Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

AN SDDP ALGORITHM FOR MULTISTAGE STOCHASTIC PROGRAMS WITH DECISION-DEPENDENT UNCERTAINTY

Nazlıcan Arslan[1], Oscar Dowson[2], and David P. Morton[1]

[1]Industrial Engineering & Management Sciences, Northwestern University, Evanston, IL, USA
[2]Dowson Farms, Auckland, NEW ZEALAND

ABSTRACT

Stochastic programming provides mathematical models and algorithms for optimizing decisions under uncertainty. In formulating a stochastic program we typically assume that the probability distributions governing the random parameters are independent of the problem's decisions. Here, we study a multistage stochastic program with decision-dependent uncertainty. At each stage, binary decisions choose from a set of probability distributions and can increase the likelihood of favorable outcomes at a certain cost. We develop a variant of the stochastic dual dynamic programming (SDDP) algorithm to approximately solve this class of problems, using a convex relaxation of the algorithm's subproblems. This allows us to handle a type of large-scale multistage decision-dependent stochastic program, which was previously inaccessible. We provide computational results for a multi-product newsvendor problem with binary marketing options.

1 INTRODUCTION

Multistage stochastic programming is a mathematical framework used to model an agent making a sequence of decisions under uncertainty. Unlike complementary approaches in simulation-optimization, Markov decision processes, and reinforcement learning, the vast majority of the stochastic programming literature assumes that the problem's randomness is exogenous, i.e., it assumes that the probability law governing the problem's stochastic process is independent of the agent's actions. In many settings this is reasonable. In a hydroelectric system, the operator's decisions regarding releasing water for power generation and storing water in reservoirs may not affect the random external inflow from precipitation and snow melt. A small financial investor's decisions to buy and sell specific assets may not alter the random gains or losses that those assets incur in the market. On the other hand, agents ranging from advertisers to public health officials take actions that aim to influence or alter a population's behavior. In turn, our mathematical models should alter, for example, the probability distribution governing the demand for a product or service or the demand for hospital beds.

The literature on stochastic programming with decision-dependent uncertainty can be divided into two main groups. The first group assumes that the agent's decisions can alter the probability distribution governing the problem's random vector. Ahmed (2000) considers two-stage stochastic programs in which first-stage design decisions parameterize the probabilities of random outcomes. He reformulates the problem as a 0-1 fractional program, and proposes a branch-and-bound algorithm. Kopa and Rusỳ (2021) consider an asset-liability management problem in which the lender's decision on the interest rate for a loan affects the customer's probability of acceptance, defaulting, and prepayment. Hellemo et al. (2018) consider a two-stage stochastic program in which the parameters of the probability distribution depend on the first-stage decisions, and we also point to their work for a review of the stochastic programming literature on decision-dependent uncertainty.

The second group of papers assume the agent's decisions determine the nature and/or timing of learning information regarding the distribution of the random parameters. Goel and Grossmann (2004) consider an offshore gas field investment problem in which information on the gas reserve and on the efficacy of potential extraction can be attained only after an initial investment in exploration. They present a disjunctive problem formulation with decision-dependent nonanticipativity constraints and propose a decomposition algorithm. Vayanos et al. (2011) consider a similar problem with decision-dependent information discovery and propose an approximation technique that is also used in robust optimization. Among other work, see Jonsbråten et al. (1998), Goel and Grossmann (2006), and Apap and Grossmann (2017) for more examples of decision-dependent information discovery.

Decision-dependent uncertainty is also considered in distributionally robust variants of stochastic programs. For example, Basciftci et al. (2021), Luo and Mehrotra (2020), and Yu and Shen (2022) study distributionally robust optimization in which the ambiguity sets—that constrain the choice of the adversarial probability distribution—are decision-dependent.

Morton et al. (2024) formulate a class of multistage stochastic programs that incorporate modeling characteristics of Markov decision processes (MDPs). Their formulation captures action-dependent one-step transition probabilities like those in an MDP and can represent both types of decision-dependent uncertainty just sketched. Our approach is closely related to that of Morton et al. but rather than altering the one-step transition probabilities in a policy graph (Dowson 2020), in our model the agent's actions directly alter the probability mass function (pmf) governing random realizations of, say, the demand distribution. (Here, we do not pursue a model in which actions enable information gathering or statistical learning.) In particular, we use a set of binary decision variables to alter the likelihood of uncertain outcomes, such as a decision to market a certain product, which can increase the likelihood of a higher demand.

Our algorithmic approach is rooted in the stochastic dual dynamic programming (SDDP) algorithm, which originated with Pereira and Pinto (1991). SDDP is typically used to solve large-scale multistage stochastic convex programs with exogenous uncertainty when the random vectors are interstage independent, or when the stochastic process satisfies certain types of inter-stage dependence, e.g., De Queiroz and Morton (2013), Downward et al. (2020), Infanger and Morton (1996), Löhndorf and Shapiro (2019), Rebennack (2016).

SDDP iteratively refines a piecewise linear convex approximation of the expected cost-to-go function at each stage by repeating two basic steps: (i) in a forward pass, a sample path is selected from the underlying stochastic process and the current policy is implemented to provide sequential, stage-wise solutions along that path; and (ii) in a backward pass, new cuts are generated at each stage's solution from the forward pass by solving the modest number of immediate descendant subproblems at each stage and using their dual variables. In the forward pass, the current piecewise linear convex approximation at each stage specifies the policy, and those convex approximations are updated by the new cuts computed during the backward pass.

