

Evaluating policies in risk-averse multi-stage stochastic programming

Václav Kozmík · David P. Morton

Received: 4 March 2013 / Accepted: 29 April 2014 / Published online: 13 May 2014
© Springer-Verlag Berlin Heidelberg and Mathematical Optimization Society 2014

Abstract We consider a risk-averse multi-stage stochastic program using conditional value at risk as the risk measure. The underlying random process is assumed to be stage-wise independent, and a stochastic dual dynamic programming (SDDP) algorithm is applied. We discuss the poor performance of the standard upper bound estimator in the risk-averse setting and propose a new approach based on importance sampling, which yields improved upper bound estimators. Modest additional computational effort is required to use our new estimators. Our procedures allow for significant improvement in terms of controlling solution quality in SDDP-style algorithms in the risk-averse setting. We give computational results for multi-stage asset allocation using a log-normal distribution for the asset returns.

Keywords Multi-stage stochastic programming · Stochastic dual dynamic programming · Importance sampling · Risk-averse optimization

Mathematics Subject Classification (2010) 90C15 · 49M27

1 Introduction

We formulate and solve a multi-stage stochastic program, which uses conditional value at risk (CVaR) as the measure of risk. Our solution procedure is based on stochastic

V. Kozmík (✉)

Department of Probability and Mathematical Statistics,
Charles University in Prague, Faculty of Mathematics and Physics, Prague, Czech Republic
e-mail: vaclav@kozmik.cz

D. P. Morton

Graduate Program in Operations Research and Industrial Engineering,
The University of Texas at Austin, Austin, TX, USA
e-mail: mortondp@gmail.com

dual dynamic programming (SDDP), which has been employed successfully in a range of applications, exhibiting good computational tractability on large-scale problem instances; see, e.g., [10, 12, 13, 17, 31, 32].

There has been very limited application of SDDP to models with the type of risk measure that we use, which involves a form of CVaR that is nested to ensure a notion of time consistency. See Ruszczyński [36] and Shapiro [40] for discussions of time-consistent risk measures in multi-stage stochastic optimization. A standard multi-stage recourse formulation has 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 time-consistent risk-averse formulations.

We are aware of three solutions that have been proposed in the literature to circumvent the difficulty we have just described. First, we can solve a risk-neutral version of the problem instance under some suitable termination criterion, and then fix the number of iterations of SDDP to solve the risk-averse model under nested CVaR. Philpott and de Matos [29] 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 [41]. However, the associated computational effort grows exponentially in the number of stages, and as Shapiro [41] discusses, the bound can be loose. Third, a non-statistical deterministic upper bound is proposed in Philpott et al. [30] based on using an inner approximation scheme. This approach is attractive in that it does not have sampling-based error, but as discussed in [30] the upper bound does not scale well as the number of state variables grows.

The purpose of this article is to propose, analyze, and computationally demonstrate a new upper bound estimator for SDDP algorithms under a nested CVaR risk measure. The computational effort required to form our bound grows linearly in the number of time stages, is not limited to models with a modest number of state variables, and the estimation procedure fits flawlessly in the standard SDDP framework. Moreover, our bound is significantly tighter than the estimator based on conditional sampling, which further facilitates application of natural termination criteria, which 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 as we illustrate with an asset allocation model.

SDDP originated in the work of Pereira and Pinto [26], and inspired a number of related algorithms [7, 10, 23, 28], which aim to improve its efficiency. Nested Benders' decomposition algorithm [5] applied to a multi-stage stochastic program requires computational effort that grows exponentially in the number of stages. SDDP-style algorithms have computational effort per iteration that instead 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 [9, 19]. SDDP-style algorithms extend to handle other types of interstage dependency, such as combining the usual finer grain SDDP dependency with a Markov chain governing a coarser grain state of the system [25, 29] or with a coarser grain scenario tree [32]. SDDP algorithms can also be coupled with a Markov decision process (MDP) and approximate dynamic programming employed on a scenario lattice for the MDP [24]. The ideas we present should be useful in these extensions of SDDP.

Other important algorithms designed to solve multi-stage stochastic programs include extensions of stochastic decomposition to the multi-stage case [16, 37] and progressive hedging [33]. While these algorithms have been developed in the risk-neutral setting, they extend in natural ways to handle the risk measure we consider here, with the caveat that the nested CVaR expression can be exactly computed only when the scenario tree is of modest size. When sampling is required, these algorithms may also benefit from the type of estimator we propose.

Risk-averse stochastic optimization has received significant attention in recent years because of its attractive properties for decision makers. The properties required of *coherent* risk measures, introduced in Artzner et al. [2], are now widely accepted for time-static risk-averse optimization. Many risk measures are known to satisfy these properties; for an overview see, for instance, Krokmal et al. [21]. A number of proposals have been put forward to extend coherent risk measures to handle multi-stage stochastic optimization. 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 . While there are multiple approaches, in addition to coherency, time consistency may be desired. This latter property states 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 measures that fail to meet this condition. While it is beyond the scope of this article, the ideas behind our proposed estimators may also apply to models with CVaR-style risk measures that do not satisfy time consistency; see Definition 3.29 of Pflug and Römisch [27], Example 6.43 of Shapiro et al. [38], and Example 3.11 of Eichhorn and Römisch [11].

The main approach to construct time-consistent risk measures involves so-called conditional risk measures, introduced in Ruszczyński and Shapiro [35]. The construction is based on nesting of the risk measures, conditional on the state of the system. While this ensures time consistency, it leads to the computational difficulties that we describe above. (See Philpott and de Matos [29] and Shapiro [41] for further discussion.) With time-consistent CVaR chosen as the risk measure, we propose a new approach to upper bound estimation to overcome these computational difficulties.

We organize the remainder of this article as follows. We present our risk-averse multi-stage model in Sect. 2 and briefly review the SDDP algorithm in Sect. 3. Section 4 extends this description to the risk-averse case. We develop and analyze the proposed upper bound estimators in Sect. 5, and we provide computational results for two asset allocation models, both with and without transaction costs, in Sect. 6. We conclude and discuss ideas for future work in Sect. 7.

2 Multi-stage risk-averse model

We formulate a multi-stage stochastic program with a nested CVaR risk measure in the same manner as Shapiro [41], largely following his notation. Hence, we provide a brief problem statement. The model has random parameters in stages $t = 2, \dots, T$, denoted $\xi_t = (\mathbf{c}_t, \mathbf{A}_t, \mathbf{B}_t, \mathbf{b}_t)$, which are stage-wise independent and governed by a known, or well-estimated, distribution. The parameters of the first stage, $\xi_1 = (\mathbf{c}_1, \mathbf{A}_1, \mathbf{b}_1)$, are assumed to be known when we make decision \mathbf{x}_1 , but only a probability distribution governing future realizations, ξ_2, \dots, ξ_T , is assumed known. The realization of ξ_2 is known when decisions \mathbf{x}_2 must be made and so on to stage T . We denote the data process up to time t by $\xi_{[t]}$, meaning $\xi_{[t]} = (\xi_1, \dots, \xi_t)$.

Our model allows specification of a different risk aversion coefficient and confidence level, denoted $\lambda_t, \alpha_t \in [0, 1]$, respectively, at each time stage, $t = 2, \dots, T$. In order to provide the nested formulation of the model we introduce the following operator, which forms a weighted sum of expectation and risk associated with random loss Z :

$$\rho_{t, \xi_{[t-1]}} [Z] = (1 - \lambda_t) \mathbf{E} [Z | \xi_{[t-1]}] + \lambda_t \text{CVaR}_{\alpha_t} [Z | \xi_{[t-1]}]. \tag{1}$$

In Eq. (1), CVaR penalizes losses in the upper α tail of Z with a typical value of α being 0.05.

