A Convergent Cutting-Plane and Partial-Sampling Algorithm for Multistage Stochastic Linear Programs with Recourse

Zhi-Long Chen

Warren B. Powell

Department of Civil Engineering and Operations Research
Princeton University
Princeton, NJ 08544

Technical Report SOR-97-11

Revised, August, 1998

Abstract

We propose an algorithm for multistage stochastic linear programs with recourse where random quantities in different stages are independent. The algorithm successively approximates expected recourse functions by building up valid cutting planes to support these functions from below. In each iteration, for the expected recourse function in each stage, one cutting-plane is generated using the dual extreme points of the next-stage problem that have been found so far. We prove that the algorithm is convergent with probability one.

Keywords: Multistage stochastic programming; Cutting plane; Sampling; Convergence with probability one

1 Introduction

Numerous real-world problems in applications such as transportation (see, e.g. Powell et al. [17]), production planning (see, e.g. Escudero et al. [8]), financial planning (see, e.g. Mulvey and Vladimirou [16]), and many other fields (see Birge [2], and Birge and Mulvey [3]), can be formulated as two-stage or multistage stochastic linear programs with recourse. The characteristics of such a problem can be summarized as follows: (1) a stage usually represents a time period; (2) the very beginning of the first stage is viewed as here and now; (3) at the beginning of each stage, we know deterministically all the data in this stage, but only know probabilistically all the data in the future stages; (4) at the beginning of the first stage, decisions must be made before the realization of random data in the future stages; (5) once the random data in a stage becomes known, correction (i.e. recourse) actions are allowed to compensate the decisions for this stage made earlier; (6) the goodness of the decision making is measured by the total cost consisting of the deterministic cost in the first stage and the total expected cost in the future stages.

Methods for stochastic linear programs can be generally classified into those which use a fixed sample of realizations (scenario-based methods) and those which iteratively sample realizations as the algorithm progresses (what we will call sampling-based methods).

Scenario-based methods normally approximate a stochastic problem using a relatively small set of realizations which allow the problem to be solved as a (typically large) linear program. Two-stage problems may be approximated using hundreds or, in special cases, thousands of scenarios, but multi-stage problems are normally restricted to much smaller samples. Once a set of scenarios has been generated, most scenario-based methods treat this sample as representing the entire problem which they then strive to solve to optimality. Examples of algorithms designed for this class of problems includes the diagonal quadratic approximation method of Mulvey and Ruszczynski [15], the augmented Lagrangian decomposition method of Rosa and Ruszczynski [20], the so-called L-shaped method of Van Slyke and Wets [21] and its generalization to multistage problems by Birge [1], and the scenario aggregation method of Rockafellar and Wets [19]. All these algorithms provide optimal solutions to what are normally approximations of the original problem.

Sampling-based methods, on the other hand, explicitly represent the complete sample space (which may, for all practical purposes, be of infinite size). Examples include stochastic linearization methods (Gupal and Bazhenov [10] and Ermoliev [7]), the auxiliary function method (Culioli and Cohen [6]), stochastic decomposition (Higle and Sen [11]), sample path optimization (Robinson [18]) and the stochastic hybrid approximation method (Cheung and Powell [4]). All these methods use successive samples to develop algorithms that converge in some probabilistic sense in the limit. In practical settings, statistical methods have to be used to determine convergence criteria and the properties of the solution after a finite number of iterations (see Higle and Sen [11]).

A popular strategy to counteract the exponential growth of multistage models has been to develop successive approximations of the recourse function. It is well known (see, for example, Wets [22] and Van Slyke and Wets [21]) that the expected recourse function in a two-stage program can be replaced with a series of Benders cuts (where the recourse function is represented using a fixed sample). Birge [1] extends this approach to multistage problems by proposing a nested Benders decomposition algorithm. The basic version of this method involves a forward pass through the time periods, using a specific set of cuts, and then a backward pass, where new cuts are generated. Other authors have studied variations of this strategy (see Wittrock [23], Gassman [9], Infanger [12], and Morton [14]). Infanger and Morton [13] show how the method can be extended to take advantage of interstage dependencies.

Nested Benders decomposition, as it is generally described (see, for example, [13]), requires solving a linear program at each time period, and for each scenario, where a scenario represents a full history of events up to that point in time. Let Ω_t represent the set of outcomes in time period t, and let $h_t = (\omega_1, \omega_2, \ldots, \omega_t)$ represent the history of the process, where $h_t \in \mathcal{H}_t = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_t$ (some authors use the notation Ω_t to represent the history, whereas we use it only to denote events within a time period). Clearly, the size of $\mathcal{H}_t$ grows exponentially with the number of stages, making nested Benders impractical for even medium-sized problems.

In this paper, we propose an new convergent algorithm for multistage stochastic linear programs with recourse that satisfy the following assumptions:

(A1) Random quantities in different stages are independent.

(A2) The sample space of random quantities in each stage is discrete and finite.

(A3) Random quantities only appear on the right-hand side of the linear constraints in each stage.

(A4) The feasible region of the problem in each stage is always nonempty and bounded.

As mentioned earlier, the assumption (A2) is necessary for all scenario-based methods. While, the assumptions (A3) and (A4) are made in many sampling-based methods including the stochastic decomposition method of Higle and Sen [11]. The nested Benders decomposition method of Birge [1] assumes (A2) and (A3). We note that the result we are going to present can be extended, after some refinement, to more general cases including the case where not only right-hand side vectors but also matrices B_t linking neighboring stages are stochastic, and the case where the feasible region of the problem in each stage can be infeasible or unbounded.