SDDP was developed for *convex* multistage stochastic programs, but it is increasingly being used to approximately solve nonconvex problems. Here, the nonconvex cost-to-go function is approximated by a cut-based convex function. This cutting-plane approximation again constructs a policy but one that is now, in general, suboptimal even as the sample size (number of backward passes) used for training the policy grows large. Importantly, in the forward pass we now solve nonconvex subproblems, which capture associated operational realities, aided by the convex cost-to-go approximation. When this approximation is a relaxation, posterior statistical bounds on the optimality gap can be computed. A key idea—and one we rely on—is to construct the convex approximation via Lagrangian duality as pioneered by Zou et al. (2019) in the SDDiP algorithm. Nonconvex SDDP algorithms have been developed and applied in a number of settings, including capacity planning in power systems (Lara et al. 2018; Lara et al. 2020), dispatching power systems under an AC power flow model (Rosemberg et al. 2022), in using a hidden Markov model in a multistage stochastic program (Siddig et al. 2021), and in a decision-dependent distributionally robust multistage problem (Yu and Shen 2022). Our work is most closely related to that of Morton et al. (2024) and Yu and Shen (2022), in that we propose to apply an SDDP variant to solve a multistage stochastic program in which the nonconvexities arise via discrete variables associated with decision-dependent uncertainty, although we do not pursue the distributionally robust formulations of the latter paper.

The paper is organized as follows. Section 2 introduces our decision-dependent multistage stochastic program and our assumptions. Section 3 presents the approximations of the cost-to-go function that we use in the forward and backward passes of our decomposition algorithm. Section 4 describes our SDDP-style decomposition algorithm. Section 5 provides numerical experiments that assess the algorithm. Finally, Section 6 concludes and provides future research directions.

2 PROBLEM STATEMENT

We consider the following T-stage stochastic program:

$$\begin{aligned} V_1(x_0,\omega_1) = \min_{x_1\geq 0, z_1} \; & c_1 x_1 + \sum_{k\in K_1} z_{k1}\left(f_{k1} + \sum_{\omega_2\in\Omega_2} p_{k2}^{\omega_2} V_2(x_1,\omega_2)\right), \\ \text{s.t. } & A_1 x_1 = B_0 x_0 + b_1, \\ & \sum_{k\in K_1} z_{k1} = 1, \\ & z_{k1}\in\{0,1\},\ k\in K_1, \end{aligned}$$

where for $t = 2,\ldots,T-1$,

$$V_t(x_{t-1},\omega_t) = \min_{x_t\geq 0, z_t} c_t^{\omega_t} x_t + \sum_{k\in K_t} z_{kt}\left(f_{kt} + \sum_{\omega_{t+1}\in\Omega_{t+1}} p_{k,t+1}^{\omega_{t+1}} V_{t+1}(x_t,\omega_{t+1})\right), \tag{1a}$$

$$\text{s.t. } A_t^{\omega_t} x_t = B_{t-1}^{\omega_t} x_{t-1} + b_t^{\omega_t}, \tag{1b}$$

$$\sum_{k\in K_t} z_{kt} = 1, \tag{1c}$$

$$z_{kt}\in\{0,1\},\ k\in K_t, \tag{1d}$$

and where

$$V_T(x_{T-1},\omega_T) = \min_{x_T\geq 0} c_T^{\omega_T} x_T, \tag{2a}$$

$$\text{s.t. } A_T^{\omega_T} x_T = B_{T-1}^{\omega_T} x_{T-1} + b_T^{\omega_T}. \tag{2b}$$

We assume $A_t^{\omega_t}\in\mathbb{R}^{m_t\times n_t}$ and other matrices and vectors are also real-valued and conform in dimension. The latter point includes treating vectors multiplying x_t as row vectors, such as $c_t^{\omega_t}$ as well as subsequent Lagrangian dual variables, λ_t, and cut gradient coefficients, β_t.

Realizations of the random parameters (A_t, B_{t-1}, b_t, c_t) are denoted $(A_t^{\omega_t}, B_{t-1}^{\omega_t}, b_t^{\omega_t}, c_t^{\omega_t}),\ \omega_t\in\Omega_t, t = 2,\ldots,T$. Here, we assume binary decisions, z_{kt}, represent investment decisions that can increase the likelihood of favorable outcomes and $f_{kt}\in\mathbb{R}$ represents the corresponding cost of investment options. Anticipating our numerical example of Section 5, and for simplicity, we will say that the z_{kt} variables represent *marketing options*. We assume x_0 is given and ω_1 is degenerate and suppressed in (A_1, B_0, b_1, c_1). In what follows we let $K_T = \emptyset$ indicate that stage T does not have marketing options. These conventions allow our entire problem to be represented more compactly, i.e., be represented by recursion (1) for $t = 1,\ldots,T$. The candidate pmfs $p_{kt}^{\omega_t}$, $\omega_t\in\Omega_t$, are indexed by $k\in K_t$ because they are selected by the binary marketing decision, z_{kt}.

We make the following assumptions throughout the paper:

Assumption 1 The sample space at stage t, Ω_t, is finite for all $t = 1,\ldots,T$, and the random parameters $(A_t, B_{t-1}, b_t, c_t),\ t = 1,\ldots,T$, are interstage independent.

Assumption 2 The subproblem defining $V_t(x_{t-1},\omega_t)$ in recursion (1) is feasible and has a finite optimal solution for every incoming feasible solution x_{t-1} and every sample point $\omega_t\in\Omega_t$, $t = 1,\ldots,T$.

3 CONVEX RELAXATION

We know $V_T(\cdot,\omega_T)$ is piecewise linear and convex, and hence so is $\sum_{\omega_T\in\Omega_T} p_{kT}^{\omega_T} V_T(\cdot,\omega_T)$ for each $k\in K_{T-1}$. The introduction of the marketing decisions $z_{k,T-1}$ means that $V_{T-1}(\cdot,\omega_{T-1})$ is effectively the minimum of a collection of convex cost-to-go functions, across $k\in K_{T-1}$, and hence is, in general, nonconvex. Mathematically we see that the product of z_{kt} and $V_{t+1}(x_t,\omega_{t+1})$ in recursion (1) introduces nonconvexity in the cost-to-go function.