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

$$\min_{\mathbf{x}_1} \mathbf{c}_1^\top \mathbf{x}_1 + \rho_{2, \xi_{[1]}} \left[\min_{\mathbf{x}_2} \mathbf{c}_2^\top \mathbf{x}_2 + \dots + \rho_{T, \xi_{[T-1]}} \left[\min_{\mathbf{x}_T} \mathbf{c}_T^\top \mathbf{x}_T \right] \right], \tag{2}$$

where the first-, second-, and final stage minimizations are constrained by $\mathbf{A}_1 \mathbf{x}_1 = \mathbf{b}_1, \mathbf{x}_1 \geq 0$; $\mathbf{A}_2 \mathbf{x}_2 = \mathbf{b}_2 - \mathbf{B}_2 \mathbf{x}_1, \mathbf{x}_2 \geq 0$; and, $\mathbf{A}_T \mathbf{x}_T = \mathbf{b}_T - \mathbf{B}_T \mathbf{x}_{T-1}, \mathbf{x}_T \geq 0$, respectively. We assume model (2) is feasible, has relatively complete recourse, and has a finite optimal value. The special case with $\lambda_t = 0, t = 2, \dots, T$, is risk-neutral because we then minimize expected cost.

Model (2) is distinguished from other possible approaches to characterizing risk, by taking the risk measure as a function of the recourse value at each stage. This ensures time consistency of the risk measure. See Rudloff et al. [34] and Ruszczyński [36] for discussions of a conditional certainty-equivalent interpretation from utility theory for such nested formulations. A solution to model (2) is a policy, and time consistency means that the resulting policy has a natural interpretation that lends itself to implementation along a sample path with realizations that unfold sequentially.

Our model, with the nested risk measure, allows a dynamic programming formulation to be developed, as is described in [29, 41]. Using as the definition of conditional value at risk,

$$\text{CVaR}_\alpha [Z] = \min_u \left(u + \frac{1}{\alpha} \mathbf{E} [Z - u]_+ \right),$$

where $[\cdot]_+ \equiv \max\{\cdot, 0\}$, we can write

$$\begin{aligned} \min_{\mathbf{x}_1, u_1} & \mathbf{c}_1^\top \mathbf{x}_1 + \lambda_2 u_1 + Q_2(\mathbf{x}_1, u_1) \\ \text{s.t.} & \mathbf{A}_1 \mathbf{x}_1 = \mathbf{b}_1 \\ & \mathbf{x}_1 \geq 0. \end{aligned} \tag{3}$$

The recourse value $Q_t(\mathbf{x}_{t-1}, \xi_t)$ at stage $t = 2, \dots, T$ is given by:

$$\begin{aligned} Q_t(\mathbf{x}_{t-1}, \xi_t) &= \min_{\mathbf{x}_t, u_t} \mathbf{c}_t^\top \mathbf{x}_t + \lambda_{t+1} u_t + Q_{t+1}(\mathbf{x}_t, u_t) \\ \text{s.t.} & \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t - \mathbf{B}_t \mathbf{x}_{t-1} \\ & \mathbf{x}_t \geq 0, \end{aligned} \tag{4}$$

where

$$Q_{t+1}(\mathbf{x}_t, u_t) = \mathbf{E} \left[(1 - \lambda_{t+1}) Q_{t+1}(\mathbf{x}_t, \xi_{t+1}) + \frac{\lambda_{t+1}}{\alpha_{t+1}} [Q_{t+1}(\mathbf{x}_t, \xi_{t+1}) - u_t]_+ \right]. \tag{5}$$

We take $Q_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} \equiv 0$ so that the objective function of model (4) reduces to $\mathbf{c}_T^\top \mathbf{x}_T$ when $t = T$.

In contrast to a multi-stage formulation rooted in expected utility, our multi-stage model with CVaR has an additional decision variable, u_t , which estimates the value-at-risk level. The recourse value at stage t depends on ξ_t rather than $\xi_{[t]}$, because we assume the process to be stage-wise independent. After introducing the auxiliary variables, u_t , the problem seems to be converted to the simpler case, involving only expectations of an additive utility. This impression may lead to the false conclusion that a traditional SDDP-style algorithm can be applied. The nested nonlinearity arising from the positive-part function precludes this, as we illustrate in the next example.

Example 1 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:

$$\mathbf{E} [u_2(Z_2) + \mathbf{E} [u_3(Z_3) | \xi_{[2]}]] = \mathbf{E} [u_2(Z_2)] + \mathbf{E} [u_3(Z_3)].$$

However, this additive form does not hold under CVaR. While we can write the composite risk measure as:

$$\text{CVaR}_\alpha [Z_2 + \text{CVaR}_\alpha [Z_3 | \xi_{[2]}]] = \text{CVaR}_\alpha [\text{CVaR}_\alpha [Z_2 + Z_3 | \xi_{[2]}]],$$

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

$$\text{CVaR}_\alpha [Z_2 + \text{CVaR}_\alpha [Z_3 | \xi_{[2]}]] \leq \text{CVaR}_\alpha [Z_2] + \text{CVaR}_\alpha [\text{CVaR}_\alpha [Z_3 | \xi_{[2]}]],$$

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

It is for the reasons illustrated in Example 1 that Philpott and de Matos [29] and Shapiro [41] point to the lack of a good upper bound estimator for model (2) when the problem has more than a very small number of stages. The natural conditional sampling estimator, discussed in [29,41], has computational effort that grows exponentially in the number of stages. The following example points to a second issue associated with estimating CVaR.

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

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

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

The inefficiency pointed to in Example 2 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 Eq. (1), 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 what follows 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. Before turning to our estimator, we discuss SDDP and its application to our risk-averse formulation in the next two sections.

3 Stochastic dual dynamic programming

We use stochastic dual dynamic programming to approximately solve model (2). SDPP does not operate directly on model (2). Instead, we first form a sample average approximation (SAA) of model (2), and SDDP approximately solves that SAA. Thus in our context SDDP forms estimators by sampling within an empirical scenario tree. We describe below how we form the empirical scenario tree. Then in the remainder of this article we restrict attention to solving that SAA via SDDP. See Shapiro [39] for a discussion of asymptotics of SAA for multi-stage problems, Philpott and Guan [28] for convergence properties of SDDP, and Chiralaksanakul and Morton [8] for procedures to assess the quality of SDDP-based policies.

Again, we assume $\xi_t, t = 2, \dots, T$, to be stage-wise independent. We further assume that for each stage $t = 2, \dots, T$ there is a known (possibly continuous) distribution P_t of ξ_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 $\hat{P}_t, t = 2, \dots, T$, and the associated empirical scenario tree is interstage independent. The scenarios generated by this procedure are equally probable, but this is not required by SDDP.

Distribution \hat{P}_t has D_t realizations and stage t has N_t scenarios, where $N_t = \prod_{i=2}^t D_i$. 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, but several tractable dependency structures are discussed in Sect. 1.

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 $a(j_t) : \hat{\Omega}_t \rightarrow \hat{\Omega}_{t-1}$, which specifies the unique stage $t - 1$ ancestor for the stage t scenario j_t . Similarly, we use $\Delta(j_t) : \hat{\Omega}_t \rightarrow 2^{\hat{\Omega}_{t+1}}$ 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 $\xi_t^{j_t}$, $j_t \in \hat{\Omega}_t$. At the last stage, we have $\xi_T^{j_T}$, $j_T \in \hat{\Omega}_T$, 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 $t = T, T - 1, \dots, 2$. 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 [26,41]. The simplest SDDP algorithm applies to the risk-neutral version of our model, which means setting $\lambda_t = 0$ for $t = 1, \dots, T$ in Eq. (1) and model (2) or equivalently in (3–5). We denote the recourse value for the risk-neutral version of our model by $Q_t^N(\mathbf{x}_{t-1}, \xi_t)$, which for $t = 2, \dots, T$, is given by:

$$\begin{aligned}
 Q_t^N(\mathbf{x}_{t-1}, \xi_t) &= \min_{\mathbf{x}_t} \mathbf{c}_t^\top \mathbf{x}_t + Q_{t+1}^N(\mathbf{x}_t) \\
 \text{s.t. } \mathbf{A}_t \mathbf{x}_t &= \mathbf{b}_t - \mathbf{B}_t \mathbf{x}_{t-1} \\
 \mathbf{x}_t &\geq 0,
 \end{aligned}
 \tag{6}$$

where

$$Q_{t+1}^N(\mathbf{x}_t) = \mathbf{E} \left[Q_{t+1}^N(\mathbf{x}_t, \xi_{t+1}) \right],
 \tag{7}$$

and where $Q_{T+1}^N(\cdot) \equiv 0$. The risk-neutral formulation is completed via model (3) with $\lambda_2 = 0$ and $Q_2(\mathbf{x}_1, u_1)$ replaced by $Q_2^N(\mathbf{x}_1)$.

