European Journal of Operational Research 209 (2011) 63–72

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

Analysis of stochastic dual dynamic programming method

Alexander Shapiro [*]

School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA

A R T I C L E   I N F O

Article history:
Received 12 January 2010
Accepted 6 August 2010
Available online 15 August 2010

Keywords:
Stochastic programming
Stochastic Dual Dynamic Programming algorithm
Sample Average Approximation method
Monte Carlo sampling
Risk averse optimization

A B S T R A C T

In this paper we discuss statistical properties and convergence of the Stochastic Dual Dynamic Programming (SDDP) method applied to multistage linear stochastic programming problems. We assume that the underline data process is stagewise independent and consider the framework where at first a random sample from the original (true) distribution is generated and consequently the SDDP algorithm is applied to the constructed Sample Average Approximation (SAA) problem. Then we proceed to analysis of the SDDP solutions of the SAA problem and their relations to solutions of the "true" problem. Finally we discuss an extension of the SDDP method to a risk averse formulation of multistage stochastic programs. We argue that the computational complexity of the corresponding SDDP algorithm is almost the same as in the risk neutral case.

1. Introduction

The goal of this paper is to analyze convergence properties of the Stochastic Dual Dynamic Programming (SDDP) approach to solve linear multistage stochastic programming problems of the form

$$\underset{\substack{A_1x_1=b_1\\ x_1\geqslant 0}}{\operatorname{Min}}\ c_1^{\mathsf T}x_1+\mathbb{E}\left[\min_{\substack{B_2x_1+A_2x_2=b_2\\ x_2\geqslant 0}} c_2^{\mathsf T}x_2+\mathbb{E}\left[\cdots+\mathbb{E}\left[\min_{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\ x_T\geqslant 0}} c_T^{\mathsf T}x_T\right]\right]\right]. \quad (1.1)$$

Components of vectors c_t, b_t and matrices A_t, B_t are modeled as random variables forming the stochastic data process[1] $\xi_t=(c_t,A_t,B_t,b_t)$, $t=2,\dots,T$, with $\xi_1=(c_1,A_1,b_1)$ being deterministic (not random). By $\xi_{[t]}=(\xi_1,\dots,\xi_t)$ we denote history of the data process up to time t. The SDDP method originated in Pereira and Pinto [11], and was extended and analyzed in several publications (e.g., [2,4,7,12]). It was assumed in those publications that the number of realizations (scenarios) of the data process is *finite*, and this assumption was essential in the implementations and analysis of the SDDP type algorithms. In many applications, however, this assumption is quite unrealistic. In forecasting models (such as ARIMA) the errors are typically modeled as having continuous (say normal or log-normal) distributions. So one of the relevant questions is what is the meaning of the introduced discretizations of the corresponding stochastic process. Related questions are convergence properties and error analysis of the method.

We make the basic assumption that the random data process is *stagewise independent*, i.e., random vector ξ_{t+1} is independent of $\xi_{[t]}=(\xi_1,\dots,\xi_t)$ for $t=1,\dots,T-1$. In some cases across stages dependence can be dealt with by adding state variables to the model. For example, suppose that parameters of the data process ξ_t other than b_t are stagewise independent (in particular are deterministic) and random vectors b_t, $t=2,\dots,T$, form a first order autoregressive process, i.e., $b_t=\Phi b_{t-1}+\varepsilon_t$, with appropriate matrix Φ and error vectors $\varepsilon_2,\dots,\varepsilon_T$ being independent of each other. Then the feasibility equations of problem (1.1) can be written as

$$b_t-\Phi b_{t-1}=\varepsilon_t,\quad B_tx_{t-1}-\Phi b_{t-1}+A_tx_t=\varepsilon_t,\quad x_t\geqslant 0,\quad t=2,\dots,T. \quad (1.2)$$

Therefore by replacing x_t with (x_t,b_t) and data process with $(c_t,A_t,B_t,\varepsilon_t)$, $t=2,\dots,T$, we transform the problem to the stagewise independent case. Of course, in this new formulation we do not need to enforce nonnegativity of the state variables b_t.

We also assume that the implementation is performed in two steps. First, a (finite) scenario tree is generated by randomly sampling from the original distribution and then the constructed problem is solved by the SDDP algorithm. A current opinion is that the approach of random generation of scenarios (the so-called Sample Average Approximation (SAA) method) is computationally intractable for solving multistage stochastic programs because of the exponential growth of the number of scenarios with increase of the number of stages (cf., [18,19]). An interesting property of the SDDP method is that the computational complexity of one run of the involved backward and forward step procedures is

* Tel.: +1 404 8946544.
E-mail address: ashapiro@isye.gatech.edu

[1] Of course, not all elements of the data vectors ξ_t should be random. For example, we can model only the right hand side vectors b_t as random while all other elements of ξ_t being fixed (known).

0377-2217/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2010.08.007

proportional to the sum of sampled data points at every stage and not to the total number of scenarios given by their product. This makes it computationally feasible to run several such backward and forward steps. Of course, this still does not give a proof of computational tractability of the true multistage problem. It also should be remembered that this nice property holds because of the stagewise independence assumption.

We also discuss an extension of the SDDP method to a risk averse formulation of multistage stochastic programs. We argue that the computational complexity of the corresponding SDDP algorithm is almost the same as in the risk neutral case.

In order to present some basic ideas we start our analysis in the next section with two-stage linear stochastic programming problems. For a discussion of basic theoretical properties of two and multi-stage stochastic programs we may refer to [21]. In Section 3 we describe the SDDP approach, based on approximation of the dynamic programming equations, applied to the SAA problem. A risk averse extension of this approach is discussed in Section 4. Finally, Section 5 is devoted to a somewhat informal discussion of this methodology.

We use the following notations and terminology throughout the paper. The notation ":=" means "equal by definition". For $a \in \mathbb{R}$, $[a]_+ := \max\{0, a\}$. By $|J|$ we denote cardinality of a finite set J. By A^{T} we denote transpose of matrix (vector) A. For a random variable Z, $\mathbb{E}[Z]$ and $\mathrm{Var}[Z]$ denote its expectation and variance, respectively. $\Pr(\cdot)$ denotes probability of the corresponding event. Given a convex function $Q(x)$ we denote by $\partial Q(x)$ its subdifferential, i.e., the set of all its subgradients, at point $x \in \mathbb{R}^n$. It is said that an affine function $\ell(x) = a + b^{\mathsf{T}}x$ is a *cutting plane*, of $Q(x)$, if $Q(x) \geqslant \ell(x)$ for all $x \in \mathbb{R}^n$. Note that cutting plane $\ell(x)$ can be strictly smaller than $Q(x)$ for all $x \in \mathbb{R}^n$. If, moreover, $Q(\bar{x}) = \ell(\bar{x})$ for some $\bar{x} \in \mathbb{R}^n$, it is said that $\ell(x)$ is a *supporting plane* of $Q(x)$. This supporting plane is given by $\ell(x) = Q(\bar{x}) + g^{\mathsf{T}}(x - \bar{x})$ for some subgradient $g \in \partial Q(\bar{x})$.

2. Two-stage programs

In this section we discuss a setting of the SDDP method applied to the following two-stage linear stochastic programming problem:

$$\operatorname*{Min}_{x \in \mathcal{X}} c^{\mathsf{T}}x + \mathcal{Q}(x), \tag{2.1}$$

where $\mathcal{X} := \{x \in \mathbb{R}^{n_1} : Ax = b, x \geqslant 0\}$, $\mathcal{Q}(x) := \mathbb{E}[Q(x,\xi)]$ and $Q(x,\xi)$ is the optimal value of the second stage problem

$$\begin{array}{ll} \operatorname*{Min}\limits_{y \in \mathbb{R}^{n_2}} & q^{\mathsf{T}}y \\ \text{s.t.} & Tx + Wy = h, y \geqslant 0. \end{array} \tag{2.2}$$

It is assumed that some/all elements of vectors q, h and matrices T, W are random. The data vector ξ is formed from elements of q, h, T, W, and the expectation in (2.1) is taken with respect to a (known) probability distribution P of ξ. In order to emphasize what probability distribution is used we sometimes write $\mathbb{E}_P[Q(x,\xi)]$ for the corresponding expectation. We use the same notation ξ to denote random vector and its particular realization; which one of these two meanings will be used in a particular situation will be clear from the context. As it was discussed in the Introduction we do not restrict our analysis to the case of a finite number of scenarios, i.e., the distribution P can be continuous. We assume, however, that we can sample, say by using Monte Carlo techniques, from the distribution of ξ.

Problem (2.2) is a linear programming problem. Its dual is the problem

$$\begin{array}{ll} \operatorname*{Max}\limits_{\pi} & \pi^{\mathsf{T}}(h - Tx) \\ \text{s.t.} & W^{\mathsf{T}}\pi \leqslant q. \end{array} \tag{2.3}$$

Both problems (2.2) and (2.3) possess optimal solutions provided that both problems are feasible. We assume that the expectation function $\mathcal{Q}(x)$ is well defined and finite valued. In particular, we assume that the second stage problem (2.2) is feasible for all $x \in \mathcal{X}$ and almost every realization of the random data. That is, we assume that the considered problem has *relatively complete recourse*. Note that for every ξ the function $Q(\cdot,\xi)$ is convex, and hence the expected value function $\mathcal{Q}(x)$ is also convex.

Since we do not make the assumption of a finite number of scenarios, several strategies are possible. Let us consider the following approach in the spirit of [11]. A random sample $\tilde{\xi}^1, \ldots, \tilde{\xi}^N$ of N (independent) realizations of the random vector ξ is generated and consequently the "true" distribution P is replaced by the (empirical) distribution P_N constructed from N scenarios $\tilde{\xi}^1, \ldots, \tilde{\xi}^N$ each taken with probability $1/N$. This results in replacing the original problem (2.1) by the so-called sample average approximation (SAA) problem

$$\operatorname*{Min}_{x \in \mathcal{X}} c^{\mathsf{T}}x + \widetilde{\mathcal{Q}}(x), \tag{2.4}$$

where

$$\widetilde{\mathcal{Q}}(x) := \mathbb{E}_{P_N}[Q(x,\xi)] = N^{-1}\sum_{j=1}^{N} Q(x, \tilde{\xi}^j). \tag{2.5}$$

We use notation $\mathfrak{S}_N := [\tilde{\xi}^1, \ldots, \tilde{\xi}^N]$ for this sample.