Features of our method include:

a) At each iteration, we solve a linear program for a single realization $\omega_t \in \Omega_t$ (as opposed to each $h_t \in \mathcal{H}_t$) at each stage t. As a result, the computational requirements of the procedure per iteration grow linearly with the number of stages, and the size of the sample space per stage.

b) We perform a simple comparison over the entire sample space Ω_t at each stage t. Thus, Ω_t may be large (say, in the tens or even hundreds of thousands) but must be finite.

c) Our method successively approximates the expected recourse function by building up valid cutting planes to support these functions from below.

d) We prove that the algorithm converges in the limit, but do not provide finite convergence.

Because we use only a partial sample (item (a) above) rather than the full sample required by the other methods, we call our method a cutting plane and partial-sampling (CUPPS) algorithm. On the other hand, the full pass over the sample space in item (b) implies the space must be finite, in contrast with true sampling techniques such as stochastic decomposition. Our method is closest to the nested Benders decomposition method of Birge [1] (with much lower computational effort per iteration) and the stochastic decomposition method of Higle and Sen [11].

The research contribution of the paper is the presentation of a new algorithm for solving multistage stochastic programs which is convergent in probability, and which is computationally tractable for problems with large numbers of outcomes per stage, and large numbers of stages. The primary limitation of our method is the limitation that is shared by all cutting-plane algorithms, which is slow convergence when we are approximating high dimensionality problems. Since the relative advantage of CUPPS over classical nested Benders in terms of execution time per iteration is obvious, we do not present any numerical experiments. Our belief is that experimental work must be conducted in the context of a specific application, with an algorithm that is able to take advantage of the structure of that problem.

This paper is organized as follows. In Section 2, we present the core idea of the CUPPS algorithm when applied to a two-stage problem and compare it to the L-shaped algorithm and the stochastic decomposition algorithm. In Section 3, we present the details of the CUPPS algorithm. Then, in Section 4 we give some preliminary results, and in Section 5 we establish the convergence of this algorithm. Finally, we conclude the paper in Section 6.

2 Core Idea and Comparison

In this section, we briefly present the core idea of our CUPPS method when applied to a two-stage problem and compare it to the two closest existing methods: the stochastic decomposition (SD) method of Higle and Sen [11] and the L-shaped (LS) method of Van Slyke and Wets [21] and Birge [1].

We begin by introducing some basic notation:

$(\Omega_t, \mathcal{F}_t, P_t)$ = probability space of the random quantities in stage t, where Ω_t is the sample space of the random quantities (hence by (A2), $|\Omega_t|$ is finite), and $\mathcal{F}_t$ is a σ-algebra and P_t a probability measure defined over Ω_t;

$\Omega_t = \{\omega_{t1}, ..., \omega_{t,q_t}\}$ = the sample space of the random quantities in stage t, where $q_t = |\Omega_t|$, and ω_{ti} is a sample in Ω_t, for all $i = 1, ..., q_t$, with $q_1 = 1$;

p_{ti} = the probability associated with each sample $\omega_{ti} \in \Omega_t$, for all $i = 1, ..., q_t$, such that $\sum_{i=1}^{q_t} p_{ti} = 1$.

$\mathcal{H}_t$ = the σ-field which represents the information available up to stage t,

x_t = vector of decisions in stage t;

c_t = vector of cost in stage t;

A_t = constraint matrix in stage t;

B_t = constraint matrix linking stage t and stage $t+1$;

$Q_t(x_{t-1}, \omega_t) = Q_t(x_{t-1}, \omega_t | \mathcal{H}_{t-1})$ = recourse function in stage $t-1$ given the history $\mathcal{H}_{t-1}$; note that the assumption (A1) guarantees the first equality;

$\bar{Q}_t(x_{t-1}) = \sum_{i=1}^{q_t} p_{ti} Q_t(x_{t-1}, \omega_{ti}) = E_{\Omega_t} Q_t(x_{t-1}, \omega_t) = E_{\Omega_t} Q_t(x_{t-1}, \omega_t | \mathcal{H}_{t-1})$ = expected recourse function in stage $t-1$ given the history $\mathcal{H}_{t-1}$; note that the assumption (A1) guarantees the second equality.

A general T-stage stochastic linear program with recourse can be formulated as follows:

$$\min \quad c_1^T x_1 + E_{\Omega_2} \left[\min c_2^T x_2 + ... + E_{\Omega_T} \left[\min c_T^T x_T \right] ... \right]$$

subject to

$$\begin{array}{lllll} A_1 x_1 & & & & = b_1 \\ B_1 x_1 & + A_2 x_2 & & & = \omega_2 \\ & ... & ... & & ... \\ & & B_{T-1} x_{T-1} & + A_T x_T & = \omega_T \end{array} \tag{1}$$

$x_t \geq 0$ for $t = 1, ..., T$, $\quad \omega_t \in \Omega_t$, for $t = 2, ..., T$,

For the problem of our interest that satisfies the assumption (A1), the formulation (1) can be equivalently rewritten in the following recursive form:

[LP$_1$]

$$Q_1 = \min_{x_1} \quad c_1^T x_1 + \bar{Q}_2(x_1) \tag{2}$$

subject to

$$A_1 x_1 = b_1 \tag{3}$$

$$x_1 \geq 0 \tag{4}$$

where the recourse function is defined by, for $t = 2, ..., T$:

[LP$_t$]

$$Q_t(x_{t-1}, \omega_t) = \min_{x_t} \quad c_t^T x_t + \bar{Q}_{t+1}(x_t) \tag{5}$$

subject to

$$\begin{aligned} A_t x_t &= \omega_t - B_{t-1} x_{t-1} && (6) \\ x_t &\geq 0 && (7) \end{aligned}$$

and $\bar{Q}_{T+1} \equiv 0$

The core idea of the CUPPS method is to successively approximate the expected recourse function in each stage by valid cutting planes that are generated based on a known subset of dual extreme points of the next-stage problem. To be more specific, let us consider the two-stage problem given by (2)-(7) with $T = 2$. For solving this problem, each iteration k of the CUPPS algorithm involves two steps. The first step solves an approximated problem of **[LP$_1$]** which is as follows.

$$\min_{x_1} \quad c_1^T x_1 + z \tag{8}$$

subject to

$$\begin{aligned} A_1 x_1 &= b_1 && (9) \\ z + \beta_i^T x_1 &\geq \alpha_i, \quad \forall \quad i = 1, ..., k && (10) \\ x_1 &\geq 0 && (11) \end{aligned}$$

where (10) represents the k cuts generated so far. These cuts are generated in the second step and approximate the expected recourse function $\bar{Q}_2(x_1)$ by supporting it from below. Note that, initially, the algorithm approximates $\bar{Q}_2(x_1)$ by the first cut that is trivial: $z \geq -\infty$. Let x_1^k denote the solution to the problem (8)-(11).

In the second step, the algorithm first randomly draws a sample, denoted as ω_2^k, from Ω_2, then solves the problem **[LP$_2$]** with $x_1 = x_1^k$ and $\omega_2 = \omega_2^k$. Assumption (A4) guarantees that both optimal primal and dual solutions to this problem can always be found. Let π_k be the dual solution of this problem. Notice that in the problem $[\mathbf{LP}_2]$, x_1 and ω_2 appear only on the right-hand side. Thus, for any given x_1^u and x_1^v with $u \neq v$, and given $\omega_{2i}, \omega_{2j} \in \Omega_2$ with $i \neq j$, any dual extreme point of the problem of $[\mathbf{LP}_2]$ with $x_1 = x_1^u$ and $\omega_2 = \omega_{2i}$ is also a dual extreme point of that with $x_1 = x_1^v$ and $\omega_2 = \omega_{2j}$. Let $\mathcal{D}^k = \{\pi_j \mid j = 1, ..., k\}$ be the set of all dual extreme points of the problem $[\mathbf{LP}_2]$ generated up to iteration k. Based on the dual extreme points in $\mathcal{D}^k$, the algorithm then generates a new cut

$$z + \beta_{k+1}^T x_1 \geq \alpha_{k+1} \tag{12}$$

with the coefficients, scalar α_{k+1} and vector β_{k+1}, given by

$$\alpha_{k+1} - \beta_{k+1}^T x_1 \equiv \sum_{j=1}^{q_2} p_{2j} \left(\tilde{\pi}_j\right)^T \left(\omega_{2j} - B_1 x_1\right) \tag{13}$$

where

$$\tilde{\pi}_j = \operatorname{argmax}\left\{\pi_i^T(\omega_{2j} - B_1 x_1^k) \mid \pi_i \in \mathcal{D}^k\right\} \tag{14}$$

Note that the summation over all possible outcomes in Ω_2 in equation (13), and the maximum operator applied in equation (14), represent, for large problems, the computational bottleneck of the procedure.

As we show in Section 4, the cut generated this way is valid for the expected recourse function $\bar{Q}_2(x_1)$, in that it supports $\bar{Q}_2(x_1)$ from below, but may not be tight, as illustrated in Figure 1.1. The effort for generating this cut involves solving one linear program, that is the problem $[\mathbf{LP}_2]$ with $x_1 = x_1^k$ and $\omega_2 = \omega_2^k$, and $O(kq_2)$ basic operations for computing α_{k+1} and β_{k+1} in (13) and (14). This cut is then added to the problem (8)-(11). The whole procedure is then repeated.

By comparison, the SD algorithm shares similar steps except that it not only generates new cuts but also updates previously generated cuts. The LS algorithm, by contrast, not only solves just one problem $[\mathbf{LP}_2]$ with $x_1 = x_1^k$ and $\omega_2 = \omega_2^k$, but instead it solves every problem $[\mathbf{LP}_2]$ with $x_1 = x_1^k$ and $\omega_2 = \omega_{2j}$ for all $j = 1, ..., q_2$. Then a new cut is generated based on the optimal dual solutions of all these q_2 problems.

