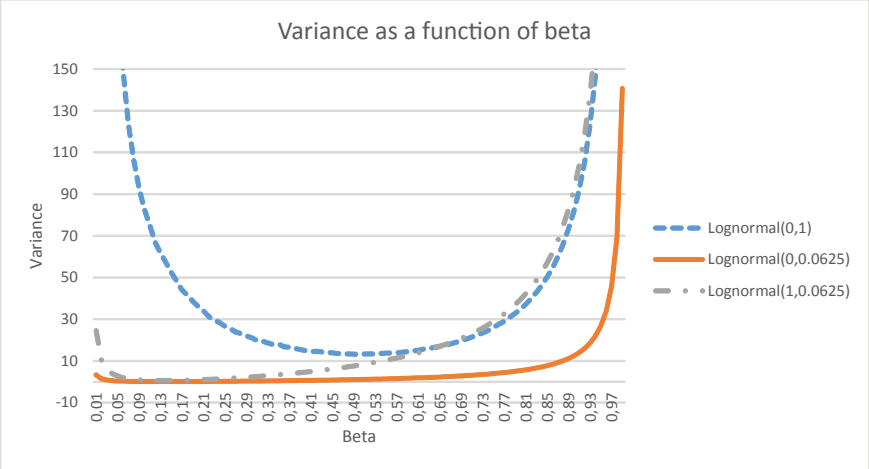


Figure 4.3: Variance as a function of β with $\alpha = 5\%$ and $\lambda = 0.5$ for log-normal distribution

4.3 Comparison with standard Monte Carlo

We have applied our importance sampling schemes to asset allocation model (1.34), with stock market indices DJA, NDX, NYA, and OEX as considered assets. We used monthly data for these indices from September 1985 until September 2011 to fit the multivariate log-normal distribution to the price ratios observed month-to-month. The data summary can be found in the Table 1.1. The confidence level was set to $\alpha_t = 5\%$ and the risk coefficients were set to $\lambda_t = \frac{1}{2}$, $t = 2, \ldots, T$.

The importance sampling pdf (3.3) is further enhanced to include the analysis performed in this section:

$$h_t(\boldsymbol{\xi}_t | \boldsymbol{x}_{t-1}) = \begin{cases} \dfrac{\beta_t}{\alpha_t} g_t, & \text{if } a_t \geq \mathrm{VaR}_{\alpha_t}\left[a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)\right] \\ \dfrac{1-\beta_t}{1-\alpha_t} g_t, & \text{if } a_t < \mathrm{VaR}_{\alpha_t}\left[a_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)\right]. \end{cases} \tag{4.20}$$

The optimal choice of the parameter β_t depends on the underlying distribution, as well as on parameters λ_t and α_t. The random inputs in our model have log-normal distribution, but even their sum, which is the total portfolio value, is not log-normal anymore. These values are then used in the multi-stage model, which leads to complicated transformations consisting of the conditional expectations mixed with CVaR. Moreover, the parameters of the distribution are different for each stage of the problem and different values of the parameter β should be chosen for each stage.

We have used Algorithm 4.4 to find the best values of β. We ran the SDDP algorithm with standard Monte Carlo setup until the algorithm reached nearly the same optimal value $\underline{z}$ as estimated by the first stage master program's objective function; i.e., the lower bound for the risk-averse model. Specifically, SDDP was terminated when lower bound did not improve by more than 10^{-6} over 10 iterations. Afterwards, we have performed a single run of the upper bound estimator and collected total of $100,000$ scenarios, equally distributed between the stages. The analysis was performed on large instances, represented by three possible numbers of stages, $T = 5, 10$ and 15, with 50 descendant scenarios per node in all of them. The results presented in Figure 4.4 show the impact of our nested transformations, the optimal value of beta is very low (0.03) for the last stage but grows significantly when progressing to the first stages (up to 0.58 for the fifteen-stage problem). The results of optimal β values show pretty convincing trend of monotonicity for 5 and 10 stages, however, they seem to be a bit unstable for the fifteen-stage problem. We believe that this is due to the sensitivity of the problem with respect to the extreme values of CVaR, which is fully exploited in the standard Monte Carlo scheme used for sampling the scenarios needed to run the Algorithm 4.4. One of the possible solutions could be to run the algorithm under importance sampling scheme and adjust its results. We would like to focus on this idea in our future research.

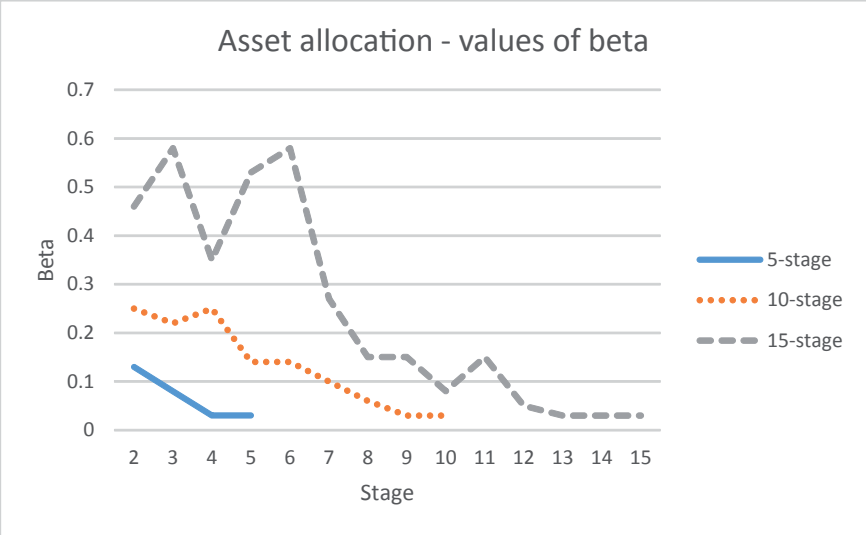


Figure 4.4: Choice of β for our asset allocation problem

We compare two estimators, both based on equation (3.9), but using the pdf (4.20) instead of (3.3) to sample i.i.d. scenarios $\boldsymbol{\xi}^j$, $j = 1, \ldots, M$. First setup $\hat{\mathcal{Q}}^s$ is based on standard Monte Carlo estimator, that means setting $\beta_t = \alpha_t = 0.05$ so

that our importance sampling pdf reduces to the standard one. The importance sampling setup $\hat{\mathcal{Q}}^i$ is based on our variance reduction analysis and values of β_t presented in Figure 4.4. We ran the SDDP algorithm again with each of the upper bound estimators until the algorithm reached nearly the same optimal value. We formed 100 i.i.d. replicates of the estimators, each of them required approximately $10,000$ linear programs to be solved. For a particular problem instance, all 100 replicates used the same single run of SDDP. We report the total number of scenarios, lower bound $\underline{z}$ and means and standard deviations of the upper bound estimators in the Table 4.1.

T	**total scenarios**	$\underline{z}$	$\hat{\mathcal{Q}}^s$ **(s.d.)**	$\hat{\mathcal{Q}}^i$ **(s.d.)**
5	$6,250,000$	-3.5212	-3.5166 (0.0168)	-3.5170 (0.0111)
10	$\approx 10^{14}$	-7.3885	-7.2833 (0.2120)	-7.2838 (0.0303)
15	$\approx 10^{25}$	-10.4060	-10.1482 (0.8184)	-10.1245 (0.1355)

Table 4.1: Comparison of the standard Monte Carlo approach and our importance sampling procedure, including the point estimates and their standard deviations (s.d.).