A possible strategy is to apply the SDDP algorithm to the SAA rather than the original problem. That is, we assume now that the sample $\mathfrak{S}_N$ is fixed[2] and discuss implementation of the SDDP algorithm to the obtained SAA problem. At the kth iteration the SDDP algorithm performs the following procedure referred to as the *backward* step. Let $\bar{x}_k \in \mathcal{X}$ be a current first stage solution and $\mathfrak{Q}_k(x)$ be an approximation of $\widetilde{\mathcal{Q}}(x)$, given by maximum of a finite number of its *supporting* planes. Next a subgradient $g_k \in \partial\widetilde{\mathcal{Q}}(\bar{x}_k)$ is computed and the new supporting plane

$$\ell_k(x) := \widetilde{\mathcal{Q}}(\bar{x}_k) + g_k^{\mathsf{T}}(x - \bar{x}_k) \tag{2.6}$$

is constructed. The current approximation is updated by replacing $\mathfrak{Q}_k(x)$ with $\mathfrak{Q}_{k+1}(x) := \max\{\mathfrak{Q}_k(x), \ell_k(x)\}$, i.e., the supporting plane $\ell_k(x)$ is added to the collection. Consequently $\bar{x}_k$ is updated by an optimal solution of the problem

$$\operatorname*{Min}_{x \in \mathcal{X}} c^{\mathsf{T}}x + \mathfrak{Q}_{k+1}(x). \tag{2.7}$$

In order to ensure that problem (2.7) has an optimal solution we may assume that the set $\mathcal{X}$ is nonempty and bounded. Since $\widetilde{\mathcal{Q}}(\cdot)$ is greater than or equal to its every supporting plane and $\mathfrak{Q}_k(\cdot)$ is given by maximum of a collection of supporting planes, we have that $\widetilde{\mathcal{Q}}(\cdot) \geqslant \mathfrak{Q}_k(\cdot)$, $k = 1, \ldots$, and hence the optimal value of problem (2.7) is less than or equal to the optimal value of the SAA problem (2.4). That is, values

$$\underline{\vartheta}_k := \inf_{x \in \mathcal{X}} \{c^{\mathsf{T}}x + \mathfrak{Q}_k(x)\} \quad k = 1, \ldots, \tag{2.8}$$

give lower bounds for the optimal value of the SAA problem.

In order to perform the above backward step at each iteration we need to make the following computations. The second stage problem (2.2) should be solved for $x = \bar{x}_k$ and each $\tilde{\xi}^j = (\tilde{h}_j, \widetilde{T}_j, \widetilde{W}_j, \tilde{q}_j)$, $j = 1, \ldots, N$. Let $\bar{y}_{kj}$ be an optimal solution of (2.2) and π_{kj} be an optimal solution of its dual (2.3) for $\xi = \tilde{\xi}_j$, $j = 1, \ldots, N$. Then

$$\widetilde{\mathcal{Q}}(\bar{x}_k) = N^{-1}\sum_{j=1}^{N} \tilde{q}_j^{\mathsf{T}}\bar{y}_{kj} \quad \text{and} \quad g_k = -N^{-1}\sum_{j=1}^{N} \widetilde{T}_j^{\mathsf{T}}\pi_{kj}. \tag{2.9}$$

[2] Since in the discussion below the sample is fixed we do not indicate in the notation of the function $\widetilde{\mathcal{Q}}(x)$ dependence on the sample size N.

This procedure requires solving the second stage problem N times. In order for that to be computationally feasible the sample size N should be not too large.

It could be noted that in the considered case of *two-stage* stochastic programming, the "backward step" procedure of the SDDP algorithm does not involve sampling of scenarios. It also could be noted that the above "backward step" procedure is the standard cutting plane algorithm (Kelley's cutting plane algorithm [6]) applied to the SAA problem (2.4). Its convergence properties are well known. It produces a sequence of iterates $\bar{x}_k$ which finds an optimal solution of problem (2.4) in a finite number of iterations (see, e.g., [14, section 2.2]).

Let us discuss now the so-called *forward* step of the SDDP method applied to the SAA problem (2.4). A random sample $\xi^1,\dots,\xi^M$ from the empirical distribution P_N is generated, i.e., $\xi^1,\dots,\xi^M$ is a subsample of the sample $\mathfrak{S}_N=[\tilde{\xi}^1,\dots,\tilde{\xi}^N]$. This subsample can be generated with or without replacement. In case of sampling without replacement, all $\xi^1,\dots,\xi^M$ are different from each other and $M\leqslant N$. If M is much smaller than N, then there is no significant practical difference between these two methods,[3] therefore we assume for the sake of simplicity that the sampling is with replacement. For a current solution $\bar{x}\in\mathcal{X}$, the second stage problem (2.2) is solved for $x=\bar{x}$ and each $\xi=\xi^j$, $j=1,\dots,M$. Let $\vartheta_j=q_j^{\mathsf{T}}\bar{y}_j$ be the respective optimal values of the second stage problem. Note that if $\bar{x}=\bar{x}_k$, then these optimal values were computed at the corresponding backward step. Let

$$\bar{\vartheta}:=\frac{1}{M}\sum_{j=1}^{M}\vartheta_j \quad\text{and}\quad \hat{\sigma}_\vartheta^2:=\frac{1}{(M-1)}\sum_{j=1}^{M}(\vartheta_j-\bar{\vartheta})^2 \tag{2.10}$$

be the respective sample average and sample variance of the optimal values ϑ_j. Conditional on the sample $\mathfrak{S}_N$, the average $\bar{\vartheta}$ is an unbiased estimate of $\widetilde{\mathcal{Q}}(\bar{x})$, i.e., $\mathbb{E}\left[\bar{\vartheta}|\mathfrak{S}_N\right]=\widetilde{\mathcal{Q}}(\bar{x})$, and

$$\left[\bar{\vartheta}-z_{\alpha/2}\hat{\sigma}_\vartheta/\sqrt{M},\bar{\vartheta}+z_{\alpha/2}\hat{\sigma}_\vartheta/\sqrt{M}\right] \tag{2.11}$$

gives an (approximate[4]) $100(1-\alpha)\%$ confidence interval for $\widetilde{\mathcal{Q}}(\bar{x})$. Here z_α denotes the $(1-\alpha)$-quantile of the standard normal distribution. For example, $z_{0.025}=1.96$ corresponds to the 95% confidence interval.

Remark 1. It was suggested in [11, p. 368] to stop the procedure if the current lower bound $\underline{\vartheta}_k$ is inside the confidence interval (2.11) (the same suggestion was made in [4, p.24]). This means that the procedure is stopped if $\underline{\vartheta}_k$ becomes greater than $\bar{\vartheta}-z_{\alpha/2}\hat{\sigma}_\vartheta/\sqrt{M}$. This, however, does not give any guarantee that the SAA problem was solved with a reasonable accuracy. The meaning of the confidence interval (2.11) is that with (approximate) probability $1-\alpha$ some number inside this interval gives an upper bound for the optimal value of the SAA problem. However, this upper bound could be any point of that interval and this will be too optimistic to assume that the lower end of the confidence interval gives such an upper bound. According to that criterion the larger the variance $\hat{\sigma}_\vartheta^2$ is the sooner we stop. Moreover, increasing the confidence $1-\alpha$ makes the critical value $z_{\alpha/2}$ bigger, and the lower end of the confidence interval can be made arbitrary small for sufficiently large confidence $1-\alpha$. That is, by increasing the confidence $1-\alpha$ the procedure could be stopped at any iteration by this stopping criterion; this does not make sense. A more meaningful criterion would be to stop the procedure if the difference between the upper confidence bound $\bar{\vartheta}+z_\alpha\hat{\sigma}_\vartheta/\sqrt{M}$ and the lower bound $\underline{\vartheta}_k$ is less than a prescribed accuracy level $\varepsilon>0$. This will suggest, with confidence of about $1-\alpha$, that the SAA problem is solved with accuracy ε. We will discuss this further below.

Remark 2. The above SDDP procedure is applied to the SAA problem (2.4) while actually the true problem (2.1) needs to be solved. So the natural question is how optimal solutions and the constructed optimality bounds of the SAA problem are related to their counterparts of the true problem. The SAA problem is a function of the sample $\mathfrak{S}_N=[\tilde{\xi}^1,\dots,\tilde{\xi}^N]$ and therefore in itself is random. For a (fixed) $\bar{x}\in\mathcal{X}$ the value $\widetilde{\mathcal{Q}}(\bar{x})$ is random and we have that $\mathbb{E}[\widetilde{\mathcal{Q}}(\bar{x})]=\mathcal{Q}(\bar{x})$. It follows that

$$\mathbb{E}[\bar{\vartheta}]=\mathbb{E}\left\{\mathbb{E}\left[\bar{\vartheta}|\mathfrak{S}_N\right]\right\}=\mathbb{E}[\widetilde{\mathcal{Q}}(\bar{x})]=\mathcal{Q}(\bar{x}),$$

i.e., $\bar{\vartheta}$ is an unbiased estimator of $\mathcal{Q}(\bar{x})=\mathbb{E}[Q(\bar{x},\xi)]$. In order to compute variance of $\bar{\vartheta}$ we use formula

$$\mathrm{Var}[\bar{\vartheta}]=\mathbb{E}\left[\mathrm{Var}(\bar{\vartheta}|\mathfrak{S}_N)\right]+\mathrm{Var}\left[\mathbb{E}(\bar{\vartheta}|\mathfrak{S}_N)\right], \tag{2.12}$$

where $\mathrm{Var}(\bar{\vartheta}|\mathfrak{S}_N)=\mathbb{E}\left[\left(\bar{\vartheta}-\mathbb{E}(\bar{\vartheta}|\mathfrak{S}_N)\right)^2)|\mathfrak{S}_N\right]$ is the conditional variance of $\bar{\vartheta}$ and $\mathrm{Var}\left[\mathbb{E}(\bar{\vartheta}|\mathfrak{S}_N)\right]=\mathbb{E}\left\{\left[\mathbb{E}(\bar{\vartheta}|\mathfrak{S}_N)-\mathbb{E}(\bar{\vartheta})\right]^2\right\}$ is variance of $\mathbb{E}\left[\bar{\vartheta}|\mathfrak{S}_N\right]=\widetilde{\mathcal{Q}}(\bar{x})$. We have that $\hat{\sigma}_\vartheta^2/M$ is an unbiased estimator of the first term in the right hand side of (2.12), and

$$\mathrm{Var}\left[\mathbb{E}(\bar{\vartheta}|\mathfrak{S}_N)\right]=\mathrm{Var}[\widetilde{\mathcal{Q}}(\bar{x})]=N^{-1}\mathrm{Var}[Q(\bar{x},\xi)].$$

Therefore the second term in the right hand side of (2.12) can be estimated by

$$\hat{\sigma}_Q^2=\frac{1}{N-1}\sum_{j=1}^{N}\left(Q(\bar{x},\tilde{\xi}^j)-\widetilde{\mathcal{Q}}(\bar{x})\right)^2.$$

This leads to the following $100(1-\alpha)\%$ confidence interval for the "true" expected value $\mathcal{Q}(\bar{x})$:

$$\left[\bar{\vartheta}-z_{\alpha/2}\sqrt{\hat{\sigma}_\vartheta^2/M+\hat{\sigma}_Q^2/N},\bar{\vartheta}+z_{\alpha/2}\sqrt{\hat{\sigma}_\vartheta^2/M+\hat{\sigma}_Q^2/N}\right]. \tag{2.13}$$

Of course, since $\widetilde{\mathcal{Q}}(\bar{x})$ is an unbiased estimate of $\mathcal{Q}(\bar{x})$, it is possible to construct a confidence interval for $\mathcal{Q}(\bar{x})$ directly as $\left[\widetilde{\mathcal{Q}}(\bar{x})-z_{\alpha/2}\hat{\sigma}_Q/\sqrt{N},\widetilde{\mathcal{Q}}(\bar{x})+z_{\alpha/2}\hat{\sigma}_Q/\sqrt{N}\right]$.