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, $Q_{t+1}^N(\mathbf{x}_t)$. On a forward pass we sample a number of linear paths through the tree. As we solve a sequence of master programs (which we specify below) along these forward paths, the cuts 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. 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.

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 .

Model (8) acts as a master program for its stage $t + 1$ descendant scenarios and acts as a subproblem for its stage $t - 1$ ancestor:

$$\hat{Q}_t = \min_{\mathbf{x}_t, \theta_t} \mathbf{c}_t^\top \mathbf{x}_t + \theta_t \tag{8a}$$

$$\text{s.t. } \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t - \mathbf{B}_t \mathbf{x}_{t-1} \quad : \pi_t \tag{8b}$$

$$\theta_t \geq \hat{Q}_{t+1}^j + \left(\mathbf{g}_{t+1}^j\right)^\top (\mathbf{x}_t - \mathbf{x}_t^j), \quad j = 1, \dots, C_t \tag{8c}$$

$$\mathbf{x}_t \geq 0. \tag{8d}$$

Decision variable θ_t in the objective function (8a), coupled with cut constraints in (8c), forms the outer linearization of the recourse function $Q_{t+1}^N(\mathbf{x}_t)$ from model (6) and Eq. (7). The structural and nonnegativity constraints in (8b) and (8d) simply repeat the same constraints from model (6). 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 $\hat{Q}_t, \hat{Q}_{t+1}^j, \mathbf{g}_{t+1}^j$, etc. we suppress this index for notational simplicity.

As we indicate in constraint (8b), we use π_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 $\mathbf{x}_{t-1} = \mathbf{x}_{t-1}^{a(j_t)}$ and with $\xi_t = \xi_t^{j_t}$ in model (8), we refer to that model as $\text{sub}(j_t)$. Given model $\text{sub}(j_t)$ and its solution \mathbf{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 $\text{sub}(j_{t+1})$, where $j_{t+1} \in \Delta(j_t)$ indexes all descendant nodes of j_t . This yields optimal values $\hat{Q}_{t+1}^{j_{t+1}}(\mathbf{x}_t)$ and dual solutions $\pi_{t+1}^{j_{t+1}}$ for $j_{t+1} \in \Delta(j_t)$. We then form

$$\mathbf{g}_{t+1}^{j_{t+1}} = - \left(\mathbf{B}_{t+1}^{j_{t+1}}\right)^\top \pi_{t+1}^{j_{t+1}}, \tag{9}$$

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

$$\hat{Q}_{t+1} = \frac{1}{D_{t+1}} \sum_{j_{t+1} \in \Delta(j_t)} \hat{Q}_{t+1}^{j_{t+1}} \tag{10}$$

$$\mathbf{g}_{t+1} = \frac{1}{D_{t+1}} \sum_{j_{t+1} \in \Delta(j_t)} \mathbf{g}_{t+1}^{j_{t+1}}. \tag{11}$$

As we indicate above, $\hat{Q}_{t+1} = \hat{Q}_{t+1}(\mathbf{x}_t)$ and $\mathbf{g}_{t+1} = \mathbf{g}_{t+1}(\mathbf{x}_t)$, and hence $\hat{Q}_{t+1} = \hat{Q}_{t+1}(\mathbf{x}_t)$ and $\mathbf{g}_{t+1} = \mathbf{g}_{t+1}(\mathbf{x}_t)$ but we suppress this dependency for notational simplicity. We also suppress the j_t index on \hat{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. For simplicity in stating the SDDP algorithm below, we assume we have known lower bounds L_t on the recourse functions.

Algorithm 1 Stochastic dual dynamic programming algorithm

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

2. Solve the stage 1 master program ($t = 1$) and obtain $\mathbf{x}_1^k, \theta_1^k$.
 Let $\underline{z}_k = \mathbf{c}_1^\top \mathbf{x}_1^k + \theta_1^k$.
3. Forward pass: sample i.i.d. paths from $\hat{\Omega}$ and index them by S^k .
 For all $j \in S^k$ {
 For $t = 2, \dots, T$ {
 Form and solve sub(j_t) to obtain $(\mathbf{x}_t^{j_t})^k$;
 }
 }
 }

Form the upper bound estimator:

$$\bar{z}_k = \mathbf{c}_1^\top \mathbf{x}_1^k + \frac{1}{|S^k|} \sum_{j \in S^k} \sum_{t=2}^T (\mathbf{c}_t^{j_t})^\top (\mathbf{x}_t^{j_t})^k. \tag{12}$$

4. If a stopping criterion, given \bar{z}_k and \underline{z}_k , is satisfied then stop and output first stage solution $\mathbf{x}_1 = \mathbf{x}_1^k$ and lower bound $\underline{z} = \underline{z}_k$.
5. Backward pass:
 For $t = T - 1, \dots, 1$ {
 For all $j \in S^k$ {
 For all descendant nodes $j_{t+1} \in \Delta(j_t)$ {
 Form and solve sub(j_{t+1}) to obtain $\hat{Q}_{t+1}^{j_{t+1}}$ and $\pi_{t+1}^{j_{t+1}}$;
 Calculate $\mathbf{g}_{t+1}^{j_{t+1}}$ using formula (9);
 }
 Calculate optimal value \hat{Q}_{t+1} using equation (10);
 Calculate cut gradient \mathbf{g}_{t+1} using equation (11);
 Append the resulting cut to the collection (8c) for stage t ;
 }
 }
 }
6. Let $k = k + 1$ and goto step 2.

See Bayraksan and Morton [4] and Homem-de-Mello et al. [17] for stopping rules that can be employed in step 4.

4 Risk-averse approach

We must modify the SDDP algorithm of Sect. 3 to handle the risk-averse model of Sect. 2. 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, \mathbf{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 another decision variable. Extending the development from the previous section, the stage t subproblem in the risk-averse case is given by:

$$\begin{aligned}
 \hat{Q}_t &= \min_{\mathbf{x}_t, u_t, \theta_t} \mathbf{c}_t^\top \mathbf{x}_t + \lambda_{t+1} u_t + \theta_t \\
 \text{s.t. } & \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t - \mathbf{B}_t \mathbf{x}_{t-1} \quad : \pi_t \\
 & \theta_t \geq \hat{Q}_{t+1}^j + \left(\mathbf{g}_{t+1}^j\right)^\top \left[\left(\mathbf{x}_t, u_t\right) - \left(\mathbf{x}_t^j, u_t^j\right)\right], \quad j = 1, \dots, C_t \\
 & \mathbf{x}_t \geq 0.
 \end{aligned}
 \tag{13}$$

While $\hat{Q}_{t+1} = \hat{Q}_{t+1}(\mathbf{x}_t)$, we now have $\hat{Q}_{t+1} = \hat{Q}_{t+1}(\mathbf{x}_t, u_t)$ and $\mathbf{g}_{t+1} = \mathbf{g}_{t+1}(\mathbf{x}_t, u_t)$ as a function of both the stage t decision and the VaR level. The subgradient of $\hat{Q}_{t+1}(\mathbf{x}_t)$ is still computed by Eq. (9). However, the function value $\hat{Q}_{t+1}(\mathbf{x}_t, u_t)$ and its subgradient have to be adjusted to respect the differences between the risk-neutral and risk-averse cases. See Shapiro [41] for the formulas that replace Eqs. (10) and (11) to obtain \hat{Q}_{t+1} and \mathbf{g}_{t+1} . In modifying the SDDP algorithm for the risk-averse formulation, these new cut computations are used in the backward pass of step 5 of Algorithm 1 to provide the piecewise linear outer approximation of $Q_{t+1}(\mathbf{x}_t, u_t)$. Shapiro et al. [42] describe an approach in which the VaR level does not appear explicitly in model (13). However, as we describe below our upper bound estimators require explicit VaR levels.