The estimators perform similarly in terms of bias from the lower bound $\underline{z}$, but our importance sampling procedure is significantly better in terms of variance. For $T = 5,\ 10,\ 15$ we achieved roughly 35%, 85% and 85% reduction of standard deviation, $\hat{\mathcal{Q}}^i$ relative to $\hat{\mathcal{Q}}^s$. Similar results can be obtained with a slightly suboptimal, but constant choice of β. When an easy approach is required, we may use for example $\beta = 0.3$, which seems to be in the low variance part of all of our normal distribution variance curves.

It may seem that we present our results only on a very simple model. However, the proposed upper bound estimation procedure can be employed also in more complex models, for example with transaction costs included in our asset allocation model. In that case, the importance sampling pdfs have to be adjusted to deal with the fact that our wealth is not enough to determine the system state. For more details, we refer to the Chapter 3. Based on the computational results presented therein, we can say that our estimator scales well with the number of stages and the additional computational effort is negligible in comparison with the effort required by the SDDP procedure. We would like to note that our sampling scheme is not restricted to the context of SDDP, but it can be applied in a similar fashion to any other Monte Carlo sampling scheme or algorithm, which employs sampling with CVaR.

5. Contamination technique

5.1 Contamination for multi-stage problems

The contamination technique for stochastic programs was developed in a series of papers as one of the tools for analysis of robustness of the optimal value with respect to deviations from the assumed probability distribution $\mathbb{P}$ and/or its parameters. The results were applied mainly to scenario-based two-stage stochastic linear programs, see e.g. [27, 29] for static and two-stage stochastic programs, [34] for CVaR and VaR risk criteria and [30] for polyhedral risk measures.

The first ideas dealing with contamination for multi-stage stochastic linear programs were presented in [28] and their application to study the influence of changes in the structure of multi-stage problems with polyhedral risk measures can be found in [35].

For construction of global contamination bounds, it is important that the stochastic program gets reformulated as

$$\min_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{x},\mathbb{P}) := \min_{\boldsymbol{x}\in\mathcal{X}} \int_{\Omega} f(\boldsymbol{x},\boldsymbol{\xi})\mathbb{P}(\mathrm{d}\boldsymbol{\xi}) \tag{5.1}$$

with $\mathcal{X}$ independent of $\mathbb{P}$, such as (1.13). Notice that the reformulations of the three models with multi-period CVaR-type risk measures presented in Section 1.4 comply with this requirement. If these basic conditions are not satisfied, local contamination bound can be constructed under additional assumptions, see [36, 37].

Possible changes in probability distribution $\mathbb{P}$ are modeled using contaminated distributions $\mathbb{P}_k$,

$$\mathbb{P}_k := (1-k)\mathbb{P} + k\mathbb{Q},\ k \in [0,1],$$

with $\mathbb{Q}$ another *fixed* probability distribution. Via contamination, robustness analysis with respect to changes in $\mathbb{P}$ gets reduced to a much simpler analysis with respect to a scalar parameter k.

Assume that (5.1) was solved for a probability distribution $\mathbb{P}$ and denote $\varphi(\mathbb{P})$ the optimal value and $\mathcal{X}^*(\mathbb{P})$ the set of optimal (first stage) solutions.

The objective function in (5.1) is linear in $\mathbb{P}$, hence

$$F(\boldsymbol{x},\mathbb{P}_k) = \int_{\Omega} f(\boldsymbol{x},\boldsymbol{\xi})\mathbb{P}_k(\mathrm{d}\boldsymbol{\xi}) = (1-k)F(\boldsymbol{x},\mathbb{P}) + kF(\boldsymbol{x},\mathbb{Q})$$

is linear in k. Suppose that the stochastic program (5.1) has an optimal solution $\varphi(\mathbb{P}_k)$ for all considered distributions $\mathbb{P}_k$, $0 \le k \le 1$. Since we restrict our analysis to the development with respect to a parameter k, we will sometimes write $\varphi(k)$ instead of $\varphi(\mathbb{P}_k)$ for notational convenience. The optimal value function

$$\varphi(k) = \min_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{x},\mathbb{P}_k)$$

is concave on $[0,1]$ which implies its continuity and existence of directional derivatives in $(0,1)$. Continuity at the point $k = 0$ is a property related to stability results for the stochastic program in question. In general, one needs a

nonempty, bounded set of optimal solutions $\mathcal{X}^*(\mathbb{P})$ of the initial stochastic program (5.1). This assumption together with stationarity of derivatives $\frac{\partial F(\boldsymbol{x},\mathbb{P}_k)}{\partial k} = F(\boldsymbol{x},\mathbb{Q}) - F(\boldsymbol{x},\mathbb{P})$ is used to derive the form of the directional derivative

$$\varphi'(0^+) = \min_{\boldsymbol{x}\in\mathcal{X}^*(\mathbb{P})} F(x,\mathbb{Q}) - \varphi(0) \tag{5.2}$$

which enters the upper bound for the concave optimal value function $\varphi(k)$

$$\varphi(0) + k\varphi'(0^+) \geq \varphi(k) \geq (1-k)\varphi(0) + k\varphi(1),\ k \in [0,1]; \tag{5.3}$$

see [27, 29, 34] and references therein. Contamination bounds (5.3) can be relaxed to

$$(1-k)\varphi(\mathbb{P}) + kF(\boldsymbol{x},\mathbb{Q}) \geq \varphi(\mathbb{P}_k) \geq (1-k)\varphi(\mathbb{P}) + k\varphi(\mathbb{Q}) \tag{5.4}$$

valid for an arbitrary $\boldsymbol{x} \in \mathcal{X}^*(\mathbb{P})$ and $k \in [0,1]$.

The development so far applies to general distributions $\mathbb{P}$ and $\mathbb{Q}$. To be able to solve the problem in practice, we usually form a Sample Average Approximation version of the problem and obtain a finite discrete distribution $\hat{\mathbb{P}}$. In order to have the contaminated problem solvable, the same property is required for the distribution $\hat{\mathbb{Q}}$. To apply the SDDP algorithm we have to assume that stage independence holds true for both the original distribution $\hat{\mathbb{P}}$ and the contaminating distribution $\hat{\mathbb{Q}}$. That way the contaminated distribution $\hat{\mathbb{P}}_k$ shares the same property.