We will provide a convex relaxation of $V_t(x_{t-1},\omega_t)$ defined in recursion (1). We do so by constructing a convex, piecewise linear lower bound. The proof that our construction is valid is inductive. The inductive hypothesis is that the cost-to-go function is bounded below by the maximum of a finite set of affine functions, i.e., cuts:

$$V_{t+1}(x_t,\omega_{t+1}) \geq \max_{\ell\in L_{t+1}^{\omega_{t+1}}} \left[\alpha_{t,\ell}^{\omega_{t+1}} + \beta_{t,\ell}^{\omega_{t+1}} x_t\right]. \tag{3}$$

By using dual extreme points of (2) and forming the corresponding cuts, condition (3) already holds for $V_T(x_{T-1},\omega_T)$ because as just indicated it is piecewise linear and convex in x_{T-1}. Thus, for $t=1,\ldots,T-1$, we propose approximating (1) by:

$$V_t^L(\bar{x}_{t-1},\omega_t) = \min_{x_t\geq 0, z_t, \theta_t, \Theta_t, x_{t-1}} \; c_t^{\omega_t} x_t + \sum_{k\in K_t} f_{kt} z_{kt} + \Theta_t, \tag{4a}$$

$$\text{s.t.} \quad A_t^{\omega_t} x_t = B_{t-1}^{\omega_t} x_{t-1} + b_t^{\omega_t}, \tag{4b}$$

$$\Theta_t \geq \sum_{k\in K_t} z_{kt} \sum_{\omega_{t+1}\in\Omega_{t+1}} p_{k,t+1}^{\omega_{t+1}} \theta_t^{\omega_{t+1}}, \tag{4c}$$

$$\theta_t^{\omega_{t+1}} \geq \alpha_{t,\ell}^{\omega_{t+1}} + \beta_{t,\ell}^{\omega_{t+1}} x_t, \;\; \ell\in L_{t+1}^{\omega_{t+1}}, \omega_{t+1}\in\Omega_{t+1}, \tag{4d}$$

$$x_{t-1} = \bar{x}_{t-1} \;\; [\lambda_t], \tag{4e}$$

$$\sum_{k\in K_t} z_{kt} = 1, \tag{4f}$$

$$z_{kt}\in\{0,1\}, \;\; k\in K_t. \tag{4g}$$

We use Θ_t as a proxy for $\sum_{k\in K_t} z_{kt} \sum_{\omega_{t+1}\in\Omega_{t+1}} p_{k,t+1}^{\omega_{t+1}} V_{t+1}(x_t,\omega_{t+1})$ and $\theta_t^{\omega_{t+1}}$ as a proxy for $V_{t+1}(x_t,\omega_{t+1})$. With a full set of cuts in (4d) we have $V_{T-1}^L(\cdot,\omega_{T-1}) = V_{T-1}(\cdot,\omega_{T-1})$, but this will not be true for $t<T-1$. Constraints (4c) are nonlinear because they involve products of the marketing decisions, z_{kt}, and approximate cost-to-go variables, $\theta_t^{\omega_{t+1}}$. That said, we can reformulate what we call the forward-pass subproblem (4), or simply the *forward subproblem*, as a linear mixed-integer program (MIP) by replacing these bilinear terms with the standard McCormick inequalities (McCormick 1976). (We do not detail the reformulation.)

We now complete our inductive proof and simultaneously describe a key step towards an implementable algorithm, i.e., we show that if inequality (3) holds for $V_{t+1}(x_t,\omega_{t+1})$ then we have a mechanism to produce cuts that yield that same inequality with t decremented by one. Following Zou et al. (2019), we created local copies of the x_{t-1} variables in (4) and we now form the Lagrangian dual of the forward subproblem by relaxing the associated fishing constraint (4e) as follows:

$$\underline{V}_t^L(\bar{x}_{t-1},\omega_t) = \max_{\lambda_t} \min_{x_t\geq 0, z_t, \theta_t, \Theta_t, x_{t-1}} \quad c_t^{\omega_t} x_t + \sum_{k\in K_t} f_{kt} z_{kt} + \Theta_t - \lambda_t(x_{t-1} - \bar{x}_{t-1}), \tag{5a}$$

$$\text{s.t.} \quad A_t^{\omega_t} x_t = B_{t-1}^{\omega_t} x_{t-1} + b_t^{\omega_t}, \tag{5b}$$

$$\Theta_t \geq \sum_{k\in K_t} z_{kt} \sum_{\omega_{t+1}\in\Omega_{t+1}} p_{k,t+1}^{\omega_{t+1}} \theta_t^{\omega_{t+1}}, \tag{5c}$$

$$\theta_t^{\omega_{t+1}} \geq \alpha_{t,\ell}^{\omega_{t+1}} + \beta_{t,\ell}^{\omega_{t+1}} x_t, \ \ell \in L_{t+1}^{\omega_{t+1}}, \omega_{t+1} \in \Omega_{t+1}, \tag{5d}$$

$$\sum_{k\in K_t} z_{kt} = 1, \tag{5e}$$

$$z_{kt} \in \{0,1\}, \ k \in K_t. \tag{5f}$$

We call relaxation (5) the *backward subproblem*, and we note that $\underline{V}_t^L(\cdot,\omega_t)$ is a piecewise linear convex function. We will use the relaxed formulation (5) in the backward pass of the SDDP algorithm. We construct lower-bounding cuts for $\underline{V}_t^L(\cdot,\omega_t)$ and hence compute a lower bound for $V_t^L(\cdot,\omega_t)$ and $V_t(\cdot,\omega_t)$ using:

$$\beta_{t-1,\ell}^{\omega_t} = \lambda_t^{\omega_t}, \tag{6a}$$

$$\alpha_{t-1,\ell}^{\omega_t} = \underline{V}_t^L(\bar{x}_{t-1},\omega_t) - \beta_{t-1,\ell}^{\omega_t}\, \bar{x}_{t-1}. \tag{6b}$$

We summarize our development in this section in the following theorem.