Remark 3. It is not difficult to show and is well known that the expectation of the optimal value of the SAA problem (2.4) is less than the optimal value of the true problem (2.1). By estimating the expectation of the optimal value of the SAA problem, may be by solving several SAA problems based on independently generated samples, and using an upper bound discussed in Remarks 1 and 2, one can estimate the optimality gap for the true problem (cf., [8,10]). Of course, by construction the optimal value $\underline{\vartheta}_{k+1}$ of problem (2.7) is less than the optimal value of the SAA problem (2.4). Therefore $\mathbb{E}[\underline{\vartheta}_k]$ is a lower bound for the optimal value of the true problem as well.

Rates of convergence of the optimal value and optimal solutions of the SAA problems to their counterparts of the true problem (2.1) are well investigated. In short, optimal values of SAA problems converge to their true counterparts at a rate of $O_p(N^{-1/2})$, which is typical for Monte Carlo type estimates. In other words in order to solve the true problem with accuracy $\varepsilon>0$ one needs to solve the SAA problem with a sample size of order $N=O(\varepsilon^{-2})$. For a de-

[3] If the sample size M is comparable to N, say N is about 3 times bigger than M, then in subsampling without replacement this should be taken into account in formulas (2.10) below (cf., [3]). However, if N is say bigger than M by a factor of 100 or more, then such corrections become practically negligible. Especially in the multistage case, discussed in the next section, the number of scenarios N is far bigger than the sample size M used in the forward step procedure.

[4] This confidence interval is based on approximation of the distribution of the average $\bar{\vartheta}$ by the corresponding normal distribution, which could be justified by application of the Central Limit Theorem.

tailed analysis of rates of convergence and statistical properties of SAA estimators we may refer to [21, Chapter 5]. Often a reasonably manageable sample size N is sufficient to approximate the true problem by an SAA problem with a practically acceptable accuracy.

It should be pointed out that the above cutting plane algorithm is not an efficient method for solving two-stage linear programs. The convergence is rather slow and there are much better algorithms for solving such problems[5] (see, e.g., [14]). We discussed it here as a preparation for an investigation of the SDDP method applied to multistage linear problems.

3. Multistage programs

Consider the linear multistage stochastic programming problem (1.1). Recall that we make the assumption that the data process $\xi_1,\dots,\xi_T$ is *stagewise independent*. Then the dynamic programming equations for problem (1.1) take the form

$$Q_t(x_{t-1},\xi_t) = \inf_{x_t\in\mathbb{R}^{n_t}} \left\{ c_t^{\mathsf T} x_t + \mathcal{Q}_{t+1}(x_t) : B_t x_{t-1} + A_t x_t = b_t, x_t \geqslant 0 \right\}, \tag{3.1}$$

where

$$\mathcal{Q}_{t+1}(x_t) := \mathbb{E}\{Q_{t+1}(x_t,\xi_{t+1})\}, \tag{3.2}$$

$t = T,\dots,2$ (with $\mathcal{Q}_{T+1}(\cdot)\equiv 0$ by definition). At the first stage problem

$$\begin{aligned} \operatorname*{Min}_{x_1\in\mathbb{R}^{n_1}} \quad & c_1^{\mathsf T} x_1 + \mathcal{Q}_2(x_1) \\ \text{s.t.} \quad & A_1 x_1 = b_1, \quad x_1 \geqslant 0, \end{aligned} \tag{3.3}$$

should be solved. We assume that the *cost-to-go* functions $\mathcal{Q}_t(\cdot)$ are finite valued, in particular we assume the relatively complete recourse.

Recall that a collection of functions $\bar{x}_t = \bar{x}_t(\xi_{[t]})$, $t = 1,\dots,T$, is said to be an (implementable) policy. Such policy gives a decision rule at every stage t of the problem based on a realization of the data process up to time t. A policy is feasible if it satisfies the feasibility constraints for almost every realization of the random data. By the classical result we have that policy $\bar{x}_t = \bar{x}_t(\xi_{[t]})$ is optimal if it satisfies the dynamic programming equations, i.e., $\bar{x}_t$ is a solution of the minimization problem in the right hand side of (3.1) (see, e.g., [21, section 3.1] for a discussion of these concepts). Note that here the cost-to-go functions $\mathcal{Q}_t(x_{t-1})$ do not depend on the data process because of the stagewise independence assumption.

We also assume that we can sample from the probability distribution P_t of the random vector ξ_t, $t = 2,\dots,T$ (recall that ξ_1 is deterministic, not random). A sample average approximation (SAA) of the "true" problem (1.1) is constructed by replacing the true distribution of ξ_t, $t = 2,\dots,T$, by the empirical distribution P_{N_t} based on a random sample

$$\tilde{\xi}_t^j = (\tilde{c}_{tj}, \widetilde{A}_{tj}, \widetilde{B}_{tj}, \tilde{b}_{tj}), \quad j = 1,\dots,N_t, \tag{3.4}$$

from P_t of size N_t. Consequently the probability distribution $P_2\times\cdots\times P_T$ of the random process $\xi_2,\dots,\xi_T$ is replaced by (finitely generated) distribution $P_{N_2}\times\cdots\times P_{N_T}$. This probability distribution (of the SAA problem) can be represented by a tree where at stage $t-1$ every node of the tree has the same branches corresponding to $\tilde{\xi}_t^1,\dots,\tilde{\xi}_t^{N_t}$. *Note that in that way the stagewise independence of the original (true) problem is preserved in the SAA problem.*

The total number of scenarios of the constructed SAA problem is $N = \prod_{t=2}^T N_t$. Even with a moderate number of scenarios per stage, say each N_t equals several hundreds, the total number of scenarios grows exponentially and quickly becomes astronomically large with increase of the number of stages. This suggests that a scenarios-based approach to multistage stochastic programming is computationally intractable (cf., [18]). On the other hand it is known that (under mild regularity conditions) in order to solve the true problem with a given accuracy $\varepsilon > 0$ the sample sizes N_t, $t = 2,\dots,T$, used to construct the SAA problem, should be of order $O(\varepsilon^{-2})$ (cf., [19],[21, section 5.8.2]). That is, if the SAA problem with reasonable sample sizes N_t could be solved in a reasonable time with a reasonable accuracy, then its solution could give a reasonable approximation for its counterpart of the true problem.

We discuss now the SDDP method applied to the SAA problem. The dynamic programming equations for the SAA problem can be written as

$$\widetilde{Q}_{tj}(x_{t-1}) = \inf_{x_t\in\mathbb{R}^{n_t}} \left\{ \tilde{c}_{tj}^{\mathsf T} x_t + \widetilde{\mathcal{Q}}_{t+1}(x_t) : \widetilde{B}_{tj} x_{t-1} + \widetilde{A}_{tj} x_t = \tilde{b}_{tj}, x_t \geqslant 0 \right\}, \quad j = 1,\dots,N_t \tag{3.5}$$

with

$$\widetilde{\mathcal{Q}}_{t+1}(x_t) = \frac{1}{N_{t+1}} \sum_{j=1}^{N_{t+1}} \widetilde{Q}_{t+1,j}(x_t), \tag{3.6}$$

$t = T,\dots,2$ and $\widetilde{\mathcal{Q}}_{T+1}(\cdot)\equiv 0$. The optimal value of the SAA problem is given by the optimal value of the first stage problem

$$\begin{aligned} \operatorname*{Min}_{x_1\in\mathbb{R}^{n_1}} \quad & c_1^{\mathsf T} x_1 + \widetilde{\mathcal{Q}}_2(x_1) \\ \text{s.t.} \quad & A_1 x_1 = b_1, x_1 \geqslant 0. \end{aligned} \tag{3.7}$$

The cost-to-go functions, of the true and SAA problems, are convex and since the number of scenarios of the SAA problem is finite, the cost-to-go functions $\widetilde{\mathcal{Q}}_t(\cdot)$ of the SAA problem are piecewise linear.

A *backward* step of the SDDP algorithm, applied to the SAA problem, can be described as follows. Let $\bar{x}_t \in \mathbb{R}^{n_t}$ be a trial decision at stage $t = 1,\dots,T-1$ and $\mathfrak{Q}_t(\cdot)$ be a current approximation of the cost-to-go function $\widetilde{\mathcal{Q}}_t(\cdot)$, given by the maximum of a collection of *cutting* planes, at stage $t = 2,\dots,T$. At stage $t = T$ we solve the problem

$$\begin{aligned} \operatorname*{Min}_{x_T\in\mathbb{R}^{n_T}} \quad & \tilde{c}_{Tj}^{\mathsf T} x_T \\ \text{s.t.} \quad & \widetilde{B}_{Tj} x_{T-1} + \widetilde{A}_{Tj} x_T = \tilde{b}_{Tj}, x_T \geqslant 0, \end{aligned} \tag{3.8}$$

for $x_{T-1} = \bar{x}_{T-1}$ and $j = 1,\dots,N_T$. Note that the optimal value of problem (3.8) is equal to $\widetilde{Q}_{Tj}(\bar{x}_{T-1})$.