One issue that remains concerns evaluation of an upper bound; i.e., an analog of estimator (12) for the risk-averse setting. As we illustrate in Example 1, we cannot expect an analogous additive estimator to be appropriate for the risk-averse setting. Example 1 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 $(\mathbf{c}_T^{j_T})^\top \mathbf{x}_T^{j_T}$. For the stage $T - 1$ ancestor scenario $j_{T-1} = a(j_T)$ we must calculate

$$(\mathbf{c}_{T-1}^{j_{T-1}})^\top \mathbf{x}_{T-1}^{j_{T-1}} + \lambda_T u_{T-1}^{j_{T-1}} + Q_T(\mathbf{x}_{T-1}^{j_{T-1}}, u_{T-1}^{j_{T-1}}).$$

The question that remains is how to estimate $Q_T(\mathbf{x}_{T-1}^{j_{T-1}}, u_{T-1}^{j_{T-1}})$. 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 1. This means that based on Eq. (5) we estimate $Q_T(\mathbf{x}_{T-1}^{j_{T-1}}, u_{T-1}^{j_{T-1}})$ by

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

Removing the expectation operator in Eq. (5), the associated recursion of the objective function in model (4) and Eq. (5) yields the following recursive estimator of $Q_t(\mathbf{x}_{t-1}^{j_{t-1}}, u_{t-1}^{j_{t-1}})$ for $t = 2, \dots, T$:

$$\hat{v}_t(\xi_{t-1}^{j_{t-1}}) = (1 - \lambda_t) \left((\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}(\xi_t^{j_t}) \right) + \lambda_t u_{t-1}^{j_{t-1}} + \frac{\lambda_t}{\alpha_t} \left[(\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}(\xi_t^{j_t}) - u_{t-1}^{j_{t-1}} \right]_+, \tag{14}$$

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

$$\hat{v}(\xi^j) = \mathbf{c}_1^\top \mathbf{x}_1 + \hat{v}_2. \tag{15}$$

Because the first stage parameters, $\xi_1^{j_1}$, are deterministic we simply write $\hat{v}_2 = \hat{v}_2(\xi_1^{j_1})$, dropping its argument.

Having selected scenario j and solved all node subproblems (13) associated with realizations $\xi_1^{j_1}, \dots, \xi_T^{j_T}$ along the sample path, we form the estimator recursively as follows. We start at the stage $T - 1$ node, compute $\hat{v}_T(\xi_{T-1}^{j_{T-1}})$, substitute it into formula (14) for $t = T - 1$ to obtain $\hat{v}_{T-1}(\xi_{T-2}^{j_{T-2}})$ and so on until we obtain \hat{v}_2 and hence can compute the value of $\hat{v}(\xi^j)$ via Eq. (15). Then if $\xi^j, j = 1, \dots, M$, are i.i.d. sample paths, sampled from the scenario tree’s empirical distribution as in step 3 in Algorithm 1, the corresponding upper bound estimator is given by:

$$U^n = \frac{1}{M} \sum_{j=1}^M \hat{v}(\xi^j). \tag{16}$$

We use the “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 (16), which is the natural analog of (12), 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 2 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 Eq. (14) 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 $\hat{v}_t(\xi_{t-1}^{j_{t-1}})$ is large, this increases the likelihood that preceding values are also large and hence multiplied by $\alpha_{t-1}^{-1}, \alpha_{t-2}^{-1}, \dots$ 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 [41] describes an estimator which uses more nodes to estimate the recourse value. This estimator for a 3-stage problem is obtained by sampling, and solving subproblems associated with i.i.d. realizations $\xi_2^1, \dots, \xi_2^{M^2}$ from the second stage and for each of these solving subproblems to estimate the future risk measure using i.i.d. realizations $\xi_3^1, \dots, \xi_3^{M^3}$ from the third stage. This requires solving subproblems at a total of $M_2 M_3$ nodes. More generally under this approach, given a stage $T - 1$ scenario $\xi_{T-1}^{j_{T-1}}$ we estimate the recourse function value by:

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

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

$$\begin{aligned} \hat{v}_t^e(\xi_{t-1}^{j_{t-1}}) &= \frac{1}{M_t} \sum_{j_t=1}^{M_t} \left[(1 - \lambda_t) \left((\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}^e(\xi_t^{j_t}) \right) \right. \\ &\quad \left. + \lambda_t u_{t-1}^{j_{t-1}} + \frac{\lambda_t}{\alpha_t} \left[(\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}^e(\xi_t^{j_t}) - u_{t-1}^{j_{t-1}} \right]_+ \right]. \end{aligned} \tag{17}$$

and finally for the upper bound estimator we compute:

$$U^e = \mathbf{c}_1^\top \mathbf{x}_1 + \hat{v}_2^e. \tag{18}$$

Shapiro [41] discusses two significant problems with the upper bound estimator (18). First, the estimator requires solving an exponential number, $\prod_{t=2}^T M_t$, of sub-problems in the number of stages (Thus the “e” superscript.) and hence is impractical unless T is small. Second, as we examine further in Sect. 6, even when we can afford to compute the bound provided by (18), the bound is not very tight. For these reasons, estimator (18) is not typically used in practice.

Philpott and de Matos [29] 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 we 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. [30]. 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 (18), and it does not have sampling error. However, as Philpott et al. [30] 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 [30] in Sect. 6.

When we restrict attention to statistical upper bound estimators, we have three possible approaches at this point. The two upper bound estimators (16) and (18) 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 an approach which provides no guarantees on solution quality, even if reasonable empirical performance has been reported in the literature. In the next section 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 previous approaches.

5 Improved upper bound estimation

We overcome the shortcomings of the upper bound estimators (16) and (18) by first focusing on the main issue causing the estimators to be poor: A relatively small fraction of the sampled scenario-tree nodes contribute to estimating CVaR, for reasons we illustrate in Example 2. To sample in a better manner we assume that for every stage, $t = 2, \dots, T$, we can cheaply evaluate a real-valued *approximation function*, $h_t(\mathbf{x}_{t-1}, \xi_t)$, which estimates the recourse value of our decisions \mathbf{x}_{t-1} after the random parameters ξ_t have been observed.

The functions h_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 (18), we instead evaluate h_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 h_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 $t + 2, \dots, T$. Sometimes this is possible via an approximation of the recourse value associated with the system’s state. For example, when dealing with some asset allocation models, we may use current wealth to define h_t .