Theorem 1 Let Assumptions 1 and 2 hold. Then for $t = 2,\ldots,T$, $\underline{V}_t^L(\cdot,\omega_t)$ is convex and

$$V_t(x_{t-1},\omega_t) \geq V_t^L(x_{t-1},\omega_t) \geq \underline{V}_t^L(x_{t-1},\omega_t) \geq \max_{\ell\in L_t^{\omega_t}} \left[\alpha_{t-1,\ell}^{\omega_t} + \beta_{t-1,\ell}^{\omega_t}\, x_{t-1}\right]. \tag{7}$$

The first three terms in (7) are the optimal values of (1), (4), and (5), respectively. The cut coefficients in the fourth term are computed via equations (6) using dual multipliers, $\lambda_t^{\omega_t}$, and function values, $\underline{V}_t^L(\bar{x}_{t-1},\omega_t)$, from the backward subproblem (5).

The final inequality in (7) still holds if we compute cuts by replacing $\lambda_t^{\omega_t}$ and $\underline{V}_t^L(\bar{x}_{t-1},\omega_t)$ with suboptimal counterparts that fail to "fully" maximize over λ_t in (5). We now have the ingredients in place to construct our variant of SDDP to approximately solve (1).

4 SDDP ALGORITHM

Algorithm 1 details our variant of SDDP. As we have discussed, the algorithm iteratively performs a forward pass (steps 5-8) and a backward pass (steps 9-15). For each stage t and achievable value of $(\bar{x}_{t-1},\omega_t)$, a policy must specify a stage-t feasible solution $\bar{x}_t$. This is accomplished in the forward pass by solving the sequence of nonconvex subproblems (4), with the cuts (4d) that have been accumulated from previous iterations. A single forward pass amounts to solving $T-1$ subproblems (4). Knowing the sequence $(\bar{x}_{t-1},\omega_t)$, $t = 2,\ldots,T$, from the forward pass, the backward pass solves a total of $\sum_{t=2}^T |\Omega_t|$ subproblems (5) to construct cuts for stages $t = T-1, T-2,\ldots,1$. Thus a single iteration of Algorithm 1 constitutes linear effort in T. Step 13 indicates that we should add $|\Omega_t|$ cuts to stage $t-1$, but in practice, we only add non-redundant cuts.

Solving both (4) and (5) requires solving linear MIPs. Executing the "$\max_{\lambda_t}$" in the backward subproblem (5) can be done using a subgradient algorithm or a bundle method and requires repeatedly solving the inner minimizing linear MIP (5). Again, we can terminate with a suboptimal λ_t, although we tend to solve subproblem (5) precisely in the final iterations of Algorithm 1.

Multiple papers in the literature pre-specify a fixed number of iterations for SDDP algorithms. We could terminate with such a rule in Algorithm 1's step 16. Using a heuristic to assess whether, for a sufficient number of iterations, growth in the lower bounds $\underline{V}_1^L(\bar{x}_0,\omega_1)$ and $V_1^L(\bar{x}_0,\omega_1)$ has stalled, or more generally only redundant cuts are being generated throughout the stages might be a better option. In Section 5 we use a fixed number of iterations, but that number is informed by marginal growth in the lower bounds. As discussed shortly, Algorithm 2 estimates the current policy's expected cost. For convex models, this upper-bound estimator and $V_1^L(\bar{x}_0,\omega_1)$ can be combined to guide termination, accounting for sequential testing with a simulation-based estimator (Bayraksan and Morton 2011; Morton 1998). For nonconvex problems, we can terminate in the same way only if we recognize that the optimality gap may not shrink to zero, even as training effort grows large.

Marketing decisions alter problem (1)'s probability mass functions. In extreme cases, in the forward pass, if we were to sample according to the pmfs specified by the marketing decisions obtained by solving subproblem (4), then we would not visit parts of the tree. As a result, we could fail to explore the tree sufficiently, and build weak approximations of the cost-to-go functions. Even in the absence of zero-value $p_{kt}^{\omega_{t+1}}$ masses, small values alter the frequency with which we visit parts of the tree, and again harm cut construction. For this reason, we sample uniformly from Ω_t in step 6 of Algorithm 1. (In our numerical example in the next section, the uniform distribution corresponds to the "no marketing" decision.) Other strategies such as alternating between sampling from $\sum_{k\in K_{t-1}} p_{kt}^{\omega_t}\bar{z}_{k,t-1}$ and sampling uniformly are possible.

Algorithm 2 takes as input the cuts (4d) for $t=1,\ldots,T-1$ in the forward subproblem (4); i.e., the algorithm takes as input the policy that Algorithm 1 gives as output. Algorithm 2 then executes that policy along n i.i.d. forward sample paths. Because x_0 is given and ω_1 is degenerate, we only need to solve the first-stage subproblem once, and we do in step 3. Steps 4-10 execute the n i.i.d. forward paths, and in step 6 we sample ω_t according to the pmf associated with the previous stage's marketing decision, along the corresponding sample path. This form of sampling now ensures that the output of Algorithm 2 is an unbiased estimator associated with Algorithm 1's policy, which includes specification of z_{kt}.