When the problem is too large to be solved precisely, we obtain only approximate suboptimal solutions. In that case, an SDDP algorithm provides a lower bound $\underline{\varphi}$ and a statistical upper bound $\overline{\varphi}$ for the optimal objective function value. Since we always have $\underline{\varphi} < \varphi$, the contamination lower bound follows easily:

$$\varphi(\hat{\mathbb{P}}_k) \geq (1-k)\underline{\varphi}(\hat{\mathbb{P}}) + k\underline{\varphi}(\hat{\mathbb{Q}}). \tag{5.5}$$

With an approximate (suboptimal) solution $\tilde{\boldsymbol{x}}^*$ of the problem with original distribution $\hat{\mathbb{P}}$ we proceed in a following fashion. Since $\tilde{\boldsymbol{x}}^*$ is feasible, but in general suboptimal for the contaminated problem with $\hat{\mathbb{P}}_k$, we have:

$$\begin{aligned} \varphi(\hat{\mathbb{P}}_k) \leq F(\tilde{\boldsymbol{x}}^*, k) &= F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{P}}) + k\frac{\partial F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{P}}_k)}{\partial k} \\ &= F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{P}}) + k\left(F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{Q}}) - F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{P}})\right). \end{aligned} \tag{5.6}$$

Therefore, following is a valid upper bound for the contaminated problem:

$$\varphi(\hat{\mathbb{P}}_k) \leq (1-k)F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{P}}) + kF(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{Q}}). \tag{5.7}$$

Since the upper bound $\overline{\varphi}$ is provided by the approximate solution $\tilde{\boldsymbol{x}}^*$ we can replace the deterministic quantity $F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{P}})$ by its statistical estimator $\overline{\varphi}$. We will use the estimators $U^{\mathbf{a}}$ from equation (3.9) and $U^{\mathbf{m}}$ from equation (3.12), developed in Chapter 3. By Proposition 3.3 and Proposition 3.5 we have that $\overline{\varphi} \to \overline{\varphi}^*$, w.p.1, as the number of scenarios used to compute $\overline{\varphi}$ grows to infinity, and that $\overline{\varphi}^* \geq F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{P}})$.

Likewise, the value $F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{Q}})$ cannot be computed directly. Instead, we form an upper bound estimator under the distribution $\hat{\mathbb{Q}}$ in a fashion similar to the

SDDP algorithm (see Step 3 in Algorithm 2.2), but this time we use decisions $\tilde{\boldsymbol{x}}^*$ as given. In a practical large-scale application, the decisions $\tilde{\boldsymbol{x}}^*$ cannot be stored in a memory. We solve the original problem and store the cuts that are collected when the algorithm ends. Then, we proceed with the upper bound estimator under the distribution $\hat{\mathbb{Q}}$ and sample the scenarios. For these sampled scenarios, we use the stored cuts to obtain decisions $\tilde{\boldsymbol{x}}^*$ and apply these under the new distribution $\hat{\mathbb{Q}}$. With this approach, we are again getting a statistical upper bound $\bar{F}(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{Q}})$ for the value of $F(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{Q}})$. Plugging this value into our formula we reach the final upper contamination bound:

$$(1-k)\overline{\varphi}(\hat{\mathbb{P}}) + k\bar{F}(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{Q}}). \tag{5.8}$$

This bound is based on the set of scenarios used to compute the upper bounds and therefore it is not deterministic. We have an asymptotic validity of this bound provided again by the results of Proposition 3.3 and Proposition 3.5:

$$(1-k)\overline{\varphi}(\hat{\mathbb{P}}) + k\bar{F}(\tilde{\boldsymbol{x}}^*, \hat{\mathbb{Q}}) \to \overline{\varphi}^*_F, \text{ w.p.1.}; \quad \varphi(\hat{\mathbb{P}}_k) \leq \overline{\varphi}^*_F, \tag{5.9}$$

as the sizes of samples to compute both upper bounds grow to infinity.

The contamination bounds we have just developed depend on the specification or choice of $\mathbb{Q}$ and $\hat{\mathbb{Q}}$, respectively. When the contaminating distribution $\mathbb{Q}$ is not fully specified, a natural idea is to use the worst distribution of the considered alternatives. For simple uncertainty sets and under the assumption of stage independence such worst-case or robust contamination bounds can be applied. For instance when a probability set is specified by a known support $[l, u]$ and expectation μ, we can show that the worst-case measure is concentrated on the points l and u for risk-neutral problems. Moreover, this can be applied stage by stage to a multi-stage problem with one resulting measure of 2^{T-1} scenarios, where T denotes number of stages.

5.2 Shape of contamination bounds

We used monthly price data of the most important assets traded on the Prague Stock Exchange, January 2009 to February 2012 in the models (1.30) without transaction costs and (1.32) with transaction costs, based on the nested CVaR risk measure. The data summary can be found in the Table 1.2. We have fitted a multidimensional correlated log-normal distribution to the adjusted price ratios to obtain the original distribution $\mathbb{P}$. The contaminating distribution $\mathbb{Q}$ was then constructed from $\mathbb{P}$ by increasing the variance by 20%. Scenario trees were constructed independently for each stage by sampling $\hat{\mathbb{P}}$ and $\hat{\mathbb{Q}}$ from these distributions. The CVaR levels α_t were always set to 5%.

We evaluated the model with fixed value of risk coefficients $\lambda_t = 0.1$ for all stages. Both cases, with transaction costs of 0.3% and without transaction costs, were considered. We have computed the contamination bounds for problems with horizon of $T = 3$ and $T = 5$ stages. In the Table 5.1, we show the setup for the scenario trees used in our algorithm.

The three-stage problems can be solved to optimality using our SDDP algorithm, meaning that there is no gap between the lower bound and the upper bound, which is formed by computing the population mean rather than sampling.

stages	descendants per node	total scenarios
3	$1,000$	$1,000,000$
5	$1,000$	10^{12}

Table 5.1: Testing problems setup for contamination bounds

Figures 5.1 and 5.2 present the obtained lower and upper contamination bounds based on inequalities (5.5) and (5.7).

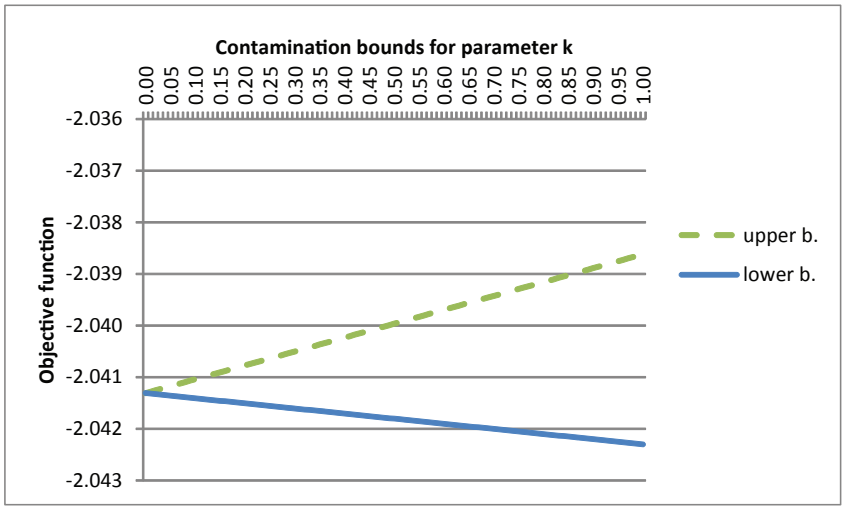


Figure 5.1: Three-stage problem contamination bounds with no transaction costs

For the case of five-stage problems we are unable to compute the solutions exactly and we provide the contamination bounds based on the lower and upper bounds from the SDDP algorithm. The lower bound based on the inequality (5.5) remains deterministic, but the terms present in inequality (5.8) are estimated for 10 times and we provide their mean as well as the empirical statistical upper bounds with confidence level of 95%. The results are presented in Figures 5.3 and 5.4.