Let $\tilde{x}_{Tj}$ be an optimal solution of problem (3.8) and $\tilde{\pi}_{Tj}$ be an optimal solution of its dual for $x_{T-1} = \bar{x}_{T-1}$ and $j = 1,\dots,N_T$. Then

$$\ell_T(x_{T-1}) := \widetilde{\mathcal{Q}}_T(\bar{x}_{T-1}) + \tilde{g}_T^{\mathsf T}(x_{T-1} - \bar{x}_{T-1}), \tag{3.9}$$

where

$$\widetilde{\mathcal{Q}}_T(\bar{x}_{T-1}) = \frac{1}{N_T}\sum_{j=1}^{N_T} \tilde{c}_{Tj}^{\mathsf T}\tilde{x}_{Tj} \quad \text{and} \quad \tilde{g}_T = -\frac{1}{N_T}\sum_{j=1}^{N_T} \widetilde{B}_{Tj}^{\mathsf T}\tilde{\pi}_{Tj} \tag{3.10}$$

is a supporting plane for $\widetilde{\mathcal{Q}}_T(\cdot)$ at $\bar{x}_{T-1}$. This supporting plane is added to the collection of supporting planes of $\mathfrak{Q}_T(\cdot)$, i.e., $\mathfrak{Q}_T(\cdot)$ is replaced by $\max\{\mathfrak{Q}_T(\cdot), \ell_T(\cdot)\}$.

Next the updated approximation $\mathfrak{Q}_T(\cdot)$ is used at stage $t = T-1$, where problem

$$\begin{aligned} \operatorname*{Min}_{x_{T-1}\in\mathbb{R}^{n_{T-1}}} \quad & \tilde{c}_{T-1,j}^{\mathsf T} x_{T-1} + \mathfrak{Q}_T(x_{T-1}) \\ \text{s.t.} \quad & \widetilde{B}_{T-1,j} x_{T-2} + \widetilde{A}_{T-1,j} x_{T-1} = \tilde{b}_{T-1,j}, x_{T-1} \geqslant 0, \end{aligned} \tag{3.11}$$

should be solved for $x_{T-2} = \bar{x}_{T-2}$ and $j = 1,\dots,N_{T-1}$. Since $\mathfrak{Q}_T(\cdot)$ is given by maximum of affine functions, problem (3.11) can be formulated as a linear programming problem with additional variables corresponding to each cutting plane used in construction of $\mathfrak{Q}_T(\cdot)$. Consider the optimal value, denoted $\underset{\sim}{Q}_{T-1,j}(x_{T-2})$, of problem (3.11), and let

[5] Unfortunately it is not clear how to extend these better algorithms to multistage programs.

$$\underset{\sim}{\mathcal{Q}}_{T-1}(x_{T-2}) := \frac{1}{N_{T-1}} \sum_{j=1}^{N_{T-1}} \underset{\sim}{Q}_{T-1,j}(x_{T-2}).$$

By construction we have that $\underset{\sim}{Q}_{T-1,j}(\cdot) \leqslant \widetilde{Q}_{T-1,j}(\cdot)$, $j = 1,\dots, N_{T-1}$, and hence $\underset{\sim}{\mathcal{Q}}_{T-1}(\cdot) \leqslant \widetilde{\mathcal{Q}}_{T-1}(\cdot)$.

By solving (3.11) and its dual for every $j = 1,\dots, N_{T-1}$, we can compute value $\underset{\sim}{\mathcal{Q}}_{T-1}(\bar{x}_{T-2})$ and a subgradient $\tilde{g}_{T-1}$ of $\underset{\sim}{\mathcal{Q}}_{T-1}(x_{T-2})$ at $x_{T-2} = \bar{x}_{T-2}$, and hence construct the supporting plane

$$\ell_{T-1}(x_{T-2}) := \underset{\sim}{\mathcal{Q}}_{T-1}(\bar{x}_{T-2}) + \tilde{g}_{T-1}^{\mathsf{T}}(x_{T-2} - \bar{x}_{T-2}), \tag{3.12}$$

for $\underset{\sim}{\mathcal{Q}}_{T-1}(x_{T-2})$ at $x_{T-2} = \bar{x}_{T-2}$. Then the approximation $\mathfrak{Q}_{T-1}(\cdot)$ is updated by replacing it with $\max\{\mathfrak{Q}_{T-1}(\cdot), \ell_{T-1}(\cdot)\}$. This process is continued backwards until at the first stage the following problem is solved

$$\begin{aligned} \underset{x_1\in\mathbb{R}^{n_1}}{\text{Min}} \quad & c_1^{\mathsf{T}}x_1 + \mathfrak{Q}_2(x_1) \\ \text{s.t.} \quad & A_1x_1 = b_1, x_1 \geqslant 0. \end{aligned} \tag{3.13}$$

Of course, the above backward step can be performed simultaneously for several values of trial decisions $\bar{x}_t$, $t = 1,\dots, T-1$. It could be also noted that starting from $t = T-1,\dots,$ values $\underset{\sim}{\mathcal{Q}}_t(x_{t-1})$ could be strictly smaller than $\widetilde{\mathcal{Q}}_t(x_{t-1})$ for some/all x_{t-1}, and the constructed planes are supporting planes for $\underset{\sim}{\mathcal{Q}}_t(\cdot)$ but could be only cutting planes for $\widetilde{\mathcal{Q}}_t(\cdot)$.

Similarly to the two-stage case we denote by $\underline{\vartheta}_k$ the optimal value of problem (3.13) at kth iteration of the backward step procedure. Since $\mathfrak{Q}_2(\cdot) \leqslant \widetilde{\mathcal{Q}}_2(\cdot)$ we have that $\underline{\vartheta}_k$ is less than or equal to the optimal value of the SAA problem. The value $\underline{\vartheta}_k$ depends on the random sample used in the construction of the SAA problem, and therefore is random. Recall that the expectation of the optimal value of the SAA problem, and hence $\mathbb{E}[\underline{\vartheta}_k]$, is smaller than the optimal value of the true problem. Therefore, on average, $\underline{\vartheta}_k$ gives a lower bound for the optimal value of the true problem.

The computed approximations $\mathfrak{Q}_2(\cdot),\dots,\mathfrak{Q}_T(\cdot)$ (with $\mathfrak{Q}_{T+1}(\cdot) \equiv 0$ by definition) and a feasible first stage solution[6] $\bar{x}_1$ can be used for constructing an implementable policy as follows. For a realization $\xi_2,\dots, \xi_T$ of the data process, decisions $\bar{x}_t$, $t = 1,\dots, T$, are computed recursively going forward with $\bar{x}_1$ being the chosen feasible solution of the first stage problem (3.13), and $\bar{x}_t$ being an optimal solution of

$$\begin{aligned} \underset{x_t\in\mathbb{R}^{n_t}}{\text{Min}} \quad & c_t^{\mathsf{T}}x_t + \mathfrak{Q}_{t+1}(x_t) \\ \text{s.t.} \quad & A_tx_t = b_t - B_t\bar{x}_{t-1}, \quad x_t \geqslant 0. \end{aligned} \tag{3.14}$$

for $t = 2,\dots, T$. These optimal solutions can be used as trial decisions in the backward step of the algorithm. Note that $\bar{x}_t$ is a function of $\bar{x}_{t-1}$ and $\xi_t = (c_t,A_t,B_t,b_t)$, i.e., $\bar{x}_t$ is a function of $\xi_{[t]} = (\xi_1,\dots,\xi_t)$, for $t = 2,\dots, T$. That is, $\bar{x}_t = \bar{x}_t(\xi_{[t]})$ is an implementable policy for the true problem (1.1). Moreover, by the construction this policy satisfies the feasibility constraints for any realization of the data process. If we restrict the data process to the generated sample, i.e., we consider only realizations $\xi_2,\dots, \xi_T$ of the data process given by scenarios of the SAA problem, then $\bar{x}_t = \bar{x}_t(\xi_{[t]})$ becomes an implementable and feasible policy for the corresponding SAA problem.

Since the policy $\bar{x}_t = \bar{x}_t(\xi_{[t]})$ is feasible, the expectation

$$\mathbb{E}\left[\sum_{t=1}^{T} c_t^{\mathsf{T}}\bar{x}_t(\xi_{[t]})\right] \tag{3.15}$$

gives an upper bound for the optimal value of the corresponding multistage problem. If we take this expectation over the true probability distribution of the random data process, then the above expectation (3.15) gives an upper bound for the optimal value of the true problem. On the other hand, if we restrict the data process to scenarios of the SAA problem, each with equal probability $1/N$, then the expectation (3.15) gives an upper bound for the optimal value of the SAA problem conditional on the sample used in construction of the SAA problem (compare with Remark 2 of the previous section).

The *forward* step of the SDDP algorithm consists in generating M random realizations (scenarios) of the data process and computing the respective optimal values $\vartheta_j := \sum_{t=1}^{T}\tilde{c}_{tj}^{\mathsf{T}}\bar{x}_{tj}$, $j = 1,\dots, M$. Consequently the sample average $\bar{\vartheta}$ and sample variance $\hat{\sigma}_{\vartheta}^2$ are calculated, by using the same formulas as in (2.10), and the respective confidence interval (2.11) is constructed. Similarly to the case of two-stage programming, discussed in Section 2, the random sample can be constructed as a subsample of the sample used in the SAA problem or can be generated directly from the original (true) distribution of the data process. In both cases the average $\bar{\vartheta}$ is an unbiased estimator of the corresponding expectation (3.15), and $\mathbb{E}[\bar{\vartheta}]$ is an upper bound for the optimal value of the true problem. In the case of subsampling, $\bar{\vartheta} + z_\alpha\hat{\sigma}_\vartheta/\sqrt{M}$ gives an upper bound for the optimal value of the SAA problem with confidence of about $1-\alpha$. Similarly in the case of direct sampling from the true distribution, $\bar{\vartheta} + z_\alpha\hat{\sigma}_\vartheta/\sqrt{M}$ gives an upper bound for the optimal value of the true problem with confidence of about $1-\alpha$.

Remark 4. As it was discussed in Section 2 (see Remark 1) the stopping criterion of ending the computational procedure the first time the lower bound $\underline{\vartheta}_k$ becomes bigger than the lower end $\bar{\vartheta} - z_{\alpha/2}\hat{\sigma}_\vartheta/\sqrt{M}$ of the confidence interval could be quite misleading. The larger the variance $\hat{\sigma}_{\vartheta}^2$ and the confidence $1-\alpha$ are, the sooner the procedure will be stopped by this criterion. Note that $\hat{\sigma}_{\vartheta}^2$ is an unbiased estimator of the variance of $\sum_{t=2}^{T}c_t^{\mathsf{T}}\bar{x}_t$, and this variance could be expected to grow with increase of the number of stages T more or less proportionally to T if the terms $c_t^{\mathsf{T}}\bar{x}_t$ of that sum have more or less the same variances and nearly uncorrelated with each other. It makes more sense to consider the difference between the upper bound $\bar{\vartheta} + z_\alpha\hat{\sigma}_\vartheta/\sqrt{M}$ and lower bound $\underline{\vartheta}_k$. This difference gives an approximate upper bound for the gap between the value of the current policy and the optimal value of the SAA problem. For a further discussion of this issue see Section 5.