Algorithm 1 Decision-dependent SDDP

1: **Input:** $\bar{x}_0$, initial state vector; initialize $\ell=0$
2: **Output:** Cuts (4d) for $t=1,\ldots,T-1$, which together with forward subproblem (4) form a policy; lower bound on optimal expected cost, $V_1^L(x_0,\omega_1)$
3: **while** termination criterion not met **do**
4: Let $\ell=\ell+1$
5: **for** $t\in\{1,\ldots,T-1\}$ **do** ▷ Forward pass
6: Sample ω_t uniformly from Ω_t
7: With input $(\bar{x}_{t-1},\omega_t)$, solve subproblem (4) to find $\bar{x}_t$
8: **end for**
9: **for** $t\in\{T,\ldots,2\}$ **do** ▷ Backward pass
10: **for** $\omega_t\in\Omega_t$ **do**
11: Solve subproblem (5) with input $(\bar{x}_{t-1},\omega_t)$
12: Use equations (6) to calculate cut coefficients $(\alpha_{t-1,\ell}^{\omega_t},\beta_{t-1,\ell}^{\omega_t})$
13: Add cuts to the stage $t-1$ version of subproblems (4) and (5)
14: **end for**
15: **end for**
16: Assess termination criteria
17: **end while**

Algorithm 2 Decision-dependent SDDP forward simulation for estimating an upper bound

Input: n, number of forward simulations; $\bar{x}_0$, initial state; ω_1, degenerate first-stage sample point; cuts (4d) for $t = 1, \ldots, T-1$, computed from Algorithm 1
Output: $\hat{v}$, upper bound estimator, and $\hat{s}$ its sample standard deviation
With input $(\bar{x}_0, \omega_1)$, solve subproblem (4) for $t = 1$ to find $\bar{x}_1$ and $\bar{z}_1$
for $j \in \{1, \ldots, n\}$ **do**
  for $t \in \{2, \ldots, T\}$ **do**
    Sample ω_t from Ω_t according to pmf $\sum_{k \in K_{t-1}} p_{kt}^{\omega_t} \bar{z}_{k,t-1}$
    With input $(\bar{x}_{t-1}, \omega_t)$, solve subproblem (4) to find $\bar{x}_t$ and $\bar{z}_t$
  end for
  Calculate the sample path's cost, $v_j = \sum_{t=1}^{T} \left[c_t^{\omega_t} \bar{x}_t + \sum_{k \in K_t} f_{kt} \bar{z}_{kt} \right]$
end for
Calculate upper bound $\hat{v} = \frac{1}{n} \sum_{j=1}^{n} v_j$ and $\hat{s}^2 = \frac{1}{n-1} \sum_{j=1}^{n} (v_j - \hat{v})^2$

5 NUMERICAL EXAMPLE

We consider a multistage, multi-product capacitated newsvendor problem with marketing to assess the performance of the algorithm we have proposed. Table 1 gives the sets, parameters, and decision variables for our example. In each time period, the agent can buy item $i \in I$ at a unit cost of b_i, and in the next period sell it for a unit price of s_i. Demand is random and "soft," i.e., the agent can sell up to ω_{it} units of product i in period t, but demand need not be satisfied. Inventory incurs a unit holding cost of h_i. The agent can invest in marketing for each product type i. We assume the set of marketing options K includes all product-marketing combinations, i.e., $|K| = 2^{|I|}$. The marketing cost for option k is f_k.

Table 1: Notation for multi-product capacitated newsvendor problem with marketing.

Sets	
$i \in I$	set of products
$k \in K$	set of marketing strategies (e.g., for $\lvert I \rvert = 2$, K={don't market, market product 1, market product 2, market both})
Parameters	
$b_i > 0$	unit cost for agent to buy product i
$s_i > 0$	unit price when agent sells product i
$h_i > 0$	unit cost of holding inventory of product i
f_k	cost of marketing strategy k
ω_{it}	random demand for product i in period t
$p_{kt}^{\omega_t}$	probability of observing random demand ω_t in period t after choosing marketing strategy k in period $t-1$
C	total budget for agent's purchases across all products $i \in I$ in each period
Decision variables	
x_{it}	inventory of product i at the end of period t
u_{it}^s	amount of product i sold in period t
u_{it}^b	amount of product i bought in period t
z_{kt}	1 if agent chooses marketing option k in time period t, 0 otherwise

Model (8) specializes problem (1) for our newsvendor example for $t = 1,\dots,T$:

$$V_t(x_{t-1},\omega_t) = \min_{\substack{x_t\geq 0, z_t,\\ u_t^b, u_t^s \geq 0}} \sum_{i\in I}\left(-s_i u_{it}^s + b_i u_{it}^b + h_i x_{it}\right) + \sum_{k\in K_t} z_{kt}\left(f_k + \sum_{\omega_{t+1}\in\Omega_{t+1}} p_{k,t+1}^{\omega_{t+1}} V_{t+1}(x_t,\omega_{t+1})\right), \quad (8a)$$

$$\text{s.t. } u_{it}^s \leq x_{i,t-1},\ i\in I, \quad (8b)$$

$$u_{it}^s \leq \omega_{it},\ i\in I, \quad (8c)$$

$$x_{it} = x_{i,t-1} - u_{it}^s + u_{it}^b,\ i\in I, \quad (8d)$$

$$\sum_{i\in I} u_{it}^b \leq C, \quad (8e)$$

$$\sum_{k\in K_t} z_{kt} = 1, \quad (8f)$$

$$z_{kt}\in\{0,1\},\ k\in K_t. \quad (8g)$$

Constraints (8b)-(8c) limit sales based on available inventory and demand. Constraint (8d) tracks inventory from one period to the next. Constraint (8e) limits total purchases in each period based on the agent's budget. Constraints (8f)-(8g) replicate their form from the general model (1), with the note that for $t = 1,\dots,T-1$ we have $K_t = K$, where K is from Table 1, and $K_T = \emptyset$ precludes marketing in the final time period. The objective function in (8a) accounts for revenue from sales, the costs from buying, holding, and marketing, along with the expected cost-to-go. We form vectors in the standard way, e.g., $\omega_t = [\omega_{it}]_{i\in I}$ and $z_t = [z_{kt}]_{k\in K}$.

In our numerical experiments, we consider $T\in\{10,15,20,25\}$, $|I|\in\{1,2,3\}$, $b_i = 2$ for all $i\in I$, $s = [8,10,12]$ for the three products, and $h_i = 0.1$ for all $i\in I$. Marketing costs 5 units for each product, so $f_k = 10$ if option k markets two products. Demand for each product has two realizations, low and high, which have equal probability if the agent does not market. Marketing a product increases its marginal probability of high demand by 0.05. Low and high demand values are 20 and 50 for product 1, 10 and 60 for product 2, and 5 and 65 for product 3. Contingent on the marketing decision, the probability distributions for demand are the same in each period.