The presented results show that for smaller problems we are able to obtain tight contamination bounds, in our testing setup with three stages we have the spread of 0.09% in the case without transaction costs and 0.17% spread in the case with transaction costs, both cases considering $k = 50\%$ contamination. For large-scale problems, we can rely on the statistically valid bound or on the mean of sampled estimates. For our five-stage setup, we obtained the spread of 1.13% and 1.03% in the analogous cases when using empirical statistical upper bounds. Even though we consider these bounds pretty tight, we can also rely on the mean estimators, which are usually used in the SDDP algorithms. That gives us spread of 0.38% and 0.19%, respectively. The straightforward interpretation of our results would then state that the results of our model can be considered stable with respect to growing variance of the underlying random distribution which drives the asset price evolution.

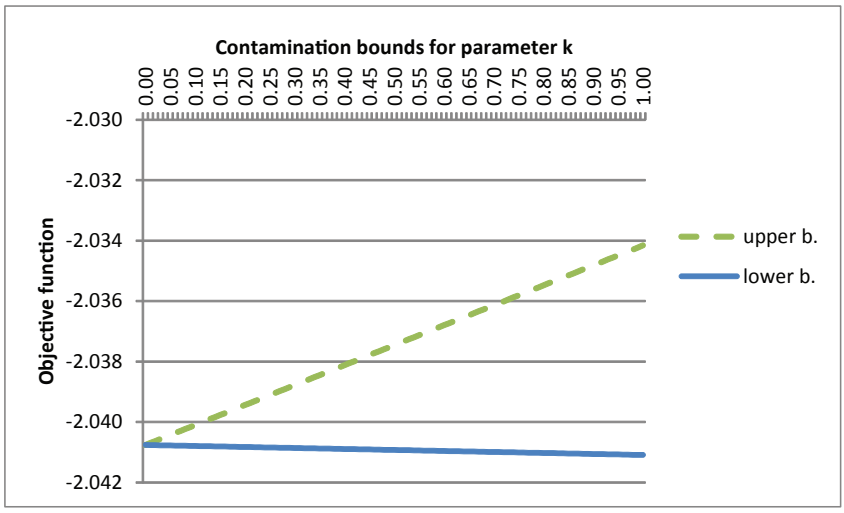


Figure 5.2: Three-stage problem contamination bounds with transaction costs 0.3%

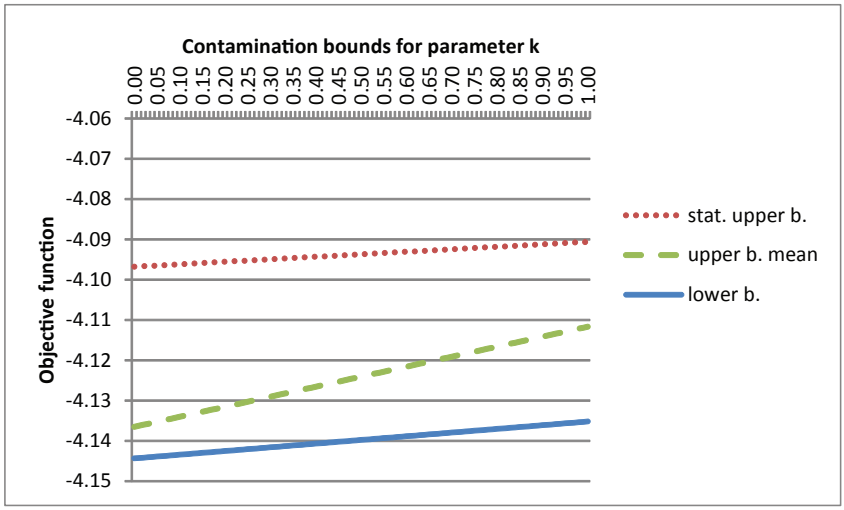


Figure 5.3: Five-stage problem contamination bounds with no transaction costs

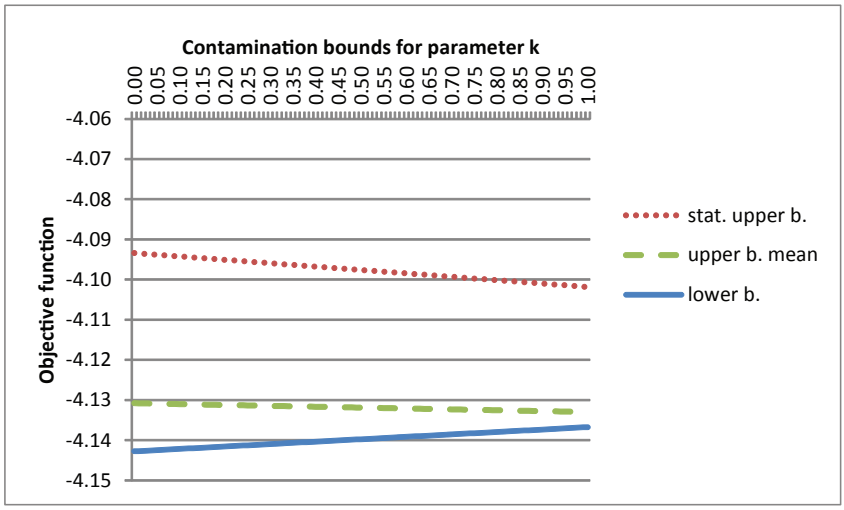


Figure 5.4: Five-stage problem contamination bounds with transaction costs 0.3%

6. Numerical illustrations

The purpose of this section is to provide results of the presented asset allocation models (1.30), (1.32) and (1.34). The models are applied under different setups, with varying number of stages, risk attitude or underlying data. The resulting solutions are then compared and the differences discussed.

6.1 Risk aversion study

We used weekly data of the most important assets traded on the Prague Stock Exchange, November 2007 to March 2012. The week-to-week ratios were adjusted to include the stock dividends, their summary can be found in the Table 1.3. Every computation including the sampling process was repeated 10 times, allowing to compute standard deviations of the solutions and objective values. The confidence level was always set to 5%.

We evaluated the model with two different settings of risk coefficients, $\lambda_t = \frac{1}{2}$ and $\lambda_t = \frac{t-1}{T}$. The first represents a stable risk-aversion, while the latter means that we want to be sure about the final portfolio value by being more risk-averse as the final stage approaches. Both cases, with transaction costs of 0.3% and without transaction costs, were considered. We have computed the optimal first-stage decisions for models with 2, 3 and 5 stages, the setup of scenario trees used in our algorithm can be found in the Table 6.1.

stages	**descendants per node**	**total scenarios**
2	$50,000$	$50,000$
3	$1,000$	$1,000,000$
5	$1,000$	10^{12}

Table 6.1: Testing problems setup for risk analysis

In all the testing cases, only three assets play a significant role in our portfolio: ČEZ, PHILIP MORRIS ČR and TELEFÓNICA C.R.. We believe that this is due to the good ratio of profit and variance for PHILIP MORRIS ČR and due to the low variance of the other two assets. We will exclude all remaining assets from our results to ease the orientation. We present the results without transaction costs first, showing the optimal first-stage decisions for constant and growing risk-aversion settings in Tables 6.2 and 6.3. The differences between the optimal portfolios for 2, 3 or 5 stages are not substantial in the constant risk coefficients setting. For variable risk aversion, we can see a slight movement to the riskier asset in the first-stage decision, putting more weight to PHILIP MORRIS ČR. This is expected as the risk settings target mostly the expectation part of our recourse function in the first stage.