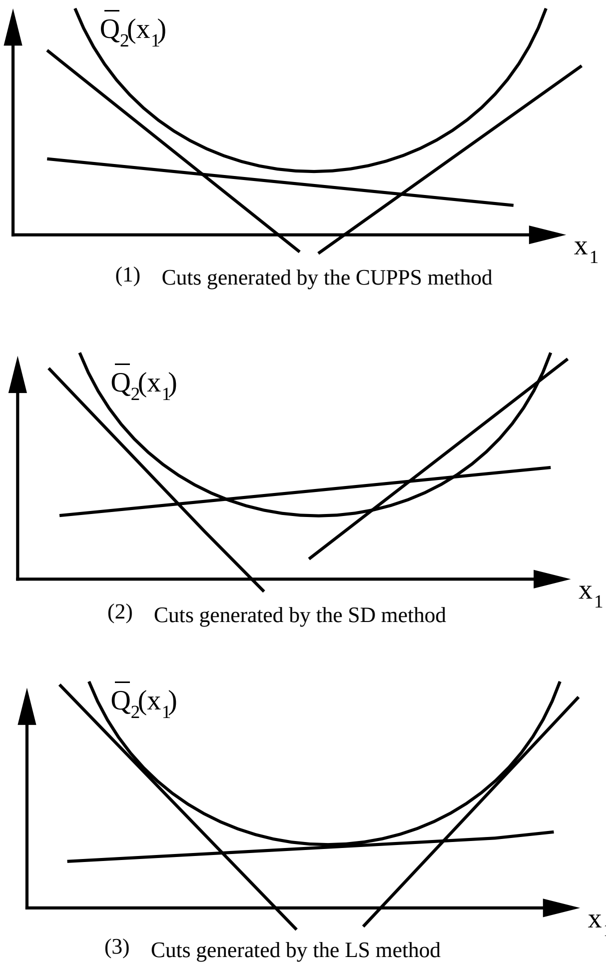

Figure 1: Comparison of cuts generated by the CUPPS, SD and LS methods

More specifically, in the SD algorithm, iteration k solves the problem $[\mathbf{LP}_2]$ with $x_1 = x_1^k$ and $\omega_2 = \omega_2^k$ and generates a new cut (12) with the coefficients α_{k+1} and β_{k+1} given by

$$\alpha_{k+1} - \beta_{k+1}^T x_1 \equiv \sum_{j=1}^{k} p_{2j} \, (\hat{\pi}_j)^T \, (\omega_2^j - B_1 x_1) \tag{15}$$

where

$$\hat{\pi}_j = \text{argmax} \left\{ \pi_i^T (\omega_2^j - B_1 x_1^j) \mid \pi_i \in \mathcal{D}^j \right\} \tag{16}$$

In general, the cut generated this way may not be a valid cut for the expected recourse function $\bar{Q}_2(x_1)$, as illustrated in Figure 1.2, because the coefficients of the cut are computed in (15) using only k samples, instead of all the samples in Ω_2. It is easy to see that the computational effort involved here is solving one linear program and performing $O(k)$ basic operations in (15) and (16) because in iteration k, one only need compute $\hat{\pi}_k$, since all other $\hat{\pi}_j$, with $1 \leq j \leq k-1$, were already computed earlier. Thus, the SD algorithm offers the lowest computational effort per iteration of all three algorithms.

To generate a new cut in iteration k, the LS algorithm solves every problem $[\mathbf{LP}_2]$ with $x_1 = x_1^k$ and $\omega_2 = \omega_{2j}$, for each $j = 1, ..., q_2$. Let u_j^k denote the optimal dual solution, obtained in iteration k, of the problem $[\mathbf{LP}_2]$ with $x_1 = x_1^k$ and $\omega_2 = \omega_{2j}$. Then the coefficients of the new cut (12) generated by the LS algorithm in iteration k are given by

$$\alpha_{k+1} - \beta_{k+1}^T x_1 \equiv \sum_{j=1}^{q_2} p_{2j} \left(u_j^k\right)^T (\omega_{2j} - B_1 x_1) \tag{17}$$

It is easy to see that α_{k+1} and β_{k+1} given by (17) satisfy:

$$\alpha_{k+1} - \beta_{k+1}^T x_1^k = \bar{Q}_2(x_1^k) \tag{18}$$

Furthermore, it is not difficult to see that the cut given by (17) is valid and hence by (18) it is a tight cut for the expected recourse function $\bar{Q}_2(x_1)$ and touches the function at the point x_1^k, as shown in Figure 1.3. The computational effort involved here consists of solving q_2 linear programs and performing $O(q_2)$ basic operations in (17).

From the above comparison, it is quite clear that to generate a cut, the LS algorithm needs the most, while the SD algorithm needs the least, computational effort among these three algorithms. On the other hand, the quality of cuts generated by the LS algorithm is the best in terms of their tightness. Thus, for two-stage problems, the CUPPS algorithm can be viewed as a method lying between the SD algorithm and the LS algorithm. The CUPPS algorithm attempts to build valid cuts, instead of stochastic cuts as in the SD algorithm, by using available dual extreme points that have been generated, instead of solving all the problems associated with the samples as in the LS algorithm.

For multistage problems, the core ideas of the nested Benders decomposition algorithm of Birge [1] and the CUPPS algorithm are similar to their respective counterparts for two-stage problems described above. Hence, we do not compare them here. The details of the CUPPS algorithm is described in the next section. See [1] for the details of the nested Benders decomposition algorithm.

3 The CUPPS Algorithm

In each iteration, the CUPPS algorithm solves an approximated problem of [$\mathbf{LP}_t$], denoted as [$\mathbf{AP}_t$], for each $t = 1, 2, ..., T-1$, and a problem [$\mathbf{LP}_T$]. In the problem [$\mathbf{AP}_t$], the expected recourse function $\bar{Q}_{t+1}(x_t)$ is approximated by some cuts that support it from below. After solving the problem [$\mathbf{AP}_t$] or [$\mathbf{LP}_T$], the algorithm generates a cut that is valid for the expected recourse function $\bar{Q}_t(x_{t-1})$, and adds this cut to the problem [$\mathbf{AP}_{t-1}$]. In the course of the algorithm, the approximated problems [$\mathbf{AP}_t$], for $t = 1, ..., T-1$, approximate the original [$\mathbf{LP}_t$] more and more accurately.

In each iteration, the algorithm generates one cut for each expected recourse function $\bar{Q}_t(x_{t-1})$, for $t = 2, ..., T$. At the very beginning, the algorithm uses the following initial cut to support the function $\bar{Q}_t(x_{t-1})$ from below:

$$z_{t+1} \geq -\infty \tag{19}$$

Certainly, this is a valid cut. Thus, there are a total of $k+1$ cuts in the problem [$\mathbf{AP}_t$] right after iteration k.