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

$$f_t(\xi_t) = \frac{1}{D_t} \mathbf{I} \left[\xi_t \in \{ \xi_t^1, \dots, \xi_t^{D_t} \} \right], \tag{19}$$

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

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 $g_t(\xi_t | \mathbf{x}_{t-1})$. This pmf is tailored specifically for use with CVaR. Alternative pmfs would be needed to apply the proposed ideas to other risk measures. Given the current state of the system we can compute the value at risk for our approximation function, $u_h = \text{VaR}_{\alpha_t} [h_t(\mathbf{x}_{t-1}, \xi_t)]$, and partition the nodes corresponding to $\xi_t^1, \dots, \xi_t^{D_t}$ into two groups by comparing their approximate value to u_h . In particular, the importance sampling pmf is:

$$g_t(\xi_t | \mathbf{x}_{t-1}) = \begin{cases} \frac{1}{2} \frac{1}{\lfloor \alpha_t D_t \rfloor} \mathbf{I} \left[\xi_t \in \left\{ \xi_t^1, \dots, \xi_t^{D_t} \right\} \right], & \text{if } h_t(\mathbf{x}_{t-1}, \xi_t) \geq u_h \\ \frac{1}{2} \frac{1}{D_t - \lfloor \alpha_t D_t \rfloor} \mathbf{I} \left[\xi_t \in \left\{ \xi_t^1, \dots, \xi_t^{D_t} \right\} \right], & \text{if } h_t(\mathbf{x}_{t-1}, \xi_t) < u_h, \end{cases} \tag{20}$$

where the $\lfloor \cdot \rfloor$ operator rounds down to the nearest integer. The factor of $\frac{1}{2}$ in the pmf $g_t(\xi_t | \mathbf{x}_{t-1})$ modifies the probability masses so that we are equally likely to draw sample observations above and below $u_h = \text{VaR}_{\alpha_t} [h_t(\mathbf{x}_{t-1}, \xi_t)]$. 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 .

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

$$\mathbf{E}_{f_t} [Z] = \mathbf{E}_{g_t} \left[Z \frac{f_t}{g_t} \right],$$

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 $\mathbf{E}_f [\cdot]$ and $\mathbf{E}_g [\cdot]$.

We can form an estimator similar to (16), except that we employ our importance sampling distributions, g_t , in place of the empirical distributions, f_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 , ξ^j , we form estimator (15), which uses recursion (14) and preserves the good scalability of the estimator with the number of stages. We carry out this for a set of samples drawn using the new measure g_t to select the sample paths.

Thus we have weights for each stage of

$$w_t(\xi_t | \mathbf{x}_{t-1}) = \frac{f_t(\xi_t)}{g_t(\xi_t | \mathbf{x}_{t-1})},$$

which yields weights along a sample path of

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

and an estimator of the form

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

This estimator is a weighted sum of the upper bounds (15) for the sampled scenarios. Normalizing the weights reduces the variability of the estimator (see Hesterberg [15]) and yields:

$$U^i = \frac{1}{\sum_{j=1}^M w(\xi^j)} \sum_{j=1}^M w(\xi^j) \hat{v}(\xi^j), \tag{21}$$

where “ $\hat{\cdot}$ ” indicates that the estimator uses importance sampling. We summarize the development so far in the following proposition.

Proposition 1 *Assume model (2) has relatively complete recourse and interstage independence. Let z^* denote the optimal value of model (2) under the empirical distributions, $\hat{P}_t, t = 2, \dots, T$, generated by i.i.d. sampling. Let ξ denote a sample path selected under the empirical distribution, and let $\hat{v}(\xi)$ be defined by (15) for that sample path. Then $\mathbf{E}_f [\hat{v}(\xi)] \geq z^*$. Furthermore if $\xi^j, j = 1, \dots, M$, are i.i.d. and generated by the pmfs (20) and U^i is defined by (21) then $U^i \rightarrow \mathbf{E}_f [\hat{v}(\xi)]$, w.p.1, as $M \rightarrow \infty$.*

Proof The optimal value of model (2) as reformulated in model (3) yields z^* . Along sample path ξ , under the assumption of relatively complete recourse, the cuts in sub-problems (13) generate a feasible policy in the space of the (\mathbf{x}_t, u_t) variables. Removing the expectation operator in equation (5), the associated recursion of the objective function in model (4) and equation (5) coincides with the recursion in equation (14). Taking expectations yields $\mathbf{E}_f [\hat{v}(\xi)] \geq z^*$.

By the law of large numbers we have that

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{j=1}^M w(\xi^j) = 1, \text{ w.p.1.} \tag{22}$$

For ξ generated by the empirical pmfs (19) and for each ξ^j , generated by the pmfs (20), we have

$$\mathbf{E}_g [w(\xi^j) \hat{v}(\xi^j)] = \mathbf{E}_f [\hat{v}(\xi)].$$

Thus by the law of large numbers we have

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{j=1}^M w(\xi^j) \hat{v}(\xi^j) = \mathbf{E}_f [\hat{v}(\xi)], \text{ w.p.1.} \tag{23}$$

From the definition of U^i from (21) we have

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

Using a converging-together result with Eqs. (22) and (23) we then have

$$U^i \rightarrow \mathbf{E}_f [\hat{v}(\xi)], \text{ w.p.1,}$$

as $M \rightarrow \infty$.

□

In the sense made precise in Proposition 1, estimator (21) provides an asymptotic upper bound on the optimal value of model (2). The naive estimator U^n of (16) is an unbiased and consistent estimator of $\mathbf{E}_f [\hat{v}(\xi)]$. However, if the functions h_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^i will have smaller variance than U^n . That said, we view estimator (21) 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, $\mathbf{Q}_t = \mathbf{Q}_t(\mathbf{x}_{t-1}, \xi_t)$ and $h_t = h_t(\mathbf{x}_{t-1}, \xi_t)$.

Assumption 1 For every stage $t = 2, \dots, T$ and decision \mathbf{x}_{t-1} the approximation function h_t satisfies:

$$\mathbf{Q}_t \geq \text{VaR}_{\alpha_t} [\mathbf{Q}_t] \text{ if and only if } h_t \geq \text{VaR}_{\alpha_t} [h_t].$$

Under Assumption 1, we can strengthen the estimator through a reformulation. Given a sample path ξ we modify the recursive estimator (14) for $t = 2, \dots, T$ as:

$$\hat{v}_t^h(\xi_{t-1}^{j_{t-1}}) = (1 - \lambda_t) \left((\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}^h(\xi_t^{j_t}) \right) \tag{24a}$$

$$+ \lambda_t u_{t-1}^{j_{t-1}} + \mathbf{I}[h_t \geq \text{VaR}_{\alpha_t} [h_t]] \frac{\lambda_t}{\alpha_t} \left[(\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}^h(\xi_t^{j_t}) - u_{t-1}^{j_{t-1}} \right]_+, \tag{24b}$$

where $\hat{v}_{T+1}^h(\xi_T^{j_T}) \equiv 0$, and we let

$$\hat{v}^h(\xi) = \mathbf{c}_1^\top \mathbf{x}_1 + \hat{v}_2^h. \tag{25}$$

Like the estimators U^n and U^i , which are based on (14) and (15), we note that the estimator we propose next, based on Eqs. (24) and (25), requires explicit estimation of the VaR-level by the u_{t-1} decision variables. With $\xi^j, j = 1, \dots, M$, i.i.d. from the pmfs (20) we form the upper bound estimator:

$$U^h = \frac{1}{\sum_{j=1}^M w(\xi^j)} \sum_{j=1}^M w(\xi^j) \hat{v}^h(\xi^j). \tag{26}$$

Proposition 2 Assume the hypotheses of Proposition 1, let ξ denote a sample path selected under the empirical distribution, let $\hat{v}^h(\xi)$ be defined by (25) for that sample path, and let Assumption 1 hold. Then $\mathbf{E}_f [\hat{v}^h(\xi)] \geq z^*$. If $\xi^j, j = 1, \dots, M$, are i.i.d. and generated by the pmfs (20) and U^h is defined by (26) then $U^h \rightarrow \mathbf{E}_f [\hat{v}^h(\xi)]$, w.p.1, as $M \rightarrow \infty$. Furthermore if subproblems (13) induce the same policy for both $\hat{v}(\xi)$ and $\hat{v}^h(\xi)$ then $\mathbf{E}_f [\hat{v}(\xi)] \geq \mathbf{E}_f [\hat{v}^h(\xi)]$.