Next we show the optimal first-stage decisions with transaction costs of 0.3% in the Tables 6.4 and 6.5. It should be noted that, in accordance with our model, transaction costs have no effect in two-stage models. We observe that the presence of the transaction costs reduces the differences found in the previous case with varying risk coefficients. We believe that this follows from the fact that varying risk coefficients require the investor to change the portfolio in every

stages	ČEZ	PHILIP M.	TELEFÓNICA
2	0.0663 (0.0087)	0.3169 (0.0081)	0.6168 (0.0092)
3	0.0510 (0.0459)	0.3112 (0.0537)	0.6273 (0.0707)
5	0.0450 (0.0307)	0.3340 (0.0268)	0.6043 (0.0571)

Table 6.2: Optimal decisions (std. deviations) with $f_t = 0$ and $\lambda_t = \frac{1}{2}$

stages	ČEZ	PHILIP M.	TELEFÓNICA
2	0.0663 (0.0087)	0.3169 (0.0081)	0.6168 (0.0092)
3	0.0597 (0.0645)	0.3429 (0.0650)	0.5792 (0.0920)
5	0.0392 (0.0415)	0.4325 (0.0678)	0.4975 (0.0652)

Table 6.3: Optimal decisions (std. deviations) with $f_t = 0$ and $\lambda_t = \frac{t-1}{T}$

stage significantly. However, with the transaction costs in mind, this could be more expensive than the loss coming from holding slightly suboptimal, but stable portfolio. The impact of the transaction costs should be weaker in cases where stages cover longer time periods instead of just weeks.

stages	ČEZ	PHILIP M.	TELEFÓNICA
2	0.0663 (0.0087)	0.3169 (0.0081)	0.6168 (0.0092)
3	0.0405 (0.0279)	0.2977 (0.0322)	0.6438 (0.0409)
5	0.0643 (0.0208)	0.3115 (0.0231)	0.6149 (0.0323)

Table 6.4: Optimal decisions (std. deviations) with $f_t = 0.3\%$ and $\lambda_t = \frac{1}{2}$

stages	ČEZ	PHILIP M.	TELEFÓNICA
2	0.0663 (0.0087)	0.3169 (0.0081)	0.6168 (0.0092)
3	0.0412 (0.0389)	0.3175 (0.0258)	0.6192 (0.0403)
5	0.0493 (0.0240)	0.3274 (0.0346)	0.6168 (0.0293)

Table 6.5: Optimal decisions (std. deviations) with $f_t = 0.3\%$ and $\lambda_t = \frac{t-1}{T}$

The impact of adding more stages to the stage-wise independent model with constant risk-aversion settings tends to be minimal. This could, however, be different in the fully general case. On the other hand, varying risk coefficients provide distinct solutions even with the independence assumption. In order to handle more general models, we would need to employ complex estimation procedures and some of the enhancements of the SDDP algorithm, such as autoregressive process [49] or SDDP coupled with Markov chain [67].

6.2 Models comparison

The purpose of this section is to compare two different models, both based on CVaR risk measure, and analyze their outputs under the changes of the probability distribution by contaminating the original one.

We used monthly price data of the most important assets traded on the Prague Stock Exchange, January 2009 to February 2012 in the models (1.32) and (1.34).

The data summary can be found in the Table 1.2. We have evaluated both models, nested and multi-period, under two different distributions. First distribution, $\mathbb{P}$, is based on the input data. Second distribution, $\mathbb{Q}$, is constructed from $\mathbb{P}$ by increasing the variance by 20% to test the stability of our solutions. All other technical details of our computations remain the same, we have repeated the process 10 times and provide the mean of the obtained solutions as well as their standard deviations. The CVaR levels α_t were always set to 5%.

We evaluated the models with three different settings of the risk coefficients $\lambda_t = 0.1$, $\lambda_t = 0.2$ and $\lambda_t = \frac{t-1}{T}$, and under assumption of no transaction costs, $f_t = 0\%$. We have considered three-stage model with $1,000$ descendants per node, with a total of $1,000,000$ scenarios.

The first stage decisions found for all considered setups of three-stage models are listed in the Table 6.6. Most of the optimal portfolios contain only the best performing assets - AAA, PEGAS and PHILIP MORRIS, some of them also contain low variance assets of TELEFÓNICA and UNIPETROL. By comparing the results, we think that both models are relatively stable with respect to variance of the underlying distribution. Direct comparison of standard deviations shows that the nested model provides more stable solutions and, moreover, the diversification of the portfolio seems to be better in the nested model. When increasing the risk aversion coefficient λ_t, solutions become more stable in both nested and multiperiod models. Therefore, a conclusion based on these illustrative results would state that the nested model provides better diversification of assets in our portfolios and better stability of the solutions with respect to the sample from the true underlying distribution.

$\boldsymbol{\lambda_t}$	**model**	**distr.**	**AAA**	**PEGAS**	**PHILIP M.**	**TELEFÓNICA**	**UNIPETROL**
0.1	nested	$\hat{\mathbb{P}}$	0.2388 (0.1133)	0.3893 (0.1109)	0.3720 (0.1011)	0.0000 (0.0000)	0.0000 (0.0000)
0.1	nested	$\hat{\mathbb{Q}}$	0.2718 (0.1600)	0.3582 (0.0902)	0.3700 (0.1565)	0.0000 (0.0000)	0.0000 (0.0000)
0.1	multiper.	$\hat{\mathbb{P}}$	0.6034 (0.3681)	0.2262 (0.2084)	0.1704 (0.2000)	0.0000 (0.0000)	0.0000 (0.0000)
0.1	multiper.	$\hat{\mathbb{Q}}$	0.6032 (0.3453)	0.1660 (0.1562)	0.2308 (0.2369)	0.0000 (0.0000)	0.0000 (0.0000)
0.2	nested	$\hat{\mathbb{P}}$	0.1774 (0.0681)	0.4132 (0.0774)	0.4032 (0.0907)	0.0000 (0.0000)	0.0063 (0.0148)
0.2	nested	$\hat{\mathbb{Q}}$	0.1730 (0.0541)	0.3471 (0.0566)	0.4545 (0.0429)	0.0000 (0.0000)	0.0254 (0.0462)
0.2	multiper.	$\hat{\mathbb{P}}$	0.3081 (0.1472)	0.2993 (0.1757)	0.3926 (0.0990)	0.0000 (0.0000)	0.0000 (0.0000)
0.2	multiper.	$\hat{\mathbb{Q}}$	0.3127 (0.1776)	0.3963 (0.0975)	0.2910 (0.1781)	0.0000 (0.0000)	0.0000 (0.0000)
$(t-1)/T$	nested	$\hat{\mathbb{P}}$	0.0817 (0.0342)	0.3206 (0.0587)	0.4458 (0.0363)	0.0402 (0.0501)	0.1118 (0.0671)
$(t-1)/T$	nested	$\hat{\mathbb{Q}}$	0.0484 (0.0341)	0.2607 (0.0427)	0.4316 (0.0387)	0.0783 (0.0526)	0.1810 (0.0407)
$(t-1)/T$	multiper.	$\hat{\mathbb{P}}$	0.1911 (0.0446)	0.3949 (0.0837)	0.4135 (0.0623)	0.0000 (0.0000)	0.0005 (0.0015)
$(t-1)/T$	multiper.	$\hat{\mathbb{Q}}$	0.1530 (0.0500)	0.3741 (0.0642)	0.4658 (0.0574)	0.0000 (0.0000)	0.0071 (0.0169)

Table 6.6: First stage decisions for three-stage models

Conclusion

We have presented three different multi-stage stochastic models based on multi-period CVaR risk measures. Their basic properties and differences have been discussed, with special attention to the time consistency property. Under the assumption of stage-wise independence stochastic dual dynamic programming algorithm is applied to models with a nested CVaR risk measure and multi-period CVaR risk measure, and illustrative comparison of the first stage solutions is given.

