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

ARTICLE INFO

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

ABSTRACT

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

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

Components of vectors c_t, b_t and matrices A_t, B_t are modeled as random variables forming the stochastic data process[1] $\xi_t=(c_t,A_t,B_t,b_t)$, $t=2,\ldots,T$, with $\xi_1=(c_1,A_1,b_1)$ being deterministic (not random). By $\xi_{[t]}=(\xi_1,\ldots,\xi_t)$ 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 $\xi_{[t]}=(\xi_1,\ldots,\xi_t)$ for $t=1,\ldots,T-1$. 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, $t=2,\ldots,T$, form a first order autoregressive process, i.e., $b_t=\Phi b_{t-1}+\varepsilon_t$, with appropriate matrix Φ and error vectors $\varepsilon_2,\ldots,\varepsilon_T$ being independent of each other. Then the feasibility equations of problem (1.1) can be written as

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

Therefore by replacing x_t with (x_t,b_t) and data process with $(c_t,A_t,B_t,\varepsilon_t)$, $t=2,\ldots,T$, 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 $A^\top$ 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 $\ell(x) = a + b^\top x$ 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 $\ell(x) = Q(\bar{x}) + g^\top(x - \bar{x})$ 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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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 $\tilde{\xi}^j = (\tilde{h}_j, \tilde{T}_j, \tilde{W}_j, \tilde{q}_j)$, $j = 1, \ldots, N$. 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

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

[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.