Remark 5. The upper bound $ub := \bar{\vartheta} + z_\alpha\hat{\sigma}_\vartheta/\sqrt{M}$ depends on sampled scenarios and hence is random. The meaning of this upper bound is that for *one* run of the forward step procedure the upper bound ub is bigger than the expected value (3.15) of the considered policy with probability $1-\alpha$. Suppose that we run the forward step procedure several, say k, times for the *same* policy, and hence compute respective upper bounds $ub_1,\dots, ub_k$. Consider the minimum $\underline{ub}_k$:=min$\{ub_1,\dots,ub_k\}$ of these upper bounds. This could be motivated by desire to pick a "best" of computed upper bounds. The probability that $\underline{ub}_k$ is an upper bound for the expected value of the considered policy is $\Pr(\cap_{i=1}^{k}A_i)$, where A_i is the event that ub_i is bigger than the expected value of the considered policy. By the Bonferroni inequality we have that $\Pr(\cap_{i=1}^{k}A_i) \geqslant 1-k\alpha$. That is, if we want $\underline{ub}_k$ to give the required upper bound with confidence $1-\alpha$, to be on safe side we need to employ the individual upper bounds at the confidence level $1-\alpha/k$, i.e., to use $\bar{\vartheta} + z_{\alpha/k}\hat{\sigma}_\vartheta/\sqrt{M}$. If the samples used in calculations of the upper bounds are independent of each other, and hence the events A_i are independent, then $\Pr(\cap_{i=1}^{k}A_i) = (1-\alpha)^k$. For small $\alpha > 0$ and not too large k we have that $(1-\alpha)^k \approx 1-k\alpha$, and hence the choice of confidence level $1-\alpha/k$ for the individual upper bounds is not very conservative. It could be mentioned that the critical values $z_{\alpha/k}$, given by the respective quantiles of the standard normal distribution, are justified by an application of the Central Limit Theorem. It also should

[6] Note that by the construction the first stage solution computed in a backward step is feasible, i.e., satisfies the constraints $A_1x_1 = b_1$, $x_1 \geqslant 0$.

be remembered that for very small probabilities α/k, i.e., for large k, approximation of the "true" quantiles by using normal distribution could be poor (this is why the Central Limit Theorem is called "central").

Remark 6. It is also possible to compare values of two policies, say associated with two different sets of the lower approximations $\mathfrak{Q}_2(\cdot),\dots,\mathfrak{Q}_T(\cdot)$. That is, let $\bar{x}_t = \bar{x}_t(\xi_{[t]})$ and $\hat{x}_t = \hat{x}_t(\xi_{[t]})$, $t = 1,\dots, T$, be two implementable and feasible policies. Generate M random realizations $\xi_2^j,\dots,\xi_T^j$, $j = 1,\dots, M$, of the data process, and for $(\xi_2,\dots,\xi_T) = (\xi_2^j,\dots,\xi_T^j)$ compute

$$Z_j := \sum_{t=1}^{T} c_t^{\mathsf{T}} \bar{x}_t(\xi_{[t]}) - \sum_{t=1}^{T} c_t^{\mathsf{T}} \hat{x}_t(\xi_{[t]}) \quad j = 1,\dots,M. \tag{3.16}$$

Note that $\mathbb{E}[Z_j]$ is equal to the difference between expected values of these two policies. Therefore we can apply the standard t-test to $Z_1,\dots, Z_M$ to construct a confidence interval for the difference between these expected values and hence to test[7] whether this difference is bigger/smaller than zero.

3.1. Convergence properties of the SDDP method

In this section we discuss convergence properties of the SDDP algorithm applied to the SAA problem governed by dynamic programming Eqs. (3.5)–(3.7). Consider optimization problems (3.14) used in the forward step procedure with approximations $\mathfrak{Q}_{t+1}(\cdot)$ constructed at kth iteration of the backward step procedure (recall that $\mathfrak{Q}_{T+1}(\cdot) \equiv 0$). We make the following assumption.

(A1) Problem (3.13) and problems (3.14), $t = 2,\dots, T$, have finite optimal values for all realizations (scenarios) of the data used in the SAA problem.

Since the functions $\mathfrak{Q}_{t+1}(\cdot)$ are convex piecewise linear it follows that if a problem of the form (3.13) or (3.14), $t = 2,\dots, T$, has finite optimal value, then it has an optimal solution.

Now let us consider cutting planes

$$\ell_t(x_{t-1}) := \underline{\widetilde{\mathcal{Q}}}_t(\bar{x}_{t-1}) + \tilde{g}_t^{\mathsf{T}}(x_{t-1} - \bar{x}_{t-1}), \tag{3.17}$$

which were constructed in the backward step procedure. The required subgradient $\tilde{g}_t \in \partial \underline{\widetilde{\mathcal{Q}}}_t(\bar{x}_{t-1})$ is computed by solving the corresponding dual problems, for $j = 1,\dots, N_t$, and averaging their optimal solutions. These dual problems can be written as linear programming problems and because of assumption (A1) have optimal solutions. If we compute basic feasible solutions of these dual problems, then we say that *basic optimal solutions* are used. In that case the number of such optimal solutions is finite, and hence the number of cutting planes of the form (3.17) is finite (cf., [14, section 2.1]).

Proposition 3.1. *Suppose that in the forward steps of the SDDP algorithm the subsampling procedure is used, assumption* (A1) *holds and in the backward steps the basic optimal solutions are employed. Then w.p.1 after a sufficiently large number of backward and forward steps of the algorithm, the forward step procedure defines an optimal policy for the SAA problem.*

[7] The above is a *paired* t-test since the same sample was used in two terms in the right hand side of (3.16). It seems that it would be advantageous to use here the paired t-test since these two terms are expected to be positively correlated.

Proof. Let us first observe that under the specified assumptions the total number of possible different realizations of functions $\mathfrak{Q}_t(\cdot)$, $t = 2,\dots, T$, and first stage solutions generated by the backward step procedure, is finite. Indeed, let us look at the construction of $\mathfrak{Q}_T(\cdot)$ at stage $t = T$, see Eqs. (3.8)–(3.10). As it was discussed above, since basic optimal solutions are used in the construction of supporting planes of $\widetilde{\mathcal{Q}}_T$ we have that there is only a finite number of different functions $\mathfrak{Q}_T(\cdot)$ that can be constructed by this procedure. Next for each possible realization of $\mathfrak{Q}_T(\cdot)$ there is only a finite number of different functions $\mathfrak{Q}_{T-1}(\cdot)$ that can be constructed by (3.11) and (3.12), and so on.

Recall that a policy $\bar{x}_t = \bar{x}_t(\tilde{\xi}_{[t]})$, $t = 1,\dots, T$, is optimal for the SAA problem if the following (dynamic programming) optimality conditions

$$\bar{x}_t(\tilde{\xi}_{[t]}) \in \arg\min_{x_t \in \mathbb{R}^{n_t}} \left\{ \tilde{c}_t^{\mathsf{T}} x_t + \widetilde{\mathcal{Q}}_{t+1}(x_t) : \widetilde{B}_t \bar{x}_{t-1}(\tilde{\xi}_{[t-1]}) + \widetilde{A}_t x_t = \tilde{b}_t, x_t \geqslant 0 \right\}, \tag{3.18}$$

hold for $t = 1,\dots, T$, and all realizations $\tilde{\xi}_2,\dots,\tilde{\xi}_T$ of the data process of the SAA problem (recall that $\widetilde{B}_0 = 0$ and $\widetilde{\mathcal{Q}}_{T+1}(\cdot) \equiv 0$). Now let $\mathfrak{Q}_t(\cdot)$, $t = 2,\dots, T$, be a current set of constructed approximation functions, $\bar{x}_1$ be the first stage solution generated by the backward step procedure and $\bar{x}_t = \bar{x}_t(\tilde{\xi}_{[t]})$, $t = 2,\dots, T$, be the corresponding forward step policy. Suppose that the forward step policy is not optimal for the SAA problem. That is, the optimality conditions (3.18) do not hold for some $t' \in \{2,\dots,T\}$ and some realization $\tilde{\xi}'_2,\dots,\tilde{\xi}'_T$ of the data process. Let t' be the largest such t, i.e., consider the first time going backwards that the optimality conditions (3.18) do not hold. Then for the trial decision $\bar{x}_{t'} = \bar{x}_{t'}(\tilde{\xi}'_{[t']})$, by adding the corresponding cutting plane $\ell_{t'}(\cdot)$ we change (increase) the current function $\mathfrak{Q}_{t'}(\cdot)$. Since the total number of scenarios of the SAA problem is finite, we have that w.p.1 the realization $\tilde{\xi}'_2,\dots,\tilde{\xi}'_T$ will happen in the forward step procedure for a sufficiently large number of iterations. Since the total number of different realizations of functions $\mathfrak{Q}_t(\cdot)$ is finite, we obtain that w.p.1 after a sufficiently large number of iterations the optimality conditions (3.18) will hold for all t and all realizations of the data process of the SAA problem. $\square$

Some remarks about this proof are now in order. The main argument of the above proof of finite convergence of the SDDP algorithm is based on finiteness of all possible realizations of the lower approximation functions. This, in turn, is based on that the total number of scenarios of the SAA problem is finite and basic optimal solutions of the dual problems are used in the backward steps. Another argument of the proof is that the forward step procedure will generate w.p.1 every possible scenario of the considered SAA problem for sufficiently large number of runs. This, in turn, is based on the assumption that the sampled scenarios of the forward step procedure are generated from the scenarios of the SAA problem and, more importantly, are generated independently of each other. Without this condition of independence of scenarios generated in the forward steps, there is no guarantee of convergence even for relatively small problems.

4. Risk averse approach

Let us look again at the two-stage problem (2.1) and (2.2). At the first stage the value $Q(x,\xi)$ is minimized *on average*. Of course, for a particular realization of the random vector ξ the second stage cost $Q(x,\xi)$ can be quite bigger than its mean (expected value) $\mathcal{Q}(x)$. Therefore, in order to control this cost one can add the constraint $Q(x,\xi) \leqslant \eta$ for some chosen constant η and trying to enforce it for *all* realizations of the random vector ξ. However, enforcing such constraint for *all* realizations of ξ could be infeasible for any reasonable value of the parameter η.