We have explained why the standard estimators for a policy value perform poorly in the nested CVaR model and presented a new approach to compute statistically valid upper bound, which is based on importance sampling. Under relatively mild conditions our most widely applicable estimator provides much better results than an existing estimator from the literature, in terms of reduced bias, smaller variance, and viability in problems with more than a few stages. We believe this type of estimator could be used to form better stopping rules for SDDP-style algorithms and possibly other algorithms, too. Such stopping rules allow for quantification of the quality of a solution coming from the algorithm, which we see as important in practice.

We have further elaborated importance sampling scheme which can be used to approximate functionals that incorporate risk via CVaR. Our sampling scheme does not apply only to SDDP, but it can be used to tweak existing procedures which rely on sampling in estimation of mean-CVaR objectives. With relatively mild effort our approach provides significantly lower variance than a standard Monte Carlo estimator.

In addition, our results are proven useful in the extension of contamination technique to cover the large-scale cases where we are not able to solve the problem precisely, but we can obtain approximate solutions through SDDP. Since the extension is based on lower and upper bounds from the SDDP algorithm, it is essential to have tight upper bound estimators in order to get reasonable contamination bounds. Numerical results with the asset allocation problem provide sufficiently tight bounds that can be used in practical applications to test stability.

We have considered a simple stock asset allocation model for the multi-stage setting and successfully employed the SDDP algorithm to solve it. We provide results based on assets from US stock market and Prague Stock Exchange for various setups. Our results show that the nested model with constant risk-aversion coefficients provides similar results in both two-stage and multi-stage settings. On the other hand, it provides distinct behavior in the case when the risk coefficients vary throughout the stages in order to represent more complicated investment strategies. We have also evaluated the effect of transaction costs in our model and pointed out that their presence could draw the dynamic model behavior closer to the static one. Comparison of the first stage decisions of nested and multi-period CVaR models indicates that the nested model provides more stable decisions with respect to the sample of the true underlying distribution.

Our models consider the case of convex combination of mean and CVaR, but CVaR can be used as a building block for more complicated risk measures.

Based on the nature of our procedures, we think that most of our results could be extended in a straightforward manner to cover spectral risk measures based on finite combination of CVaR. Some interesting examples could come from convex combination of CVaR risk measures with different tail levels, such as 5% combined with 1%. Another convex combination, which is called average absolute deviation from median, can be found in Denneberg [24].

Future research could include further characterization of the statistical properties of the proposed upper bound estimators or could focus on the type of approximation functions useful for the importance sampling schemes. Further future work could include application of the upper bound estimator in other problem settings in which multi-stage stochastic programs see pervasive use, such as hydroelectric scheduling under inflow uncertainty. We believe that other risk measures will lend themselves to our ideas, and developing and analyzing analogous estimators is another topic for further research.

Our numerical experiments could be also enhanced to provide extensive comparison of all three presented models. With the assumption of stage-wise independence, scenario reduction techniques should be studied in order to solve problems with very large number of stages. More general structures without the stage-wise independence assumption would provide another topic for further application of our ideas. In such case, traditional SDDP algorithm does not apply and the procedures presented in this thesis should be further analyzed and possibly extended to handle the general case.

Bibliography

[1] KOZMÍK, V. and KOPA, M. *Stability of Mean-Risk Models with Log-Normal Distribution of Returns*. In: DLOUHÝ, M. and SKOČDOPOLOVÁ, V., eds. Proceedings of the 29^{th} International Conference on Mathematical Methods in Economics. Praha: Professional Publishing, 2011, pp. 363–368, ISBN 978-80-7431-058-4.

[2] KOZMÍK, V. *Multistage risk-averse asset allocation with transaction costs*. In: RAMÍK, J. and STAVÁREK, D., eds. Proceedings of the 30^{th} International Conference on Mathematical Methods in Economics. Karviná: Silesian University in Opava, School of Business Administration in Karviná, 2012, pp. 455–460, ISBN 978-80-7248-779-0.

[3] KOZMÍK, V. and MORTON, D. *Evaluating policies in risk-averse multistage stochastic programming*. Mathematical Programming. 2014, doi: 10.1007/s10107-014-0787-8.

[4] DUPAČOVÁ, J. and KOZMÍK, V. *Structure of Risk-Averse Multistage Stochastic Programs*. OR Spectrum. 2014, doi: 10.1007/s00291-014-0379-2.

[5] KOZMÍK, V. *On Variance Reduction of Mean-CVaR Monte Carlo Estimators*. Computational Management Science. 2014, doi: 10.1007/s10287-014-0225-7.

[6] DUPAČOVÁ, J. and KOZMÍK, V. *Stress Testing for Risk-Averse Stochastic Programs*. To appear. 2014.

[7] ACERBI, C. *Spectral measures of risk: A coherent representation of subjective risk aversion*. Journal of Banking and Finance. 2002, vol. 9, pp. 1505–1518.

[8] ARTZNER, P., DELBAEN, F., EBER, J.-M.and HEATH, D. *Coherent measures of risk*. Mathematical Finance. 1999, vol. 9, pp. 203–228.

[9] ARTZNER, P., DELBAEN, F., EBER, J.-M., HEATH, D. and KU, H. *Coherent multiperiod risk adjusted values and Bellman's principle*. Annals of Operations Research. 2007, vol. 152, pp. 2–22.

[10] BAYRAKSAN, G. and MORTON, D. P. *Assessing solution quality in stochastic programs via sampling*. In: OSKOOROUCHI, M., GRAY, P. and GREENBERG, H., eds. Tutorials in operations research. Hannover: Informs. 2009, pp. 102–122, ISBN 978-1-877640-24-7.

[11] BAYRAKSAN, G. and MORTON, D. P. *A sequential sampling procedure for stochastic programming*. Operations Research. 2011, vol. 59, pp. 898–913.

[12] BEALE, E.M.L. On minimizing a convex function subject to linear inequalities. Journal of the Royal Statistical Society, Series B. 1955, vol. 17, pp. 173–184.

[13] BENDERS, J. F.: *Partitioning procedures for solving mixed-variables programming problems*. Numerische Mathematik. 1962, vol. 4, pp. 238–252

[14] BERNOULLI, D. *Exposition of a new theory on the measurement of risk*. 1738. In: SOMMER, L., translator. Econometrica. 1954, vol. 22, pp. 22-36.

[15] BIRGE, J. R. *Decomposition and partitioning methods for multistage stochastic linear programs*. Operations Research. 1985, vol. 33, pp. 989–1007.

[16] BIRGE, J. R. and LOUVEAUX, F. V.: *A multicut algorithm for two-stage stochastic linear programs*. European Journal of Operational Research. 1988, vol. 34, pp. 384–392.