Suppose that right after iteration $k-1$, the approximated problems, [$\mathbf{AP}_1$] and [$\mathbf{AP}_t$], for $t = 2, ..., T-1$, are as follows:

[$\mathbf{AP}_1$]:

$$\hat{Q}_1^k = \min_{x_1, z_2} \quad c_1^T x_1 + z_2 \tag{20}$$

subject to

$$\begin{aligned} A_1 x_1 &= b_1 && (21) \\ z_2 + \left(\beta_2^i\right)^T x_1 &\geq \alpha_{2i}, \quad \forall \ i = 1, ..., k && (22) \\ x_1 &\geq 0 && (23) \end{aligned}$$

[$\mathbf{AP}_t$]:

$$\hat{Q}_t^k(x_{t-1}, \omega_t) = \min_{x_t, z_{t+1}} \quad c_t^T x_t + z_{t+1} \tag{24}$$

subject to

$$A_t x_t = \omega_t - B_{t-1} x_{t-1} \tag{25}$$

$$z_{t+1} + \left(\beta_{t+1}^i\right)^T x_t \geq \alpha_{t+1,i}, \quad \forall \quad i = 1, ..., k \tag{26}$$

$$x_t \geq 0 \tag{27}$$

where (26) represents k cuts that have been generated up to iteration $k-1$ for the expected recourse function $\bar{Q}_{t+1}(x_t)$. We describe later how these cuts are generated.

Then in the next iteration, i.e. iteration k, the CUPPS algorithm first solves the problem $[\mathbf{AP}_1]$. Let the primal and dual solutions be denoted, respectively, by (x_1^k, z_2^k), and (π_1^k, ρ_1^k), where π_1^k and ρ_1^k are, respectively, the vectors representing the dual solutions corresponding to (21) and (22). Next, for each $t = 2, ..., T$ in this order, the algorithm first draws a sample, denoted by ω_t^k, from Ω_t, then solves the problem $[\mathbf{AP}_t]$ for $t < T$ (or $[\mathbf{LP}_T]$ for $t = T$) with $x_{t-1} = x_{t-1}^k$ and $\omega_t = \omega_t^k$ and gets the primal and dual solutions denoted, respectively, by (x_t^k, z_{t+1}^k) and (π_t^k, ρ_t^k) (or x_T^k and π_T^k), where π_t^k and ρ_t^k are, respectively, the vectors representing the dual solutions corresponding to (25) (or (6) in the problem $[\mathbf{LP}_T]$) and (26).

Denote, for $t = 2, ..., T$,

$\mathcal{D}_t^k =$ the set of all the dual extreme points generated so far right after iteration k for the problem $[\mathbf{AP}_t]$, for $t < T$, or the problem $[\mathbf{LP}_T]$ for $t = T$.

Then after the problem $[\mathbf{AP}_t]$ (or $[\mathbf{LP}_T]$) is solved in iteration $k-1$, this set of dual extreme points is updated by, for $t = 2, ..., T-1$,

$$\mathcal{D}_t^k = \mathcal{D}_t^{k-1} \bigcup \left\{(\pi_t^k, \rho_t^k)\right\} \tag{28}$$

and for $t = T$,

$$\mathcal{D}_T^k = \mathcal{D}_T^{k-1} \bigcup \left\{\pi_T^k\right\} \tag{29}$$

Note that elements in the set $\mathcal{D}_t^k$ may have different dimensions. Elements generated earlier have smaller dimensions than those generated later. Throughout this paper, whenever we calculate the inner product of or compare two vectors with different dimensions, we assume that the vector with a fewer dimension is extended by attaching zeros to it such that it has the same dimension as the other vector. We show in section 5 that any element in $\mathcal{D}_t^k$ generated earlier than iteration k, if extended accordingly by attaching zeros to it, is still a dual extreme point of the problem $[\mathbf{AP}_t]$ formed later than iteration k.

Based on the dual extreme points in the set $\mathcal{D}_t^k$ that have been generated so far, the algorithm then generates a new cut, i.e. the $(k+1)$-st cut, for the function $\bar{Q}_t(x_{t-1})$. This cut is given by,

$$z_t + \left(\beta_t^{k+1}\right)^T x_{t-1} \geq \alpha_{t,k+1} \tag{30}$$

with

$$\beta_t^{k+1} = \sum_{i=1}^{q_t} p_{ti} B_{t-1}^T \pi_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k), \quad \text{for } 2 \leq t \leq T \tag{31}$$

$$\alpha_{t,k+1} = \sum_{i=1}^{q_t} p_{ti} \left[\omega_{ti}^T \pi_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k) + \left(\alpha_{t+1}^k\right)^T \rho_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k)\right], \quad \text{for } 2 \leq t \leq T-1 \tag{32}$$

$$\alpha_{T,k+1} = \sum_{i=1}^{q_T} p_{Ti} \left(\omega_{Ti}\right)^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k) \tag{33}$$

where, for any $\omega_t \in \Omega_t$, and $2 \leq t \leq T-1$,

$$\left(\pi_t(x_{t-1}^k, \omega_t, \mathcal{D}_t^k), \rho_t(x_{t-1}^k, \omega_t, \mathcal{D}_t^k)\right) = \operatorname{argmax}\left\{\pi_t^T \left(\omega_t - B_{t-1} x_{t-1}^k\right) + \rho_t^T \alpha_{t+1}^k \mid (\pi_t, \rho_t) \in \mathcal{D}_t^k\right\} \tag{34}$$

and for any $\omega_T \in \Omega_T$,

$$\pi_T(x_{T-1}^k, \omega_T, \mathcal{D}_T^k) = \operatorname{argmax}\left\{\pi_T^T \left(\omega_T - B_{T-1} x_{T-1}^k\right) \mid \pi_T \in \mathcal{D}_T^k\right\} \tag{35}$$

and, for $2 \leq t \leq T$, α_t^k is the vector defined as

$$\alpha_t^k = (\alpha_{t,1}, \ldots, \alpha_{t,k}) \tag{36}$$

This new cut (30) is added to the preceding problem $[\mathbf{AP}_{t-1}]$. Then the algorithm moves forward to solve the next problem $[\mathbf{AP}_{t+1}]$. Similarly a new cut is generated and added to the then preceding problem $[\mathbf{AP}_t]$. Finally, the algorithm solves the problem $[\mathbf{LP}_T]$ and generats a new cut and adds this cut to the problem $[\mathbf{AP}_{T-1}]$. This ends iteration k.