The budget, C, plays a key role in model (8). When the budget is sufficiently large, the problem is relatively easy to solve and the solution mimics that of a collection of single-product problems. Similarly, if C is sufficiently small then the optimal policy is to restrict orders and marketing to the product with the highest profit margin. We observed that a budget, C, which corresponds to roughly the 75th percentile of aggregate demand when we do not market, yields interesting and challenging test problems.

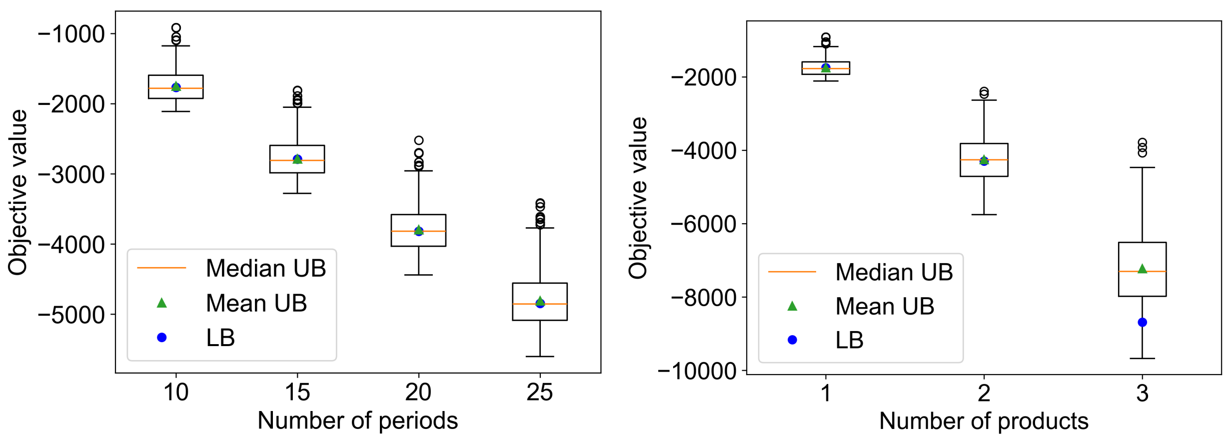
Assumption 1 is satisfied given that the demand has a finite distribution that is independent across the periods. For a given feasible inventory level $x_{t-1}\geq 0$, and random demand $\omega_{it}\geq 0$, we can always set the marketing decision $z_t = (z_{kt})_{k\in K}$ to any unit vector, and let $u_{it}^b = 0$, $u_{it}^s = \min\{x_{i,t-1},\omega_{it}\}$, and $x_{it} = x_{i,t-1} - u_{it}^s$ for $i\in I$ implying that Assumption 2 is satisfied. Thus the conditions for Theorem 1 hold.

The subproblems (4) and (5) in Algorithms 1 and 2 are solved using Gurobi 11.0.1 (Gurobi Optimization, LLC. 2024). The Lagrangian dual problem associated with subproblem (5) is solved with a subgradient algorithm. The experiments were conducted on a macOS laptop with 2.3 GHz Quad-Core Intel Core i7 processor with 16 GB of RAM, and the algorithms were implemented using Python 3.9.

We ran Algorithm 1 for a fixed number of 50 iterations, which was large enough so that subsequent iterations would change the cost-to-go approximations only marginally. We then ran Algorithm 2 for $n = 1000$ i.i.d. forward paths, and Figure 1 shows results for six example problems (one repeats across the subfigures). The boxes correspond to the 25th and 75th percentile of the realized cost (and the whiskers cover about 99% of the simulated samples). Figure 1a shows results for a single-product example as the number of stages, T, grows. While the spread of the distribution of v_j values, $j = 1,\dots,1000$ (see Algorithm 2), grows with T, the deterministically valid lower bound $V_1^L(x_0,\omega_1)$ from Algorithm 1 and the Monte Carlo upper bound $\hat{v}$ from Algorithm 2 are close, despite our algorithm using a convex relaxation of a nonconvex problem. Figure 1b shows that as the number of products grows the spread of the distribution again grows, and for $|I| = 3$ products, the gap between the upper and lower bounds is substantial. Figure 1's box plots depict the spread in the distribution of the sample population, i.e., the v_j values, $j = 1, \dots, n$.

Another point of interest is the uncertainty associated with the estimator $\hat{v}$ from Algorithm 2. The standard errors, i.e., $\hat{s}/\sqrt{n}$, associated with the first set of examples from Figure 1a are 8, 9, 11, and 12, respectively. Similarly, for the second set of examples from Figure 1b, the standard errors are 8 (repeated), 20, and 33, respectively. With these sampling errors, we can construct confidence intervals for the upper bound and the optimality gap. As percentage of the upper bound, a 95% confidence intervals on the optimality gap range from 0.7% to 1.4% for the first five examples and then jump to 21% for the sixth instance. Preliminary experiments with more sophisticated Lagrangian methods, which are not reported here, reduced the gap in the final instance by more than half.

Figure 2 shows the inventory levels and marketing decisions based on an optimized policy over a single simulated sample path. Figure 2a depicts results of the policy for a single-product example with $T = 25$, which indicates that the agent should market when the inventory level exceeds approximately 40 units. Figure 2b shows a similar policy for a two-product example with $T = 10$. The product with higher profit margin is prioritized, has higher inventory level, and is always marketed. For the second product, marketing is limited to periods when the inventory level is high enough.