[17] BLUM, M., FLOYD, R. W., PRATT, V., RIVEST, R. L. and TARJAN, R. E. *Linear time bounds for median computations*. In: FISCHER, P. C., ZEIGER, H. P., ULLMAN, J. D. and ROSENBERG, A. L., eds. Proceedings of the fourth annual ACM symposium on Theory of computing. Denver: ACM, 1972 pp. 119-124.

[18] BONNANS, J. F. and SHAPIRO, A. *Perturbation Analysis of Optimization Problems*. New York: Springer, 2000, ISBN 9781461271291.

[19] CARPENTIER, P., CHANCELIER, J. P., COHEN, G., DE LARA, M. and GIRARDEAU, P. *Dynamic consistency for stochastic optimal control problems*. Annals of Operations Research. 2012, vol. 200, pp. 247–263.

[20] CHEN, Z. L., POWELL, W. B. *Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse*. Journal of Optimization Theory and Applications. 1999, vol. 102, pp. 497–524.

[21] CHIRALAKSANAKUL, A. and MORTON, D. P.: *Assessing policy quality in multi-stage stochastic programming*. Stochastic Programming E-Print Series. 2004, vol. 12. Available at: http://www.speps.org/.

[22] COIN-OR FOUNDATION. *COIN-OR Optimization Tools* [software]. [cit. 2014-04-14] Available at: http://www.coin-or.org/.

[23] DANTZIG, G.B. *Linear programming under uncertainty*. Management Science. 1955, vol. 1, pp. 197–206.

[24] DENNEBERG, D.. *Premium calculation: Why standard deviation should be replaced by absolute deviation*. ASTIN Bulletin International Actuarial Association. 1990, vol. 20, pp. 181–190.

[25] DE QUEIROZ, A. R. and MORTON, D. P. *Sharing cuts under aggregated forecasts when decomposing multi-stage stochastic programs*. Operations Research Letters. 2013, vol. 41, pp. 311–316.

[26] DONOHUE, C. J. and BIRGE, J. R. *The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse*. Algorithmic Operations Research. 2006, vol. 1, pp. 20–30.

[27] DUPAČOVÁ, J.. *Stability and sensitivity analysis for stochastic programming*. Annals of Operations Research. 1990, vol. 27, pp. 115–142.

[28] DUPAČOVÁ, J. *Postoptimality for multistage stochastic linear programs*. Annals of Operations Research. 1995, vol. 56, pp. 65–78.

[29] DUPAČOVÁ, J. *Scenario based stochastic programs: Resistance with respect to sample*. Annals of Operations Research. 1996, vol. 64, pp. 21–38.

[30] DUPAČOVÁ, J. *Risk objectives in two-stage stochastic programming problems*. Kybernetika. 2008, vol. 44, pp. 227–242.

[31] DUPAČOVÁ, J., CONSIGLI, G. and WALLACE, S. W. *Scenarios for multistage stochastic programs*. Annals of Operations Research. 2000, vol. 100, pp. 25–53.

[32] DUPAČOVÁ, J., HURT, J. and ŠTĚPÁN, J. *Stochastic Modeling in Economics and Finance*. Dordrecht: Kluwer Academic Publishers, 2002, ISBN 1-4020-0840-6.

[33] DUPAČOVÁ, J., GRÖWE-KUSKA, N. and RÖMISCH, W. *Scenario reduction in stochastic programming: An approach using probability metrics*. Mathematical Programming. 2003, vol. 95, pp. 493–511.

[34] DUPAČOVÁ, J. and POLÍVKA, J.. *Stress testing for VaR and CVaR*. Quantitiative Finance. 2007, vol. 7, pp. 411–421.

[35] DUPAČOVÁ, J., BERTOCCHI, M. and MORIGGIA, V. *Testing the structure of multistage stochastic programs*. Computational Management Science. 2009, vol. 6, pp. 161–185.

[36] DUPAČOVÁ, J. and KOPA, M. *Robustness in stochastic programs with risk constraints*. Annals of Operations Research. 2012, vol. 200, pp. 55–74.

[37] DUPAČOVÁ, J. and KOPA, M. *Robustness of optimal portfolios under risk and stochastic dominance constraints*. European Journal of Operational Research. 2014, vol. 234, pp. 434–441.

[38] EICHHORN, A. and RÖMISCH, W. *Polyhedral risk measures in stochastic programming*. SIAM Journal on Optimization. 2005, vol. 16, pp. 69–95.

[39] FIACCO, A. V. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. New York: Academic Press, 1983, ISBN 0-12-254450-1.

[40] GOOR, Q., KELMAN, R. and TILMANT, A. *Optimal multipurpose-multireservoir operation model with variable productivity of hydropower plants*. Journal of Water Resources Planning and Management. 2011, vol. 137, pp. 258–267.

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

[42] GUROBI OPTIMIZATION. *Gurobi Optimizer* [software]. [cit. 2014-04-14] Available at: http://www.gurobi.com/products/gurobi-optimizer/gurobi-overview.

[43] HEITSCH, H. RÖMISCH, W. *Scenario tree reduction for multistage stochastic programs*. Computational Management Science. 2009, vol. 6, pp. 117–133.

[44] HESTERBERG, T. C. *Weighted average importance sampling and defensive mixture distributions*. Technometrics. 1995, vol. 37, pp. 185–194.

[45] HIGLE, J. L. and SEN, S.: *Stochastic Decomposition: An Algorithm for Two-Stage Linear Programs with Recourse*. Mathematics of Operations Research. 1991, vol. 16, pp. 650–669.

[46] HIGLE, J. L., RAYCO B. and SEN, S.. *Stochastic scenario decomposition for multistage stochastic programs*. IMA Journal of Management Mathematics. 2010, vol. 21, pp. 39–66.

[47] HOMEM-DE-MELLO, T., DE MATOS, V. L. and FINARDI, E. C.: *Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling*. Energy Systems. 2011, vol. 2, pp. 1-31.

[48] IBM. *IBM ILOG CPLEX Optimization Studio* [software]. [cit. 2014-04-14] Available at: http://www.ibm.com/software/integration/optimization/cplex-optimization-studio/.

[49] INFANGER, G. and MORTON, D. P. *Cut sharing for multistage stochastic linear programs with interstage dependency*. Mathematical Programming. 1996, vol. 75, pp. 241–256.

[50] KLEYWEGT, A. J., SHAPIRO, A. and HOMEM-DE-MELLO, T. *The sample average approximation method for stochastic discrete optimization*. SIAM Journal on Optimization. 2001, vol. 12, pp. 479–502.

[51] KNOPP, R. *Remark on algorithm 334 [G5]: normal random deviates*. Communications of the ACM. 1966, vol. 12, pp. 281.

[52] KOVACEVIC, R. and PFLUG, G.CH. *Time consistency and information monotonicity of multiperiod acceptability functionals*. In: ALBRECHER, H., RUNGGALDIER,W. J. and SCHACHERMAYER, W., eds. Radon Series on Computational and Applied Mathematics. 2009, pp. 1–23.

[53] KROKHMAL, P., ZABARANKIN, M. and URYASEV, S. *Modeling and Optimization of Risk*. Surveys in Operations Research and Management Science. 2011, vol. 16, pp. 49–66.

[54] KUPPER, M. and SCHACHERMAYER, W. *Representation results for law invariant time consistent functions*. Mathematics and Financial Economics. 2009, vol. 2, pp. 189–210.

[55] L'ECUYER, P. *L'Ecuyer random streams generator* [software]. [cit. 2014-04-14] Available at: http://www.iro.umontreal.ca/~lecuyer/myftp/streams00/.