A more realistic approach would be to enforce this constraint with a certain confidence level, i.e., to write it in the following form of a chance (probabilistic) constraint

$$\Pr\{Q(x,\xi) \leqslant \eta\} \geqslant 1-\alpha, \tag{4.1}$$

where $\alpha \in (0,1)$ is a small number (significance level). Equivalently the above constraint can be written as

$$\mathsf{V@R}_\alpha[Q(x,\xi)] \leqslant \eta, \tag{4.2}$$

where Value-at-Risk, $\mathsf{V@R}_\alpha[Z]$, of random variable Z is defined as

$$\mathsf{V@R}_\alpha[Z] := \inf\{u : \Pr(Z \leqslant u) \geqslant 1-\alpha\},$$

i.e., $\mathsf{V@R}_\alpha[Z]$ is the left-side $(1-\alpha)$-quantile of the distribution of Z. This leads to the following risk averse formulation of the two-stage program

$$\begin{array}{ll} \underset{x\in\mathcal{X}}{\text{Min}} & c^{\mathsf T}x + \mathcal{Q}(x) \\ \text{s.t.} & \mathsf{V@R}_\alpha[Q(x,\xi)] \leqslant \eta. \end{array} \tag{4.3}$$

The difficulty with the above risk averse formulation (4.3) is that the function $\mathsf{V@R}_\alpha[Q(x,\xi)]$ typically is not convex in x, even if $Q(\cdot,\xi)$ is convex, and is difficult to handle numerically. It was suggested in [13] to replace chance constraint (4.2) by the following conservative approximation

$$\mathsf{CV@R}_\alpha[Q(x,\xi)] \leqslant \eta, \tag{4.4}$$

where the Conditional Value-at-Risk of a random variable Z is defined as

$$\mathsf{CV@R}_\alpha[Z] := \inf_{u\in\mathbb{R}}\left\{u + \alpha^{-1}\mathbb{E}[Z-u]_+\right\}. \tag{4.5}$$

The minimum in the right hand side of the above definition is attained at $u^* = \mathsf{V@R}_\alpha[Z]$, and hence

$$\mathsf{CV@R}_\alpha[Z] = \mathsf{V@R}_\alpha[Z] + \alpha^{-1}\mathbb{E}[Z - \mathsf{V@R}_\alpha(Z)]_+.$$

That is, $\mathsf{CV@R}_\alpha[Z]$ is bigger than $\mathsf{V@R}_\alpha[Z]$ by the nonnegative amount $\alpha^{-1}\mathbb{E}[Z-\mathsf{V@R}_\alpha(Z)]_+$. It follows that (4.4) is indeed a conservative approximation of (4.2) in the sense that the constraint (4.4) is stricter than (4.2). Note also that convexity of $Q(\cdot,\xi)$ is preserved in the function $\mathsf{CV@R}_\alpha[Q(\cdot,\xi)]$ (cf., [16, Lemma 3.1]).

Replacing chance constraint (4.2) with the constraint (4.4) leads to the following risk averse formulation

$$\begin{array}{ll} \underset{x\in\mathcal{X}}{\text{Min}} & c^{\mathsf T}x + \mathbb{E}[Q(x,\xi)] \\ \text{s.t.} & \mathsf{CV@R}_\alpha[Q(x,\xi)] \leqslant \eta. \end{array} \tag{4.6}$$

Note again that the feasible set of problem (4.6) is contained in the feasible set of problem (4.3). It could be convenient to move the constraint (4.4) into the objective function. That is, consider

$$\rho_\lambda(Z) := (1-\lambda)\mathbb{E}[Z] + \lambda\mathsf{CV@R}_\alpha[Z],$$

viewed as a real valued function defined on the space $L_1(\Omega,\mathcal{F},P)$ of integrable random variables Z. Consider further the problem

$$\underset{x\in\mathcal{X}}{\text{Min}}\, c^{\mathsf T}x + \rho_\lambda[Q(x,\xi)]. \tag{4.7}$$

The parameter $\lambda \in [0,1]$ can be tuned for a compromise between optimizing on average and risk control. The function $\rho_\lambda(\cdot)$ is an example of the so-called coherent risk measures (cf., [1]). An advantage of formulation (4.7) over (4.6) is that the constraint (4.4) can make the problem (4.6) infeasible, even in the case of relatively complete recourse, while problem (4.7) does not have this drawback. On the other hand the parameter η has a more intuitive explanation in formulation (4.6) than parameter λ in formulation (4.7).

By using definition (4.5) of $\mathsf{CV@R}_\alpha$ we can write problem (4.7) as

$$\underset{x\in\mathcal{X},u\in\mathbb{R}}{\text{Min}}\, c^{\mathsf T}x + \lambda u + \mathbb{E}\left\{(1-\lambda)Q(x,\xi) + \lambda\alpha^{-1}[Q(x,\xi)-u]_+\right\}. \tag{4.8}$$

Consequently we can write problem (4.7) as the following two-stage problem (cf., [21, pp. 294–295]):

$$\underset{x\in\mathcal{X},u\in\mathbb{R}}{\text{Min}}\, c^{\mathsf T}x + \lambda u + \mathbb{E}[V(x,u,\xi)], \tag{4.9}$$

where $V(x,u,\xi)$ is the optimal value of the second stage problem

$$\begin{array}{ll} \underset{y\in\mathbb{R}^{n_2}}{\text{Min}} & (1-\lambda)q^{\mathsf T}y + \lambda\alpha^{-1}[q^{\mathsf T}y-u]_+ \\ \text{s.t.} & Tx + Wy = h, y \geqslant 0. \end{array} \tag{4.10}$$

Note that problem (4.10) can be written as the linear program:

$$\begin{array}{ll} \underset{y\in\mathbb{R}^{n_2},v\in\mathbb{R}}{\text{Min}} & (1-\lambda)q^{\mathsf T}y + \lambda\alpha^{-1}v \\ \text{s.t.} & q^{\mathsf T}y - u \leqslant v,\ v \geqslant 0, \\ & Tx + Wy = h, y \geqslant 0. \end{array} \tag{4.11}$$

Problem (4.9) can be viewed as a linear two-stage stochastic program with (first stage) decision variables $x \in \mathbb{R}^{n_1}$ and $u \in \mathbb{R}$, and second stage problem (4.11) with (second stage) decision variables $y \in \mathbb{R}^{n_2}$ and $v \in \mathbb{R}$. Consequently the methodology discussed in Section 2 can be applied in a straightforward way.

The above approach can be extended to a multistage setting. That is, an extension of the two-stage risk averse formulation (4.7) to the multistage setting[8] is obtained by replacing in Eq. (3.2) the expectation operators with the (conditional) risk measures

$$\rho_{t|\xi_{[t-1]}}[Z] := (1-\lambda_t)\mathbb{E}\left[Z|\xi_{[t-1]}\right] + \lambda_t\mathsf{CV@R}_{\alpha_t}\left[Z|\xi_{[t-1]}\right], \tag{4.12}$$

with $\lambda_t \in [0,1]$ and $\alpha_t \in (0,1)$ being chosen parameters (see [21, section 6.7.3]). The corresponding dynamic programming equations for $t = T,\ldots,2$ take the form

$$Q_t(x_{t-1},\xi_t) = \inf_{x_t\in\mathbb{R}^{n_t}}\left\{c_t^{\mathsf T}x_t + \mathcal{Q}_{t+1}(x_t) : B_tx_{t-1} + A_tx_t = b_t, x_t \geqslant 0\right\}, \tag{4.13}$$

where

$$\mathcal{Q}_{t+1}(x_t) := \rho_{t+1|\xi_{[t]}}[Q_{t+1}(x_t,\xi_{t+1})] \tag{4.14}$$

with $\mathcal{Q}_{T+1}(\cdot) \equiv 0$. At the first stage problem

$$\begin{array}{ll} \underset{x_1\in\mathbb{R}^{n_1}}{\text{Min}} & c_1^{\mathsf T}x_1 + \mathcal{Q}_2(x_1) \\ \text{s.t.} & A_1x_1 = b_1, x_1 \geqslant 0, \end{array} \tag{4.15}$$

should be solved. Note that because of the stagewise independence condition, the cost-to-go functions $\mathcal{Q}_{t+1}(x_t)$ do not depend on the random data process.

These dynamic programming equations correspond to the following nested formulation of the risk averse problem (cf., [17])

$$\underset{\substack{A_1x_1=b_1\\x_1\geqslant 0}}{\text{Min}}\, c_1^{\mathsf T}x_1 + \rho_{2|\xi_1}\left[\underset{\substack{B_2x_1+A_2x_2=b_2\\x_2\geqslant 0}}{\min}\, c_2^{\mathsf T}x_2 + \cdots + \rho_{T|\xi_{[T-1]}}\left[\underset{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\x_T\geqslant 0}}{\min}\, c_T^{\mathsf T}x_T\right]\right]. \tag{4.16}$$

An intuitive explanation of this approach is that at tth stage of the process one tries to control an upper limit of the corresponding cost-to-go function $Q_{t+1}(x_t,\xi_{t+1})$ for different realizations of the data process. By using definition (4.5) of Conditional Value-at-Risk we can write formula (4.14) in the form

$$\mathcal{Q}_{t+1}(x_t) = \inf_{u_t}\mathbb{E}\left\{(1-\lambda_{t+1})Q_{t+1}(x_t,\xi_{t+1}) + \lambda_{t+1}\left(u_t + \alpha_{t+1}^{-1}[Q_{t+1}(x_t,\xi_{t+1}) - u_t]_+\right)\right\}. \tag{4.17}$$

As it was mentioned earlier the minimum in the right hand side of (4.17) is attained at

[8] It is also possible to extend formulation (4.6) to a multistage setting in a similar way.

$u_t^* = \mathsf{V@R}_{\alpha_{t+1}}[Q_{t+1}(x_t, \xi_{t+1})]$.

A balance between optimizing on average and risk aversion can be controlled by tuning the parameters λ_t and α_t at different stages of the process. It also could be noted that the above formulation (4.16) satisfies a condition of *time consistency* (cf., [15,20]).