Proof Let $(\mathbf{x}_1, u_1), \dots, (\mathbf{x}_{T-1}, u_{T-1}), \mathbf{x}_T$ be the feasible sequence to models (3) and (4) for $t = 2, \dots, T$, specified by (13) along sample path ξ . The result $\mathbf{E}_f [\hat{v}(\xi)] \geq \mathbf{E}_f [\hat{v}^h(\xi)]$ holds because $\mathbf{I}[h_t \geq \text{VaR}_{\alpha_t} [h_t]]$ can preclude some positive terms in the recursion (24) that are included in (14).

The terms in (24b) are used to estimate CVaR. Thus to establish the rest of the proposition it suffices to show:

$$\begin{aligned} & \text{VaR}_{\alpha_t} [Q_t] + \frac{1}{\alpha_t} \mathbf{E} \left[[Q_t - \text{VaR}_{\alpha_t} [Q_t]]_+ \right] \\ & \leq u_{t-1} + \frac{1}{\alpha_t} \mathbf{E} \left[\mathbf{I}[h_t \geq \text{VaR}_{\alpha_t} [h_t]] [Q_t - u_{t-1}]_+ \right] \end{aligned}$$

because the rest of the proof then follows in the same fashion as that of Proposition 1.

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

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

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 $u_{t-1} \geq \text{VaR}_{\alpha_t} [Q_t]$, and the last equality follows from Assumption 1.

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

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

where the inequality holds because $\mathbf{P} [Q_t \geq \text{VaR}_{\alpha_t} [Q_t]] \geq \alpha_t$ and $u_{t-1} < \text{VaR}_{\alpha_t} [Q_t]$. (Note that we would instead have $\mathbf{P} [Q_t \geq \text{VaR}_{\alpha_t} [Q_t]] = \alpha_t$ if we were in the continuous case.) This completes the proof as the desired result holds in both cases. □

As Proposition 2 indicates, U^h provides an asymptotic upper bound estimator for the optimal value of model (2). It also provides a tighter upper bound in expectation than estimators U^n and U^i . We also anticipate that estimator U^h will have smaller variance than U^i provided the “induce the same policy” hypothesis holds. Note the same-policy hypothesis is not needed for the proposition’s consistency result. As we discuss in Sect. 4, when a sample path is such that the positive-part term in (14) is positive that term is multiplied by $\alpha_t^{-1} = 20$ (say), and this increases the likelihood that as we decrement t we obtain large values that are repeatedly multiplied by $\alpha_{t-1}^{-1}, \alpha_{t-2}^{-1}$, etc. This repeated multiplication should occur for some samples, but it can also occur when it should not. The indicator function in U^h helps avoid this issue and hence tends to reduce variance.

Under Assumption 1, the approximation function, h_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 1 to incorporate the notion of what we call a *margin function*, $m_t(\mathbf{x}_{t-1}, \xi_t)$, in order for our type of upper bound estimator to address a broader class of stochastic programs. Under Assumption 2, 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 2 For every stage $t = 2, \dots, T$ and decision \mathbf{x}_{t-1} we have real-valued functions $h_t(\mathbf{x}_{t-1}, \xi_t)$ and $m_t(\mathbf{x}_{t-1}, \xi_t)$ which satisfy:

$$\text{if } h_t < m_t \text{ then } \mathbf{Q}_t < \text{VaR}_{\alpha_t} [\mathbf{Q}_t].$$

Given a sample path ξ we modify the recursive estimators (14) and (24) for $t = 2, \dots, T$ as:

$$\begin{aligned} \hat{v}_t^m(\xi_{t-1}^{j_{t-1}}) &= (1 - \lambda_t) \left((\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}^m(\xi_t^{j_t}) \right) \\ &+ \lambda_t u_{t-1}^{j_{t-1}} + \mathbf{I}[h_t \geq m_t] \frac{\lambda_t}{\alpha_t} \left[(\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{v}_{t+1}^m(\xi_t^{j_t}) - u_{t-1}^{j_{t-1}} \right]_+, \end{aligned} \tag{27}$$

where $\hat{v}_{T+1}^m(\xi_T^{j_T}) \equiv 0$, and we let

$$\hat{v}^m(\xi) = \mathbf{c}_1^\top \mathbf{x}_1 + \hat{v}_2^m. \tag{28}$$

With $\xi^j, j = 1, \dots, M$, i.i.d. and from the pmfs (20), which use functions h_t , we form the upper bound estimator:

$$U^m = \frac{1}{\sum_{j=1}^M w(\xi^j)} \sum_{j=1}^M w(\xi^j) \hat{v}^m(\xi^j). \tag{29}$$

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

approximation function via the pmfs (20). However, the estimator based on (28) is more generally applicable than the estimator based on (25) because we can drop Assumption 1 and instead require only the weaker implication of Assumption 2.

Proposition 3 *Assume the hypotheses of Proposition 1, let ξ denote a sample path selected under the empirical distribution, let $\hat{V}^m(\xi)$ be defined by (28) for that sample path, and let Assumption 2 hold. Then $E_f[\hat{V}^m(\xi)] \geq z^*$. If $\xi^j, j = 1, \dots, M$, are i.i.d. and generated by the pmfs (20) and U^m is defined by (29) then $U^m \rightarrow E_f[\hat{V}^m(\xi)]$, w.p.1, as $M \rightarrow \infty$. Furthermore if subproblems (13) induce the same policy for both $\hat{V}(\xi)$ and $\hat{V}^m(\xi)$ then $E_f[\hat{V}(\xi)] \geq E_f[\hat{V}^m(\xi)]$. Finally, if Assumption 1 also holds and subproblems (13) induce the same policy for both $\hat{V}^m(\xi)$ and $\hat{V}^h(\xi)$ then $E_f[\hat{V}^m(\xi)] \geq E_f[\hat{V}^h(\xi)]$.*

Proof We have:

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

where the first inequality comes from following the steps of the proof of Proposition 2 (in both of the cases considered) and the second inequality follows from Assumption 2. From this we have $E_f[\hat{V}^m(\xi)] \geq z^*$, and the consistency result for U^m follows in the same manner as in the proof of Proposition 1. Inequality $E_f[\hat{V}(\xi)] \geq E_f[\hat{V}^m(\xi)]$ holds because $\mathbf{I}[h_t \geq m_t]$ can preclude some positive terms in the recursion (27) that are included in (14). Finally, $E_f[\hat{V}^m(\xi)] \geq E_f[\hat{V}^h(\xi)]$ holds because under Assumptions 1 and 2 the indicator $\mathbf{I}[h_t \geq m_t]$ allows inclusion of some positive terms that the indicator $\mathbf{I}[h_t \geq \text{VaR}_{\alpha_t} [h_t]]$ does not. \square

In order to ensure that U^h 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 1, and this limits applicability of the estimator in some cases. Assumption 2 weakens considerably this requirement, and widens the applicability of estimator U^m . While U^m again provides an asymptotic upper bound estimator for the optimal value of model (2), the price we pay is that it is weaker than U^h as Proposition 3 indicates.

For the types of approximation and margin functions (h_t and m_t) that we envision, our importance-sampling estimators (U^i, U^h , and U^m) require modest additional computation relative to estimator U^n , which uses samples from the empirical pmfs (19). 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 h_t and m_t at each of these D_t nodes and determining $u_h = \text{VaR}_{\alpha_t} [h_t]$, which can be done by sorting with effort $O(D_t \log D_t)$ or in linear time in D_t (see [6]). 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 Sects. 3–5, 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. For example, in computing the quantile u_h we should form the cumulative distribution function rather than simply sorting.