[56] LINOWSKY, K. and PHILPOTT, A. B. *On the convergence of sampling-based decomposition algorithms for multi-stage stochastic programs*. Journal of Optimization Theory and Applications. 2005, vol. 125, pp. 349–366.

[57] MARKOWITZ, H. M. *Portfolio Selection*. The Journal of Finance. 1952, vol. 7, pp. 77–91.

[58] MARKOWITZ, H. M. *Portfolio Selection: Efficient Diversification of Investments*. New York: John Wiley & Sons, 1959.

[59] VON NEUMANN, J. and MORGENSTERN, O. *Theory of Games and Economic Behavior*. Princeton: Princeton University Press, 1944.

[60] NICTA. *Armadillo C++ linear algebra library* [software]. [cit. 2014-04-14] Available at: http://arma.sourceforge.net/.

[61] PENNANEN, T. and PERKKIÖ, A.-P. *Stochastic programs without duality gaps*. Mathematical Programming Series B. 2012, vol. 136, pp. 91–110.

[62] PEREIRA, M. V. F. and PINTO, L. M. V. G. *Multi-stage stochastic optimization applied to energy planning*. Mathematical Programming. 1991, vol. 52, pp. 359–375.

[63] PFLUG, G. CH. *Scenario tree generation for multiperiod financial optimization by optimal discretization*. Mathematical Programming. 2001, vol. 89, pp. 251–271.

[64] PFLUG, G. CH. and RÖMISCH, W. *Modeling, Measuring and Managing Risk*. Singapore: World Scientific Publishing, 2007, ISBN 978-9812707406.

[65] PFLUG, G. CH. and PICHLER, A. *Approximations for Probability Distributions and Stochastic Optimization Problems*. In: BERTOCCHI, M., CONSIGLI, G. and DEMPSTER, M. A. H., eds. Stochastic Optimization Methods in Finance and Energy. New York: Springer, 2011, ISBN 978-1-4419-9585-8.

[66] PHILPOTT, A. B. and GUAN, Z. *On the convergence of sampling-based methods for multi-stage stochastic linear programs*. Operations Research Letters. 2008, vol. 36, pp. 450–455.

[67] PHILPOTT, A. B. and DE MATOS, V. L. *Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion*. European Journal of Operational Research. 2012, vol. 218, pp. 470–483.

[68] PHILPOTT, A. B., DALLAGI, A. and GALLET, E. *On Cutting Plane Algorithms and Dynamic Programming for Hydroelectricity Generation*. In: PFLUG, G. CH., KOVACEVIC, R. M. and VESPUCCI, M. T., eds. Handbook of Risk Management in Energy Production and Trading. New York: Springer, 2013, ISBN 978-1461490340.

[69] PHILPOTT, A. B., DE MATOS, V. L. and FINARDI, E. C. *On Solving Multistage Stochastic Programs with Coherent Risk Measures*. Operations Research. 2013, vol. 61, pp. 957–970.

[70] REBENNACK, S., FLACH, B., PEREIRA, M. V. F. and PARDALOS, P. M. *Stochastic hydro-thermal scheduling under* CO_2 *emissions constraints*. IEEE Transactions on Power Systems. 2012, vol. 27, pp. 58–68.

[71] ROCKAFELLAR, R. T. and WETS, R. J.-B. *Nonanticipativity and* $\mathcal{L}^1$ *martingales in stochastic optimization problems*. Mathematical Programmming Study. 1976, vol. 6, pp. 170–187.

[72] ROCKAFELLAR, R. T. and WETS, R. J-B. *Scenarios and policy aggregation in optimization under uncertainty*. Mathematics of Operations Research. 1991, vol. 16, pp. 119–147.

[73] ROCKAFELLAR, R. T. and WETS, R. J-B. *Variational Analysis*. Berlin: Springer, 1998, ISBN 978-3-642-02431-3.

[74] ROCKAFELLAR, R. T. and URYASEV, S. *Conditional value at risk for general loss distributions*. Journal of Banking & Finance. 2002, vol. 26, pp. 1443–1471.

[75] RÖMISCH, W. *Stability of Stochastic Programming Problems*. In: RUSZCZYŃSKI, A. and SHAPIRO, A., eds. Handbooks in Operations Research and Management Science: Stochastic Programming. Amsterdam: Elsevier, 2003, ISBN 978-0444508546.

[76] ROORDA, B. and SCHUMACHER, J. M. *Time consistency conditions for acceptability measures, with an application to Tail Value at Risk*. Insurance Mathematics and Economics. 2007, vol. 40, pp. 209–230.

[77] RUDLOFF, B, STREET, A. and VALLADAO, D. *Time consistency and risk averse dynamic decision models: Definition, interpretation and practical consequences*. European Journal of Operational Research. 2014, vol. 234, pp. 743–750.

[78] RUSZCZYŃSKI, A.: *A regularized decomposition method for minimizing a sum of polyhedral functions*. Mathematical Programming. 1986, vol. 35, pp. 305–333.

[79] RUSZCZYŃSKI, A. *Risk-averse dynamic programming for Markov decision processes*. Mathematical Programming. 2010, vol. 125, pp. 235–261.

[80] RUSZCZYŃSKI, A. and SHAPIRO, A. *Conditional risk mappings*. Mathematics of Operations Research. 2006, vol. 31, pp. 544–561

[81] SHAPIRO, A. *Inference of statistical bounds for multistage stochastic programming problems*. Mathematical Methods of Operations Research. 2003, vol. 58, pp. 57–68.

[82] SHAPIRO, A. *On a time consistency concept in risk averse multistage stochastic programming*. Operations Research Letters. 2009, vol. 37, pp. 143–147.

[83] SHAPIRO, A. *Analysis of stochastic dual dynamic programming method*. European Journal of Operational Research. 2011, vol. 209, pp. 63–72.

[84] SHAPIRO, A. *Time consistency of dynamic risk measures*. Operations Research Letters. 2012, vol. 40, pp. 436–439.

[85] SHAPIRO, A. and NEMIROVSKI, A. *On Complexity of Stochastic Programming Problems*. Applied Optimization. 2005, vol. 99, pp. 111–146.

[86] SHAPIRO, A., DENTCHEVA, D. and RUSZCZYŃSKI, A. *Lectures on Stochastic Programming: Modeling and Theory*. Philadelphia: SIAM-Society for Industrial and Applied Mathematics, 2009, ISBN 978-1107025127

[87] SHAPIRO, A., TEKAYA, W., DA COSTA, J. P. and SOARES, M. P. *Risk neutral and risk averse stochastic dual dynamic programming method*. European Journal of Operational Research. 2013, vol. 224, pp. 375–391.

[88] SEN, S. and ZHOU, Z.: *Multistage stochastic decomposition: a bridge between stochastic programming and approximate dynamic programming*. SIAM Journal on Optimization. 2014, vol. 24, pp. 127–153

[89] VAN SLYKE, R. M. and WETS, J.-B.: *L-Shaped linear programs with applications to optimal control and stochastic programming*. SIAM Journal of Applied Mathematics. 1969, vol. 17, pp. 638–663.

[90] WEST, M. and HARRISON, J.: *Bayesian Forecasting and Dynamic Models, Second Edition*. New York: Springer. 1997, ISBN 978-0387947259.