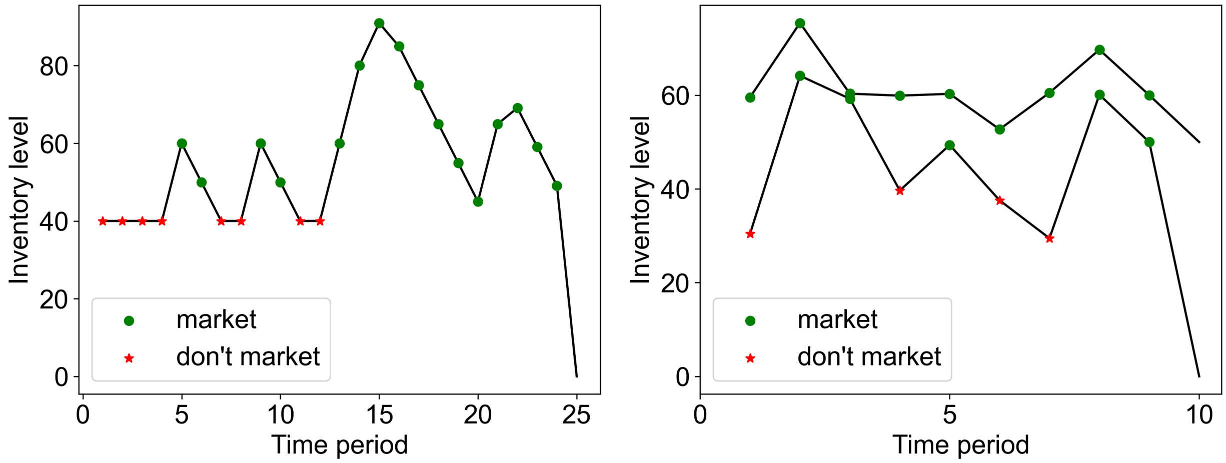
(a) $|I| = 1$ (b) $T = 10$

Figure 1: Box plots of the cost (objective function) incurred for $n = 1000$ simulations from Algorithm 2. The blue dots represent the lower bound, $V_1^L(x_0, \omega_1)$, calculated using Algorithm 1. The green triangles represent $\hat{v}$, the estimated expected cost calculated using $n = 1000$ sample path simulations. Part 1a shows results for $T = 10, 15, 20, 25$ periods for a family of single-product instances. Part 1b shows similar results for $|I| = 1, 2, 3$ products when $T = 10$.

6 CONCLUSION

In this paper, we studied a multistage stochastic program with decision-dependent uncertainty. We assumed an agent can choose from a set of probability distributions at each time period to increase the likelihood of favorable outcomes with a corresponding cost. We solved our problem with an SDDP-style algorithm that relies on a convex relaxation of the subproblems, and yet forms an unbiased Monte Carlo estimator of the cost associated with the policy that our SDDP algorithm constructs for the nonconvex problem. We provided computational results for a multistage, multi-product capacitated newsvendor problem with marketing.

We described and implemented cutting planes for the cost-to-go function that are derived from a relatively straightforward Lagrangian relaxation of the state variable's fishing constraint. Our subgradient



(a) $|I| = 1$ (b) $|I| = 2$

Figure 2: Inventory levels and marketing decisions according to Algorithm 1's optimized policy on a single simulated sample path. Black lines shows the inventory level for each product in each time period. The red stars represent the decision to not market a product, and the green dots represent the decision to market. Part 2a shows results for a single-product example over 25 periods. Part 2b shows results for a two-product example over 10 periods.

algorithm is also relatively simple and sub-optimizes over the dual multiplier. We empirically observed that the nested nature of cut computation over multiple stages can amplify weakness in cuts. Future work should include pursuing more sophisticated methods for generating cuts via the Lagrangian dual.

In our numerical results we did not compare the algorithm that we propose against competitors because we do not know how else to approximately solve the problem while producing reasonable lower bounds. To our knowledge, other approaches to stochastic programs with decision-dependent uncertainty, including those that we review in Section 1, are not meant to scale to problems with more than a few stages. We can imagine different types of convex relaxations that include a trade-off between requisite computational effort and tightness, and this is a possible direction for future work.

We sampled uniformly from Ω_t in step 6 of Algorithm 1, as opposed to sampling from $\sum_{k \in K_{t-1}} p_{kt}^{\omega_t} \bar{z}_{k,t-1}$. This choice helps us explore the scenario tree and helps to generate cuts at a richer set of values of the inventory state variable, x_t, at each stage. However, even with this uniform sampling scheme, marketing decisions still influence our cost-to-go approximation. For example, in the problem of Section 5 an agent who markets makes more aggressive product purchases, leading to larger inventory levels. An alternative for the forward pass of Algorithm 1 would be to sample a random marketing decision—instead of using an optimized decision—and then solve the forward subproblem (4) with that marketing decision. This should enable greater exploration, and may be particularly important in the early iterations of the algorithm.

We assumed that marketing decisions can change the probability distribution of random outcomes in a way that is known a priori to the agent. In many settings the agent will learn the effect of marketing based on demand realizations. Thus an important future extension of our work is to simultaneously model decision-dependent uncertainty and statistical learning. Ensuring the forward pass of an SDDP algorithm sufficiently explores the state space at each stage will grow in importance in such a setting.

ACKNOWLEDGMENTS

This work was supported by grant no. 17STQAC00001-04-00US from the U.S. Department of Homeland Security (DHS). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DHS. The authors thank two anonymous referees for comments that improved the paper.

REFERENCES

Ahmed, S. 2000. *Strategic Planning Under Uncertainty: Stochastic Integer Programming Approaches*. Ph.D. thesis, Department of Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois. https://www.proquest.com/openview/74b7de04266ce57cd23f8a7849cb853c/1?pq-origsite=gscholar&cbl=18750&diss=y, accessed 22[th] September 2024.

Apap, R. M. and I. E. Grossmann. 2017. "Models and Computational Strategies for Multistage Stochastic Programming under Endogenous and Exogenous Uncertainties". *Computers & Chemical Engineering* 103:233–274.

Basciftci, B., S. Ahmed, and S. Shen. 2021. "Distributionally Robust Facility Location Problem under Decision-dependent Stochastic Demand". *European Journal of Operational Research* 292(2):548–561.