By using formula (4.17) we can proceed in writing dynamic programming Eqs. (4.13) and (4.14) as follows. At last stage $t = T$ we have that $Q_T(x_{T-1}, \xi_T)$ is equal to the optimal value of problem

$$\begin{aligned} &\underset{x_T \in \mathbb{R}^{N_T}}{\text{Min}} \quad c_T^{\mathsf T} x_T \\ &\text{s.t.} \quad B_T x_{T-1} + A_T x_T = b_T, \quad x_T \geqslant 0 \end{aligned} \tag{4.18}$$

and

$$\mathcal{Q}_T(x_{T-1}) = \inf_{u_{T-1}} \mathbb{E}\{(1-\lambda_T) Q_T(x_{T-1}, \xi_T) + \lambda_T u_{T-1} + \lambda_T \alpha_T^{-1}[Q_T(x_{T-1}, \xi_T) - u_{T-1}]_+\}. \tag{4.19}$$

At stage $t = T - 1$ we have that $Q_{T-1}(x_{T-2}, \xi_{T-1})$ is equal to the optimal value of problem

$$\begin{aligned} &\underset{x_{T-1} \in \mathbb{R}^{N_{T-1}}}{\text{Min}} \quad c_{T-1}^{\mathsf T} x_{T-1} + \mathcal{Q}_T(x_{T-1}) \\ &\text{s.t.} \quad B_{T-1} x_{T-2} + A_{T-1} x_{T-1} = b_{T-1}, \quad x_{T-1} \geqslant 0. \end{aligned} \tag{4.20}$$

By using (4.19) and (4.20) we can write that $Q_{T-1}(x_{T-2}, \xi_{T-1})$ is equal to the optimal value of problem

$$\begin{aligned} &\underset{x_{T-1} \in \mathbb{R}^{N_{T-1}}, u_{T-1} \in \mathbb{R}}{\text{Min}} \quad c_{T-1}^{\mathsf T} x_{T-1} + \lambda_T u_{T-1} + \mathcal{Q}_T(x_{T-1}, u_{T-1}) \\ &\text{s.t.} \quad B_{T-1} x_{T-2} + A_{T-1} x_{T-1} = b_{T-1}, \quad x_{T-1} \geqslant 0, \end{aligned} \tag{4.21}$$

where

$$\mathcal{Q}_T(x_{T-1}, u_{T-1}) := \mathbb{E}\{(1-\lambda_T) Q_T(x_{T-1}, \xi_T) + \lambda_T \alpha_T^{-1}[Q_T(x_{T-1}, \xi_T) - u_{T-1}]_+\}. \tag{4.22}$$

By continuing this process backward we can write dynamic programming Eqs. (4.13) and (4.14) for $t = T, \ldots, 2$ as

$$Q_t(x_{t-1}, \xi_t) = \inf_{x_t \in \mathbb{R}^{n_t}, u_t \in \mathbb{R}} \{c_t^{\mathsf T} x_t + \lambda_{t+1} u_t + \mathcal{Q}_{t+1}(x_t, u_t) : B_t x_{t-1} + A_t x_t = b_t, x_t \geqslant 0\}, \tag{4.23}$$

where

$$\mathcal{Q}_{t+1}(x_t, u_t) = \mathbb{E}\{(1-\lambda_{t+1}) Q_{t+1}(x_t, \xi_{t+1}) + \lambda_{t+1} \alpha_{t+1}^{-1}[Q_{t+1}(x_t, \xi_{t+1}) - u_t]_+\}, \tag{4.24}$$

with $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and $\lambda_{T+1} := 0$. At the first stage problem

$$\begin{aligned} &\underset{x_1 \in \mathbb{R}^{n_1}, u_1 \in \mathbb{R}}{\text{Min}} \quad c_1^{\mathsf T} x_1 + \lambda_2 u_1 + \mathcal{Q}_2(x_1, u_1) \\ &\text{s.t.} \quad A_1 x_1 = b_1, \quad x_1 \geqslant 0, \end{aligned} \tag{4.25}$$

should be solved. Note that in this formulation decision variables at tth stage are $x_t \in \mathbb{R}^{n_t}$ and $u_t \in \mathbb{R}$, and with some abuse of notation we denote the cost-to-go functions in (4.24) by $\mathcal{Q}_{t+1}(x_t, u_t)$ while these functions are different from the cost-to-go functions $\mathcal{Q}_{t+1}(x_t)$ defined in (4.17). Note also that functions $\mathcal{Q}_{t+1}(x_t, u_t)$ are convex.

Now consider the corresponding SAA problem based on random samples (3.4). For the SAA problem dynamic programming Eqs. (4.23) and (4.24) can be written, going backwards in time, as follows. Starting with $\widetilde{\mathcal{Q}}_{T+1}(\cdot,\cdot) \equiv 0$, for $t = T, \ldots, 2$, compute

$$\widetilde{Q}_{tj}(x_{t-1}) = \inf_{x_t \in \mathbb{R}^{n_t}, u_t \in \mathbb{R}} \left\{\tilde{c}_{tj}^{\mathsf T} x_t + \lambda_{t+1} u_t + \widetilde{\mathcal{Q}}_{t+1}(x_t, u_t) : \widetilde{B}_{tj} x_{t-1} + \widetilde{A}_{tj} x_t = \tilde{b}_{tj}, x_t \geqslant 0\right\}, \tag{4.26}$$

for $j = 1, \ldots, N_t$, and set

$$\widetilde{\mathcal{Q}}_t(x_{t-1}, u_{t-1}) = \frac{1}{N_t} \sum_{j=1}^{N_t} \left\{(1-\lambda_t) \widetilde{Q}_{tj}(x_{t-1}) + \lambda_t \alpha_t^{-1} \left[\widetilde{Q}_{tj}(x_{t-1}) - u_{t-1}\right]_+\right\}. \tag{4.27}$$

At the first stage solve problem of the form (4.25) with $\mathcal{Q}_2(\cdot,\cdot)$ replaced by $\widetilde{\mathcal{Q}}_2(\cdot,\cdot)$.

By the chain rule for subdifferentials we have that the subdifferential of the (convex) function $\phi(x_{t-1}, u_{t-1}) := [\widetilde{Q}_{tj}(x_{t-1}) - u_{t-1}]_+$ at a point $(x_{t-1}, u_{t-1}) = (\bar{x}_{t-1}, \bar{u}_{t-1})$ can be written as

$$\partial\phi(\bar{x}_{t-1}, \bar{u}_{t-1}) = \begin{cases} [0,0] & \text{if} \quad \widetilde{Q}_{tj}(\bar{x}_{t-1}) < \bar{u}_{t-1}, \\ \bigcup\limits_{g \in \partial \widetilde{Q}_{tj}(\bar{x}_{t-1})} [g, -1] & \text{if} \quad \widetilde{Q}_{tj}(\bar{x}_{t-1}) > \bar{u}_{t-1}, \\ \bigcup\limits_{\substack{g \in \partial \widetilde{Q}_{tj}(\bar{x}_{t-1}) \\ t \in [0,1]}} [tg, -t] & \text{if} \quad \widetilde{Q}_{tj}(\bar{x}_{t-1}) = \bar{u}_{t-1}. \end{cases} \tag{4.28}$$

Consequently, if $\tilde{g}_{tj} \in \partial \widetilde{Q}_{tj}(\bar{x}_{t-1})$, $j = 1, \ldots, N_t$, then a subgradient of $\widetilde{\mathcal{Q}}_t(x_{t-1}, u_{t-1})$ at $(\bar{x}_{t-1}, \bar{u}_{t-1})$ is given by

$$\frac{1}{N_t}\left[(1-\lambda_t) \sum_{j=1}^{N_t} \tilde{g}_{tj} + \lambda_t \alpha_t^{-1} \sum_{j \in \mathcal{J}_t} \tilde{g}_{tj}, -\lambda_t \alpha_t^{-1} |\mathcal{J}_t|\right] \tag{4.29}$$

where

$$\mathcal{J}_t := \left\{j : \widetilde{Q}_{tj}(\bar{x}_{t-1}) > \bar{u}_{t-1}, \quad j = 1, \ldots, N_t\right\}.$$

One can proceed now in backward steps of the SDDP algorithm in a way similar to the risk neutral case by adding cutting planes of the cost-to-go functions $\widetilde{\mathcal{Q}}_t(x_{t-1}, u_{t-1})$.

Let us consider now construction of the corresponding forward step procedure. Given a feasible first stage solution $(\bar{x}_1, \bar{u}_1)$ and a current set of piecewise linear lower approximations $\mathfrak{Q}_t(x_{t-1}, u_{t-1})$ of cost-to-go functions $\widetilde{\mathcal{Q}}_t(x_{t-1}, u_{t-1})$, $t = 2, \ldots, T$, we can proceed iteratively forward by solving problems

$$\begin{aligned} &\underset{x_t \in \mathbb{R}^{n_t}, u_t \in \mathbb{R}}{\text{Min}} \quad c_t^{\mathsf T} x_t + \lambda_{t+1} u_t + \mathfrak{Q}_{t+1}(x_t, u_t) \\ &\text{s.t.} \quad A_t x_t = b_t - B_t x_{t-1}, \quad x_t \geqslant 0, \end{aligned} \tag{4.30}$$

for a (randomly) generated scenario $\xi_2, \ldots, \xi_T$. Let $(\bar{x}_t, \bar{u}_t)$, $t = 1, \ldots, T$, be respective optimal solutions. These solutions can be used in constructions of cutting planes in the backward step procedure.

We have that $\bar{x}_t = \bar{x}_t(\xi_{[t]})$ and $\bar{u}_t = \bar{u}_t(\xi_{[t]})$, $t = 1, \ldots, T$, are functions of the data process, and $\bar{x}_t(\xi_{[t]})$ gives a feasible and implementable policy. Value of this policy can be computed by using nested formulation (4.16). At the last stage $t = T$ we need to compute value $\rho_{T|\xi_{[T-1]}}[c_T^{\mathsf T} \bar{x}_T]$. This value is given by

$$\mathfrak{v}_T := \mathbb{E}\{(1-\lambda_T) c_T^{\mathsf T} \bar{x}_T + \lambda_T \bar{u}_T + \lambda_T \alpha_T^{-1}[c_T^{\mathsf T} \bar{x}_T - \bar{u}_T]_+ | \xi_{[T-1]}\}.$$

Note that the (conditional) Value-at-Risk of $c_T^{\mathsf T} \bar{x}_T(\xi_{[T]})$ is replaced here by its estimate $\bar{u}_t$ given by the considered policy, and $\mathfrak{v}_T = \mathfrak{v}_T(\xi_{[T-1]})$ is a function $\xi_{[T-1]}$. At stage $t = T - 1$ we need to compute $\rho_{T-1|\xi_{[T-2]}}[c_{T-1}^{\mathsf T} \bar{x}_{T-1} + \mathfrak{v}_T]$, which is given by

$$\mathfrak{v}_{T-1} := \mathbb{E}\{(1-\lambda_{T-1})(c_{T-1}^{\mathsf T} \bar{x}_{T-1} + \mathfrak{v}_T) + \lambda_{T-1} \bar{u}_{T-1} + \lambda_{T-1} \alpha_{T-1}^{-1}[c_{T-1}^{\mathsf T} \bar{x}_{T-1} + \mathfrak{v}_T - \bar{u}_T]_+ | \xi_{[T-2]}\}.$$

Note that $\mathfrak{v}_{T-1} = \mathfrak{v}_{T-1}(\xi_{[T-2]})$ is a function of $\xi_{[T-2]}$, and so on going backwards in time.