Now we are ready to give the details of the CUPPS algorithm.

The CUPPS Algorithm

Step 0: For $t = 1, 2, ..., T-1$, formulate the initial approximated problem $[\mathbf{AP}_t]$ with only one initial cut given by (19). Set the set of dual extreme points $\mathcal{D}_t^1 = \phi$ for all $t = 2, 3, ..., T$. Set iteration counter $k = 1$.

Step 1: Solve the problem $[\mathbf{AP}_1]$. Get the optimal primal solution (x_1^k, z_2^k) and optimal solution value $\hat{Q}_1^k$.

Step 2: For $t = 2, ..., T-1$ in this order, do the following. Draw a random sample ω_t^k from Ω_t. For a given $x_{t-1} = x_{t-1}^k$, and $\omega_t = \omega_t^k$, solve the problem $[\mathbf{AP}_t]$. Get the optimal primal solution (x_t^k, z_{t+1}^k), optimal solution value $\hat{Q}_t^k(x_{t-1}^k, \omega_t^k)$, and the optimal dual solution (π_t^k, ρ_t^k). Get the latest set of dual extreme points $\mathcal{D}_t^k$ by (28). Generate the $(k+1)$-st cut given by (30). Add this cut to the problem $[\mathbf{AP}_{t-1}]$.

Step 3: Draw a random sample ω_T^k from Ω_T. For $x_{T-1} = x_{T-1}^k$, and $\omega_T = \omega_T^k$, solve the problem $[\mathbf{LP}_T]$. Get the optimal primal solution x_T^k, optimal solution value $Q_T(x_{T-1}^k, \omega_T^k)$, and the optimal dual solution π_T^k. Get the latest set of dual extreme points $\mathcal{D}_T^k$ by (29). Generate the $(k+1)$-st cut given by (30) with $t = T$. Add this cut to the problem $[\mathbf{AP}_{T-1}]$.

Step 4: Set $k = k + 1$. Go to Step 1.

4 Preliminary Results

In this section, we prove that all the cuts generated in the CUPPS algorithm for the expected recourse function $\bar{Q}_t(x_{t-1})$, for all $t = 2, ..., T$, are valid, that is, they support $\bar{Q}_t(x_{t-1})$ from below. Also, we give two basic results that are used in section 5.

Define, for any $k \geq 1$, and $1 \leq t \leq T-1$,

$$\hat{\bar{Q}}_t^k(x_{t-1}) = \sum_{i=1}^{q_t} p_{ti} \hat{Q}_t^k(x_{t-1}, \omega_{ti})$$

Lemma 4.1 In the CUPPS algorithm, each cut added to the problem $[\mathbf{AP}_{T-1}]$ is a valid cut for the function $\bar{Q}_T(x_{T-1})$.

Proof: Clearly, in the problem $[\mathbf{AP}_{T-1}]$, the very first cut given by (19) with $t = T-1$ is valid for $\bar{Q}_T(x_{T-1})$. Now consider the k-th cut in $[\mathbf{AP}_{T-1}]$ for any $k \geq 2$. Clearly, for any given x_{T-1} and ω_{Ti}, for any $i = 1, ..., q_T$, $\pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k)$ is a dual feasible solution to the problem $[\mathbf{LP}_T]$ with a given x_{T-1} and $\omega_T = \omega_{Ti}$. This implies that, for any x_{T-1} and $i = 1, ..., q_T$,

$$Q_T(x_{T-1}, \omega_{Ti}) \geq (\omega_{Ti} - B_{T-1}x_{T-1})^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k)$$

First multiplying by p_{Ti}, then summing over all i on the both sides, we have

$$\begin{aligned} \bar{Q}_T(x_{T-1}) &\geq \sum_{i=1}^{q_T} p_{Ti}\,(\omega_{Ti} - B_{T-1}x_{T-1})^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k) \\ &= \sum_{i=1}^{q_T} p_{Ti}(\omega_{Ti})^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k) - \sum_{i=1}^{q_T} p_{Ti} x_{T-1}^T B_{T-1}^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k) \\ &= \alpha_{T,k+1} - \left(\beta_T^{k+1}\right)^T x_{T-1} \end{aligned}$$

This shows that the k-th cut in $[\mathbf{AP}_{T-1}]$ is valid for $\bar{Q}_T(x_{T-1})$. ...

Lemma 4.2 In the CUPPS algorithm, for any t with $1 \leq t \leq T-1$, and any k with $k \geq 1$, every element in the set $\mathcal{D}_t^k$ is a dual extreme point of the problem $[\mathbf{AP}_t]$ formed right after any iteration j with $j \geq k-1$.