6 Computational results

We present computational results for applying SDDP with the upper bound estimators we describe in Sects. 4 and 5 to two asset allocation models under our CVaR risk measure. We present results for our four new upper bound estimators: (i) U^n from equation (16); (ii) U^i from Eq. (21); (iii) U^h from Eq. (26); and, (iv) U^m from Eq. (29). We compare their performance with that of the existing upper bound estimator from the literature: U^e from Eq. (18). The two asset allocation models we consider differ only in whether we include transaction costs or not. Without transaction costs we can use estimator U^h , but we can only use estimator U^m when we include transaction costs.

We begin with a simple asset allocation model without transaction costs and emphasize that our primary purpose is to compare the upper bound estimators as opposed to building a high-fidelity model for practical use. At stage t the decisions \mathbf{x}_t denote the allocations (in units of a multiple of a base currency, say USD), and \mathbf{p}_t denotes 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 (4) specializes to:

$$Q_t(\mathbf{x}_{t-1}, \xi_t) = \min_{\mathbf{x}_t, u_t} -\mathbf{1}^\top \mathbf{x}_t + \lambda_{t+1} u_t + Q_{t+1}(\mathbf{x}_t, u_t) \tag{30a}$$

$$\text{s.t. } \mathbf{1}^\top \mathbf{x}_t = \mathbf{p}_t^\top \mathbf{x}_{t-1} \tag{30b}$$

$$\mathbf{x}_t \geq 0, \tag{30c}$$

except that in the first stage: (i) the right-hand side of (30b) is instead 1 and (ii) because $-\mathbf{1}^\top \mathbf{x}_1$ is then identically -1 , we drop this constant from the objective function.

The assets in our allocation model consist of the stock market indices DJIA, 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. An empirical scenario tree was then constructed by sampling from the log-normal distribution, using the polar method [20] for sampling the underlying normal distributions. The L’Ecuyer random generator [22] was used to generate the required uniform random variables. We implemented the SDDP algorithm in C++ software, using CPLEX [18] to solve the required linear programs and the Armadillo [1] library for matrix computations. The confidence level was set to $\alpha_t = 5\%$ and the risk coefficients were set to $\lambda_t = \frac{t-1}{T}$, $t = 2, \dots, T$ so that risk aversion increases in later stages. Table 1 shows the sizes of the empirical scenario trees for our test problem instances.

Table 1 Sizes of empirical scenario trees for test problem instances

Stages (T)	Descendants per node (D_t)	Total scenarios ($ \Omega^T $)
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	$\approx 10^{24}$

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

$$h_t(\mathbf{x}_{t-1}, \xi_t) = -\mathbf{p}_t^\top \mathbf{x}_{t-1}.$$

Note that this function meets the requirements of Assumption 1, because when we have no transaction costs the specific allocations in the vector \mathbf{x}_{t-1} can be rebalanced with no loss. This is captured mathematically in model (30) in that $h_t = -\mathbf{p}_t^\top \mathbf{x}_{t-1}$ acts as the sole state variable; i.e., the equality constraint of (30b) could be used to replace the first term in the objective function of (30a). Hence, we see \mathbf{Q}_t also depends solely on h_t and that dependence is monotonic in h_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 1 for the risk-averse model. Specifically, SDDP was terminated when \underline{z} agreed across the four runs and did not improve by more than 10^{-6} over 10 iterations. A total of 100 iterations of SDDP sufficed to accomplish this for problem instances with $T = 2, \dots, 5$ and a total of 200 iterations sufficed for the larger instances with $T = 10$ and 15. For estimators $U^n, U^i,$ and U^h on problem instances with $T = 2, 3, 4,$ and 5 we used respective sample sizes of $M = 1,001, 501, 334,$ and 251. In this way, forming the estimators required solving around 1,000 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^e we must specify a sample size M_t for each stage: For $T = 2$ we used $M_2 = 1,000$. For $T = 3$ we used $M_2 = M_3 = 32$ because this means forming the estimator requires solving $32^2 \approx 1,000$ linear programs and this allows for a fair comparison with the single-path estimators $U^n, U^i,$ and U^h . With similar reasoning for $T = 4$ we used $M_t = 11 \forall t$, and for $T = 5$ we used $M_t = 6 \forall t$. And for the largest value of T for which we compute $U^e, T = 10,$ we used $M_t = 3 \forall t$.

Table 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

Table 2 Comparison of four upper bound estimators, including the point estimates and their standard deviations (SD) for the model with no transaction costs

T	\underline{z}	U^n (SD)	U^i (SD)	U^h (SD)	U^e (SD)
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

of SDDP. Each cell in Table 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^h , are revealed as the number of stages grows. Note that estimators U^n , U^i and U^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 Eq. (24); compare the performance of U^i and U^h in the table. For $T = 2, \dots, 5$ the variance reduction of U^h relative to U^e grows from roughly 3 to 25 to 50 to 300. The smaller standard deviations of U^h could facilitate its use in a sensible stopping rule.

For our second set of problem instances the model incorporates transaction costs. This allows us to show how to implement our upper bound estimation procedure in a more complex model. 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. We must modify the rebalancing equation between stage $t - 1$ and stage t to include the transaction costs of $f_t \mathbf{1}^\top |x_t - x_{t-1}|$, where the $|\cdot|$ function applies component-wise. Linearizing we obtain the following special case of model (4):

$$\begin{aligned}
 Q_t(x_{t-1}, \xi_t) = & \min_{x_t, z_t, u_t} -\mathbf{1}^\top x_t + \lambda_{t+1} u_t + Q_{t+1}(x_t, u_t) \\
 & \text{s.t. } \mathbf{1}^\top x_t + f_t \mathbf{1}^\top z_t = \mathbf{p}_t^\top x_{t-1} \\
 & z_t - x_t \geq -x_{t-1} \\
 & z_t + x_t \geq x_{t-1} \\
 & x_t \geq 0.
 \end{aligned}$$

We again use the approximation function $h_t(x_{t-1}, \xi_t) = -\mathbf{p}_t^\top x_{t-1}$. With nonzero transaction costs the conditions of Assumption 1 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

$\mathbf{p}_t^\top \mathbf{x}_{t-1}$, consisting only of stock A may be preferred to another portfolio with larger value of $\mathbf{p}_t^\top \mathbf{x}_{t-1}$ 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:

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

Functions h_t and m_t satisfy Assumption 2, and we can apply the upper bound estimator U^m . This construction, of course, increases the bias of the estimator as we indicate in Proposition 3. However, if the transaction costs are modest compared to market volatility, we may expect our estimator to provide reasonable results.

Table 3 reports results in the same manner as Table 2, now comparing U^m and U^e . The value \underline{z} is computed in the same way we describe above. From Table 2 we see that the point estimate U^h 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^m from Table 3 are weaker as expected, dropping from 99.6 to 98.7 to 90.5%. Note that these same values for U^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^m over U^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.

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^n and U^h . For each estimator we used a sample size of $M = 1,000$ and the subproblems in each stage had 1,000 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^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^h grew from 6.8 s to 30.0 s to 66.5 s for the same instances. (The standard deviations associated with the run times are at most 1% of the average.) Like estimator U^n , the computational effort we require to compute U^h scales well with the number of stages. We note that the estimator U^h does have additional computational overhead, relative to U^n , that

Table 3 Comparison of the estimators for the model with transaction costs

T	\underline{z}	U^m (SD)	U^e (SD)
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

grew from an average 0.3 s to 1.2 s to 2.9 s 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^h compared to U^n . We obtained very similar scaling results with T for the estimator U^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^h and U^m) and the inner approximation bound of Philpott et al. [30]. The computational work in [30] 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), and the gap reported in Table 3 of [30] with 10,000 cuts is of similar magnitude. With 10,000 cuts the inner approximation of [30] 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.

7 Conclusion

We have presented a new approach to compute upper bound estimators for multi-stage stochastic programs that incorporate risk via CVaR. 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 and upper bounds allow for quantification of the quality of a proposed policy, which we see as important in practice.