Bayraksan, G. and D. P. Morton. 2011. "A Sequential Sampling Procedure for Stochastic Programming". *Operations Research* 59(4):898–913.

De Queiroz, A. R. and D. P. Morton. 2013. "Sharing Cuts under Aggregated Forecasts when Decomposing Multi-stage Stochastic Programs". *Operations Research Letters* 41(3):311–316.

Downward, A., O. Dowson, and R. Baucke. 2020. "Stochastic Dual Dynamic Programming with Stagewise-dependent Objective Uncertainty". *Operations Research Letters* 48:33–39.

Dowson, O. 2020. "The Policy Graph Decomposition of Multistage Stochastic Programming Problems". *Networks* 76(1):3–23.

Goel, V. and I. E. Grossmann. 2004. "A Stochastic Programming Approach to Planning of Offshore Gas Field Developments under Uncertainty in Reserves". *Computers & Chemical Engineering* 28(8):1409–1429.

Goel, V. and I. E. Grossmann. 2006. "A Class of Stochastic Programs with Decision Dependent Uncertainty". *Mathematical Programming* 108(2):355–394.

Gurobi Optimization, LLC. 2024. "Gurobi Optimizer Reference Manual". https://www.gurobi.com, accessed 22[th] September 2024.

Hellemo, L., P. I. Barton, and A. Tomasgard. 2018. "Decision-dependent Probabilities in Stochastic Programs with Recourse". *Computational Management Science* 15(3):369–395.

Infanger, G. and D. P. Morton. 1996. "Cut Sharing for Multistage Stochastic Linear Programs with Interstage Dependency". *Mathematical Programming* 75(2):241–256.

Jonsbråten, T. W., R. J. Wets, and D. L. Woodruff. 1998. "A Class of Stochastic Programs with Decision Dependent Random Elements". *Annals of Operations Research* 82(0):83–106.

Kopa, M. and T. Rusỳ. 2021. "A Decision-dependent Randomness Stochastic Program for Asset–liability Management Model with a Pricing Decision". *Annals of Operations Research* 299(1):241–271.

Lara, C. L., D. S. Mallapragada, D. J. Papageorgiou, A. Venkatesh, and I. E. Grossman. 2018. "Deterministic Electric Power Infrastructure Planning: Mixed-integer Programming Model and Nested Decomposition Algorithm". *European Journal of Operational Research* 271(3):1037–1054.

Lara, C. L., J. D. Siirola, and I. E. Grossmann. 2020. "Electric Power Infrastructure Planning under Uncertainty: Stochastic Dual Dynamic Integer Programming (SDDiP) and Parallelization Scheme". *Optimization and Engineering* 21:1243–1281.

Löhndorf, N. and A. Shapiro. 2019. "Modeling Time-dependent Randomness in Stochastic Dual Dynamic Programming". *European Journal of Operational Research* 273(2):650–661.

Luo, F. and S. Mehrotra. 2020. "Distributionally Robust Optimization with Decision Dependent Ambiguity Sets". *Optimization Letters* 14:2565–2594.

McCormick, G. P. 1976. "Computability of Global Solutions to Factorable Nonconvex Programs: Part I-Convex Underestimating Problems". *Mathematical Programming* 10(1):147–175.

Morton, D. P. 1998. "Stopping Rules for a Class of Sampling-based Stochastic Programming Algorithms". *Operations Research* 46(5):710–718.

Morton, D. P., N. Arslan, O. Dowson, and B. K. Pagnoncelli. 2024. "Markov Decision Processes and Multi-stage Stochastic Programming: Modeling and Computation". In *25[th] International Symposium on Mathematical Programming*. July 21[st]-26[th], Montreal, Canada.

Pereira, M. V. and L. M. Pinto. 1991. "Multi-stage Stochastic Optimization Applied to Energy Planning". *Mathematical Programming* 52:359–375.

Rebennack, S. 2016. "Combining Sampling-based and Scenario-based Nested Benders Decomposition Methods: Application to Stochastic Dual Dynamic Programming". *Mathematical Programming* 156(1-2):343–389.

Rosemberg, A., A. Street, J. D. Garcia, T. Silva, D. Valladão, and O. Dowson. 2022. "Assessing the Cost of Network Simplifications in Hydrothermal Power Systems". *IEEE Transactions on Sustainable Energy* 13(1):196–206.

Siddig, M., Y. Song, and A. Khademi. 2021. "Maximum-posterior Evaluation for Partially Observable Multistage Stochastic Programming". In *Proceedings of the 2021 IISE Annual Conference*. May 22$^{\text{th}}$-25$^{\text{th}}$, Montreal, Canada.

Vayanos, P., D. Kuhn, and B. Rustem. 2011. "Decision Rules for Information Discovery in Multi-stage Stochastic Programming". In *2011 50th IEEE Conference on Decision and Control and European Control Conference*. December 12$^{\text{th}}$-15$^{\text{th}}$, Orlando, Florida, USA, 7368-7373.

Yu, X. and S. Shen. 2022. "Multistage Distributionally Robust Mixed-integer Programming with Decision-dependent Moment-based Ambiguity Sets". *Mathematical Programming* 196:1025–1064.

Zou, J., S. Ahmed, and X. A. Sun. 2019. "Stochastic Dual Dynamic Integer Programming". *Mathematical Programming* 175:461–502.

AUTHOR BIOGRAPHIES

NAZLICAN ARSLAN is a Ph.D. candidate in the Department of Industrial Engineering and Management Sciences at Northwestern University. Her research focuses on stochastic and robust optimization. Her email address is nazlicanarslan2024@u.northwestern.edu.

OSCAR DOWSON is based in Auckland, New Zealand, where he works on open source software for mathematical optimization. His email address is oscar@dowsonfarms.co.nz.

DAVID P. MORTON is the Walter P. Murphy Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University. His research interests include stochastic and large-scale optimization. His email address is david.morton@northwestern.edu.