Proof: First, it is easy to see that any element (π_t, ρ_t) in the set $\mathcal{D}_t^k$ is a dual extreme point of the problem $[\mathbf{AP}_t]$ generated in some iteration i with $i \leq k$. Let DP1 denote the dual of the problem $[\mathbf{AP}_t]$ formed right after iteration $i-1$. Then, (π_t, ρ_t) is an extreme point of the problem DP1. Let DP2 denote the dual of the problem $[\mathbf{AP}_t]$ formed right after iteration j for any given $j \geq k$. Then in order to prove the lemma, we need only to show that (π_t, ρ_t), if extended by adding a proper number of zeros to its end, is also an extreme point of the problem DP2.

Since the algorithm adds one cut to the problem $[\mathbf{AP}_t]$ in each iteration and once a cut is added it will always be there, the problems DP1 and DP2 are the same except that in DP2 there are $j-i$ more columns. Thus (π_t, ρ_t) is feasible to DP2 if we extend it by adding $j-i$

zeros to its end. Denote this extended vector by (π_t, ρ_t^+), where $(\rho_t^+)_l = (\rho_t)_l$ for $l = 1, ..., i$ and

$$(\rho_t^+)_l = 0, \quad \text{for } l = i+1, ..., j. \tag{37}$$

In the following, we show that (π_t, ρ_t^+) is an extreme point of DP2. We prove it by contradiction. If it is not, then there exist two different solutions of DP2, denoted, (π_t^1, ρ_t^1) and (π_t^2, ρ_t^2), such that

$$(\pi_t, \rho_t^+) = 1/2(\pi_t^1, \rho_t^1) + 1/2(\pi_t^2, \rho_t^2) \tag{38}$$

Clearly, $(\rho_t^1)_l \geq 0$ and $(\rho_t^2)_l \geq 0$ for all $l = 1, ..., j$. Hence, by (37), we have that $(\rho_t^1)_l = (\rho_t^2)_l = 0$ for all $l = i+1, ..., j$. Now define two vectors $\hat{\rho}_t^1$ and $\hat{\rho}_t^2$ with dimension i such that $(\hat{\rho}_t^1)_l = (\rho_t^1)_l$ and $(\hat{\rho}_t^2)_l = (\rho_t^2)_l$ for all $l = 1, ..., i$. Then the fact that $(\pi_t^1, \rho_t^1) \neq (\pi_t^2, \rho_t^2)$ implies that

$$(\pi_t^1, \hat{\rho}_t^1) \neq (\pi_t^2, \hat{\rho}_t^2) \tag{39}$$

and (38) implies that

$$(\pi_t, \rho_t) = 1/2(\pi_t^1, \hat{\rho}_t^1) + 1/2(\pi_t^2, \hat{\rho}_t^2) \tag{40}$$

On the other hand, it is easy to show that both $(\pi_t^1, \hat{\rho}_t^1)$ and $(\pi_t^2, \hat{\rho}_t^2)$ are feasible for DP1. Then (39) and (40) are in contradiction with the fact that (π_t, ρ_t) is an extreme point of DP1. This shows that (π_t, ρ_t^+) must be an extreme point of DP2. ...

Lemma 4.3 In the CUPPS algorithm, for any $\tau = 1, ..., T-1$, each cut added to the problem $[\mathbf{AP}_\tau]$ is a valid cut for the function $\bar{Q}_{\tau+1}(x_\tau)$.

Proof: We prove this by induction on τ. For $\tau = T-1$, this result has been proved in Lemma 4.1. Now assume that for any given $t < T-1$, this result is true for $\tau = t$. We need to prove that this result is also true for $\tau = t-1$. For any given $k \geq 1$, it is easy to see that right after iteration k, the problem $[\mathbf{AP}_t]$ is equivalent to the following problem:

$$\min_{x_t} \left\{ c_t^T x_t + F_{t+1}^k(x_t) \mid (25), (27) \right\} \tag{41}$$

where $F_{t+1}^k(x_t)$ is a piecewise linear function formed by the k cuts in (26). Since, by the induction assumption, these cuts are valid for the function $\bar{Q}_{t+1}(x_t)$, then $F_{t+1}^k(x_t) \le \bar{Q}_{t+1}(x_t)$ for any x_t. On the other hand, the feasible region of the problem (41) is the same as that of the problem $[\mathbf{LP}_t]$. So it must be true that the optimal objective function value of the problem (41) is no more than that of the problem $[\mathbf{LP}_t]$, i.e. for any x_{t-1} and ω_t,

$$\hat{Q}_t^k(x_{t-1}, \omega_t) \le Q_t(x_{t-1}, \omega_t)$$

Taking expectation of both sides, we have

$$\bar{\hat{Q}}_t^k(x_{t-1}) \le \bar{Q}_t(x_{t-1}) \tag{42}$$

On the other hand, by Lemma 4.2, any element in $\mathcal{D}_t^k$ is a dual extreme point of the problem $[\mathbf{AP}_t]$ in iteration k. Thus, for any $\omega_{ti} \in \Omega_t$, with $1 \le i \le q_t$, the vector $\left(\pi_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k), \rho_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k)\right)$ is a dual extreme point of the problem $[\mathbf{AP}_t]$ in iteration k. This means that, for any x_{t-1} and $i = 1, ..., q_t$,

$$\hat{Q}_t^k(x_{t-1}, \omega_{ti}) \ge (\omega_{ti} - B_{t-1}x_{t-1})^T \pi_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k) + \left(\alpha_{t+1}^k\right)^T \rho_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k)$$