In order to estimate value of this policy by sampling, and hence to construct a respective upper bound, one would need to employ a conditional sampling. To simplify the presentation let us assume for the moment that $T = 3$. Generate independent samples ξ_{ti}, $i = 1, \ldots, M_t$, from respective random vectors ξ_t, $t = 2,3$. For every ξ_{2i}, $i = 1, \ldots, M_2$, we can estimate $\mathfrak{v}_3(\xi_{2i})$ by

$$\hat{\mathfrak{v}}_{3i} = \frac{1}{M_3} \sum_{j=1}^{M_3} \left\{(1-\lambda_3) c_{3j}^{\mathsf T} \bar{x}_{3j} + \lambda_3 \bar{u}_{3j} + \lambda_3 \alpha_3^{-1}[c_3^{\mathsf T} \bar{x}_{3j} - \bar{u}_{3j}]_+\right\},$$

where the optimal values $(\bar{x}_{3j}, \bar{u}_{3j})$ correspond to scenarios (ξ_{2i}, ξ_{3j}), $j = 1, \ldots, M_3$. Note that $\hat{\mathfrak{v}}_{3i}$ is a function of ξ_{2i}, $i = 1, \ldots, M_2$. Consequently $\mathfrak{v}_2$ is estimated by

$$\hat{\mathfrak{v}}_2 = \frac{1}{M_2} \times \sum_{i=1}^{M_2} \left\{ (1-\lambda_2)(c_{2i}^{\mathsf{T}} \bar{x}_{2i} + \hat{\mathfrak{v}}_{3i}) + \lambda_2 \bar{u}_{2i} + \lambda_2 \alpha_2^{-1} [c_2^{\mathsf{T}} \bar{x}_{2i} + \hat{\mathfrak{v}}_{3i} - \bar{u}_{2i}]_+ \right\}.$$

Finally, the policy value is estimated by $c_1^{\mathsf{T}} \bar{x}_1 + \hat{\mathfrak{v}}_2$.

For T-stage programs this estimation procedure requires solving $M = \prod_{t=2}^{T} M_t$ linear programs and for a large number of stages could be impractical. Anyway for a large number of stages the considered sample based upper bounds typically are too loose and practically are not very useful. We will discuss this further at the end of next section.

5. Discussion

One run of the backward step procedure requires solving $1 + N_2 + \cdots + N_T$ linear programming problems. Each of these problems has a fixed number of decision variables and constraints with additional variables and constraints corresponding to cutting planes of the approximate functions $\mathfrak{Q}_t(\cdot)$. That is, complexity of one run of the backward step procedure is more or less proportional to the sum of the sample sizes, while the total number of scenarios is given by the product of the sample sizes. Therefore for a not too large number $\sum_{t=2}^{T} N_t$ one can run a reasonable number of backward steps of the algorithm, while the total number of scenarios $N = \prod_{t=2}^{T} N_t$ could be astronomically large. Similarly, one run of the forward step procedure involves solving $T-1$ linear programming problems and could be run for a reasonably large number of repetitions M.

The proof of finite convergence of the SDDP algorithm, given in Section 3.1, is based on that the SAA problem has a finite number of scenarios. The finiteness arguments used in that proof are typical for proofs of that type (cf., [12]). Although correct from a mathematical point of view, such proofs can be misleading. If the number of possible cuts is huge and the number of scenarios is astronomically large, and hence the probability of randomly generating a particular scenario is practically negligible, then an unrealistically large time could be required for such an algorithm to converge to the true optimum. Moreover, as it was pointed out, for two-stage programming the SDDP algorithm becomes the classical Kelley's cutting plane algorithm [6]. Worst case analysis of Kelley's algorithm is discussed in [9, pp.158–160], with an example of a problem where an ε-optimal solution cannot be obtained by this algorithm in less than $(\frac{1}{2\ln 2})1.15^{n-1}\ln(\varepsilon^{-1})$ calls of the oracle, i.e., the number of oracle calls grows exponentially with increase of the dimension n of the problem. It was also observed empirically that Kelley's algorithm could behave quite poorly in practice. Unfortunately it is not clear how more efficient, bundle type algorithms, can be extended to a multistage setting. Of course, in the multistage setting it can only become worse. So the bottom line is that it could be impossible to solve the SAA problem to optimality and the true problem can be computationally intractable (cf., [18]). However, with a reasonable computational effort the SDDP algorithm could produce a practically acceptable and implementable policy. Moreover, although it could be impossible to solve the multistage problem to optimality, one can compare performance of different competing policies by sampling (see Remark 6).

It was argued (see Remark 4) that the stopping criterion suggested in [11] could be too optimistic. A more realistic approach could be to stop the SDDP algorithm when the lower bounds $\underline{\vartheta}_k$ start to stabilize (cf., [5]). It could be possible to test statistically whether further runs of the SDDP algorithm significantly improve the associated policy by employing a t-test type methodology discussed in Remark 6. It also should be pointed out that stabilization of the lower bounds $\underline{\vartheta}_k$ does not mean that the SAA problem is solved to optimality. It simply means that further runs of the SDDP algorithm do not significantly improve the constructed policy.

One can try to make various improvements in a practical implementation of the SDDP method. One obvious improvement could be a procedure of removing some "redundant" cutting planes in the computational process. This should be done with a care since such procedure could alter statistical properties of computed upper and lower bounds on which a stopping criterion could be based.

We discussed here the SDDP method applied to the SAA rather than the original (true) problem. This has an advantage of separating the study of statistical properties of sample average approximations and computational properties of the SDDP algorithm applied to a finitely generated problem. It is also possible to consider a version of the SDDP method applied directly to the true problem. This would require a further investigation.

The risk averse approach, discussed in Section 4, aims at finding a compromise between optimizing (minimizing) the average cost and trying to control the upper limit of cost-to-go functions at every stage of the process. The corresponding parameters λ_t and α_t allow to tune this compromise. The computational complexity of the backward steps of the SDDP method, adjusted to the considered risk averse approach, is almost the same as in the risk neutral case. The forward step of the SDDP method has two goals. It provides trial decisions for the backward steps and allows a construction of an upper bound for value of the considered policy. Unfortunately, it is not easy and straightforward to construct respective upper bounds in the considered risk averse procedure. On the other hand, for large multistage programs such upper bounds typically are too loose to be practically useful anyway even in the risk neutral case. What could be more meaningful is to compare the expected values of constructed risk averse and risk neutral policies, say by using the statistical t-test discussed in Remark 6.

Acknowledgment

This research was partly supported by the NSF award DMS-0914785 and ONR award N000140811104.

References

[1] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath, Coherent measures of risk, Mathematical Finance 9 (1999) 203–228.
[2] Z.L. Chen, W.B. Powell, Convergent cutting plane and partialsampling algorithm for multistage stochastic linear programs with recourse, Journal of Optimization Theory and Applications 102 (1999) 497–524.
[3] W.G. Cochran, Sampling Techniques, third ed., John Wiley & Sons, New York, 1977.
[4] C.J. Donohue, J.R. Birge, The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse, Algorithmic Operations Research 1 (2006) 20–30.
[5] M. Hindsberger, A.B. Philpott, Stopping criteria in sampling strategies for multistage SLP-problems, presented at the conference "Applied Mathematical Programming and Modelling", Varenna, Italy, June 2002.
[6] J.E. Kelley, The cutting-plane method for solving convex programs, Journal of the Society for Industrial and Applied Mathematics 8 (1960) 703–712.
[7] K. Linowsky, A.B. Philpott, On the convergence of sampling based decomposition algorithms for multistage stochastic programs, Journal of Optimization Theory and Applications 125 (2005) 349–366.
[8] W.K. Mak, D.P. Morton, R.K. Wood, Monte Carlo bounding techniques for determining solution quality in stochastic programs, Operations Research Letters 24 (1999) 47–56.
[9] Yu. Nesterov, Introductory Lectures on Convex Optimization, Kluwer, Boston, 2004.
[10] V.I. Norkin, G.Ch. Pflug, A. Ruszczyński, A branch and bound method for stochastic global optimization, Mathematical Programming 83 (1998) 425–450.
[11] M.V.F. Pereira, L.M.V.G. Pinto, Multi-stage stochastic optimization applied to energy planning, Mathematical Programming 52 (1991) 359–375.

[12] A.B. Philpott, Z. Guan, On the convergence of stochastic dual dynamic programming and related methods, Operations Research Letters 36 (2008) 450–455.
[13] R.T. Rockafellar, S.P. Uryasev, Optimization of conditional value-at-risk, The Journal of Risk 2 (2000) 21–41.
[14] A. Ruszczyński, Decomposition methods, in: A. Ruszczyński, A. Shapiro (Eds.), Stochastic Programming, Handbook in OR & MS, vol. 10, North-Holland Publishing Company, Amsterdam, 2003.
[15] A. Ruszczyński, Risk-averse dynamic programming for Markov decision processes, Mathematical Programming, Series B, in press.
[16] A. Ruszczyński, A. Shapiro, Optimization of convex risk functions, Mathematics of Operations Research 31 (2006) 433–452.
[17] A. Ruszczyński, A. Shapiro, Conditional risk mappings, Mathematics of Operations Research 31 (2006) 544–561.
[18] A. Shapiro, A. Nemirovski, On complexity of stochastic programming problems, in: V. Jeyakumar, A.M. Rubinov (Eds.), Continuous Optimization: Current Trends and Applications, Springer, 2005, pp. 111–144.
[19] A. Shapiro, On complexity of multistage stochastic programs, Operations Research Letters 34 (2006) 1–8.
[20] A. Shapiro, On a time consistency concept in risk averse multi-stage stochastic programming, Operations Research Letters 37 (2009) 143–147.
[21] A. Shapiro, D. Dentcheva, A. Ruszczyński, Lectures on Stochastic Programming: Modeling and Theory, SIAM, Philadelphia, 2009.