Future research could include further characterization of the statistical properties of the proposed estimators, parametric tuning of the importance sampling distribution, or could focus on the type of approximation functions useful for the importance sampling schemes. Further future work could include application of the 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.

Acknowledgments We thank two anonymous referees and an Associate Editor for comments that improved the paper. The research was partly supported by the Czech Science Foundation through grant 402/12/G097, by the Defense Threat Reduction Agency through Grant HDTRA1-08-1-0029, and by the National Science Foundation through Grant ECCS-1162328.

References

1. Armadillo C++ linear algebra library. <http://arma.sourceforge.net/>
2. Artzner, P., Delbaen, F., Eber, J.-M., Heath, D.: Coherent measures of risk. *Math. Financ.* **9**, 203–228 (1999)
3. Bayraksan, G., Morton, D.P.: Assessing solution quality in stochastic programs via sampling. In: Oskoorouchi, M., Gray, P., Greenberg, H. (eds.) *Tutorials in Operations Research*. pp. 102–122. INFORMS, Hanover (2009). ISBN: 978-1-877640-24-7
4. Bayraksan, G., Morton, D.P.: A sequential sampling procedure for stochastic programming. *Oper. Res.* **59**, 898–913 (2011)
5. Birge, J.R.: Decomposition and partitioning methods for multistage stochastic linear programs. *Oper. Res.* **33**, 989–1007 (1985)
6. Blum, M., Floyd, R. W., Pratt, V., Rivest, R. L., Tarjan, R. E.: Linear time bounds for median computations. In: *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, pp 119–124 (1972)
7. Chen, Z.L., Powell, W.B.: Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. *J. Optim. Theory Appl.* **102**, 497–524 (1999)
8. Chiralaksanakul, A., Morton, D.P.: Assessing policy quality in multi-stage stochastic programming. *Stochastic Programming E-Print Series*, vol. 12 (2004). Available at: <http://www.speps.org/>
9. de Queiroz, A.R., Morton, D.P.: Sharing cuts under aggregated forecasts when decomposing multi-stage stochastic programs. *Oper. Res. Lett.* **41**, 311–316 (2013)
10. Donohue, C.J., Birge, J.R.: The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithm. Oper. Res.* **1**, 20–30 (2006)
11. Eichhorn, A., Römisch, W.: Polyhedral risk measures in stochastic programming. *SIAM J. Optim.* **16**, 69–95 (2005)
12. Goor, Q., Kelman, R., Tilmant, A.: Optimal multipurpose-multireservoir operation model with variable productivity of hydropower plants. *J. Water Resour. Plan. Manag.* **137**, 258–267 (2011)
13. Guigues, V.: SDDP for some interstage dependent risk averse problems and application to hydrothermal planning. *Comput. Optim. Appl.* **57**, 167–203 (2014)
14. Guigues, V., Römisch, W.: Sampling-based decomposition methods for multistage stochastic programs based on extended polyhedral risk measures. *SIAM J. Optim.* **22**, 286–312 (2012)
15. Hesterberg, T.C.: Weighted average importance sampling and defensive mixture distributions. *Technometrics* **37**, 185–194 (1995)
16. Higle, J.L., Rayco, B., Sen, S.: Stochastic scenario decomposition for multistage stochastic programs. *IMA J. Manag. Math.* **21**, 39–66 (2010)
17. Homem-de-Mello, T., de Matos, V.L., Finardi, E.C.: Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Syst.* **2**, 1–31 (2011)
18. IBM ILOG CPLEX. <http://www.ibm.com/software/integration/optimization/cplex-optimization-studio/>
19. Infanger, G., Morton, D.P.: Cut sharing for multistage stochastic linear programs with interstage dependency. *Math. Prog.* **75**, 241–256 (1996)
20. Knopp, R.: Remark on algorithm 334 [G5]: normal random deviates. *Commun. ACM* **12**, 281 (1966)
21. Krokmal, P., Zabarankin, M., Uryasev, S.: Modeling and optimization of risk. *Surv. Oper. Res. Manag. Sci.* **16**, 49–66 (2011)
22. L'Ecuyer random streams generator. <http://www.iro.umontreal.ca/~lecuyer/myftp/streams00/>
23. Linowsky, K., Philpott, A.B.: On the convergence of sampling-based decomposition algorithms for multi-stage stochastic programs. *J. Optim. Theory Appl.* **125**, 349–366 (2005)
24. Löhndorf, N., Wozabal, D., Minner, S.: Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. *Oper. Res.* **61**, 810–823 (2013)

25. Mo, B., Gjelsvik, A., Grundt, A.: Integrated risk management of hydro power scheduling and contract management. *IEEE Trans. Power Syst.* **16**, 216–221 (2001)
26. Pereira, M.V.F., Pinto, L.M.V.G.: Multi-stage stochastic optimization applied to energy planning. *Math. Prog.* **52**, 359–375 (1991)
27. Pflug, GCh., Römisch, W.: *Modeling, Measuring and Managing Risk*. World Scientific Publishing, Singapore (2007)
28. Philpott, A.B., Guan, Z.: On the convergence of sampling-based methods for multi-stage stochastic linear programs. *Oper. Res. Lett.* **36**, 450–455 (2008)
29. Philpott, A.B., de Matos, V.L.: Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *Eur. J. Oper. Res.* **218**, 470–483 (2012)
30. Philpott, A.B., de Matos, V.L., Finardi, E.C.: On solving multistage stochastic programs with coherent risk measures. *Oper. Res.* **61**, 957–970 (2013)
31. Philpott, A.B., Dallagi, A., Gallet, E.: On cutting plane algorithms and dynamic programming for hydroelectricity generation. In: Pflug, GCh., Kovacevic, R.M., Vespucci, M.T. (eds.) *Handbook of Risk Management in Energy Production and Trading*. Springer, UK (2013)
32. Rebennack, S., Flach, B., Pereira, M.V.F., Pardalos, P.M.: Stochastic hydro-thermal scheduling under CO₂ emissions constraints. *IEEE Trans. Power Syst.* **27**, 58–68 (2012)
33. Rockafellar, R.T., Wets, R.J.-B.: Scenarios and policy aggregation in optimization under uncertainty. *Math. Oper. Res.* **16**, 119–147 (1991)
34. Rudloff, B., Street, A., Valladao, D.: Time consistency and risk averse dynamic decision models: definition, interpretation and practical consequences. Available at: <http://www.optimization-online.org/>
35. Ruszczyński, A., Shapiro, A.: Conditional risk mappings. *Math. Oper. Res.* **31**, 544–561 (2006)
36. Ruszczyński, A.: Risk-averse dynamic programming for Markov decision processes. *Math. Prog.* **125**, 235–261 (2010)
37. Sen, S., Zhou, Z.: Multistage stochastic decomposition: a bridge between stochastic programming and approximate dynamic programming. *SIAM J. Optim.* **24**, 127–153 (2014)
38. Shapiro, A., Dentcheva, D., Ruszczyński, A.: *Lectures on Stochastic Programming: Modeling and Theory*. SIAM-Society for Industrial and Applied Mathematics (2009)
39. Shapiro, A.: Inference of statistical bounds for multistage stochastic programming problems. *Math. Methods Oper. Res.* **58**, 57–68 (2003)
40. Shapiro, A.: On a time consistency concept in risk averse multistage stochastic programming. *Oper. Res. Lett.* **37**, 143–147 (2009)
41. Shapiro, A.: Analysis of stochastic dual dynamic programming method. *Eur. J. Oper. Res.* **209**, 63–72 (2011)
42. Shapiro, A., Tekaya, W., da Costa, J.P., Soares, M.P.: Risk neutral and risk averse stochastic dual dynamic programming method. *Eur. J. Oper. Res.* **224**, 375–391 (2013)
43. West, M., Harrison, J.: *Bayesian Forecasting and Dynamic Models*, 2nd edn. Springer, New York (1997)