First multiplying by p_{ti}, then summing over all i on the both sides, we have

$$\begin{aligned}
\bar{\hat{Q}}_t^k(x_{t-1}) &\ge \sum_{i=1}^{q_t} p_{ti} \left[(\omega_{ti} - B_{t-1}x_{t-1})^T \pi_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k) + \left(\alpha_{t+1}^k\right)^T \rho_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k)\right] \\
&= -\sum_{i=1}^{q_t} p_{ti} \left(\pi_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k)\right)^T B_{t-1}x_{t-1} \\
&\quad + \sum_{i=1}^{q_t} p_{ti} \left[\omega_{ti}^T \pi_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k) + \left(\alpha_{t+1}^k\right)^T \rho_t(x_{t-1}^k, \omega_{ti}, \mathcal{D}_t^k)\right] \\
&= -\left(\beta_t^{k+1}\right) x_{t-1} + \alpha_{t,k+1}
\end{aligned}$$

This, together with (42), gives that

$$\bar{Q}_t(x_{t-1}) \ge \alpha_{t,k+1} - \left(\beta_t^{k+1}\right)^T x_{t-1}$$

This shows that the $(k+1)$-st cut in the problem $[\mathbf{AP}_{t-1}]$, generated right after the problem $[\mathbf{AP}_t]$ is solved in iteration k, is a valid cut for the function $\bar{Q}_t(x_{t-1})$. This shows that the result is true when $\tau = t-1$. Therefore, by induction, we have shown the lemma. ...

Lamma 4.4 The following all hold:

(1) $Q_1 \geq \hat{Q}_1^{k+1} \geq \hat{Q}_1^k$, for any $k \geq 1$;

(2) $Q_t(x_{t-1}, \omega_t) \geq \hat{Q}_t^{k+1}(x_{t-1}, \omega_t) \geq \hat{Q}_t^k(x_{t-1}, \omega_t)$, for any $k \geq 1$, $\omega_t \in \Omega_t$, x_{t-1}, and $2 \leq t \leq T$;

(3) $\bar{Q}_t(x_{t-1}) \geq \hat{\bar{Q}}_t^{k+1}(x_{t-1}) \geq \hat{\bar{Q}}_t^{k}(x_{t-1})$, for any $k \geq 1$, x_{t-1}, and $2 \leq t \leq T$.

Proof: These results are straightforward from Lemmas 4.1 and 4.3. Thus we omit the proofs for them. ...

Remark 4.5 If we run infinitely many iterations of the CUPPS algorithm, then, with probability one, each particular sample ω_{ti}, for any $i = 1, ..., q_t$, and $t = 2, ..., T$, will be drawn infinitely many times.

Proof: In one iteration of the CUPPS algorithm, exactly one sample is drawn from Ω_t, for each $t = 2, ..., T$. For any given t and i with $2 \leq t \leq T$ and $1 \leq i \leq q_t$, the probability that the sample ω_{ti} is drawn out in one iteration is $p_{ti} > 0$. If we run infinitely many iterations of the algorithm, then by the well-known Borel-Cantelli Lemma (see, e.g. Chung [5]), the probability that ω_{ti} is drawn out for infinitely many times is one. This shows the result. ...

Remark 4.6 Given any LP: $\min\{c^T x \mid Ax = b, x \geq 0\}$, define $\mathcal{B}$ to be the set of all possible right-hand side vector b such that there exists an optimal solution to the LP with $b \in \mathcal{B}$. Define a function over the set $\mathcal{B}$, $f(b)$ to be the optimal objective value of the LP with $b \in \mathcal{B}$. Then the function f is continuous in the set $\mathcal{B}$, i.e. for any given $\epsilon > 0$ and $b \in \mathcal{B}$, there is a $\delta > 0$ such that $|f(b) - f(h)| < \epsilon$ for any $h \in \mathcal{B}$ with $||b - h|| < \delta$.

5 Limiting Behavior of the Algorithm

In this section, we analyze the limiting behavior of the CUPPS algorithm and prove that the algorithm is convergent with probability one. First, Lemma 5.1 provides bounds on certain convergent sequences. Then, we prove convergence using a classical inductive proof. Lemma 5.2 demonstrates the convergence of the solution value of the problem [$\mathbf{AP}_{T-1}$] to that of the problem [$\mathbf{LP}_{T-1}$]. Finally, Lemma 5.3 shows that the result is true, by induction, for all remaining stages. The heart of our proof is contained in Lemma 5.2, while the inductive proof in Lemma 5.3 is similar in style to that of Lemma 5.2.

Lemma 5.1 For any given infinite subset $\mathcal{J}$ of $\mathcal{N} = \{1, 2, ...\}$, if the sequence of vectors $\left\{x_{T-1}^k\right\}_{k\in\mathcal{J}}$ converges to some vector x_{T-1}^0, then for any given $\epsilon > 0$,

(a) there exists an integer v_1 such that for any dual extreme point, π_T, of the problem [$\mathbf{LP}_T$],

$$|\pi_T^T B_{T-1}(x_{T-1}^l - x_{T-1}^m)| < \epsilon/6, \quad \text{for any } l, m > v_1 \text{ and } l, m \in \mathcal{J}, \tag{43}$$

(b) there exists an integer v_{2i}, for every $1 \leq i \leq q_T$, such that for any $l, m > v_{2i}$ and $l, m \in \mathcal{J}$:

$$|Q_T(x_{T-1}^m, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})| < \epsilon/12 \tag{44}$$

and furthermore, let $v_2 = \max\{v_{21}, v_{22}, ..., v_{2q_T}\}$, we have for any $l, m > v_2$ and $l, m \in \mathcal{J}$, and $1 \leq i \leq q_T$:

$$|Q_T(x_{T-1}^l, \omega_{Ti}) - Q_T(x_{T-1}^m, \omega_{Ti})| < \epsilon/6 \tag{45}$$

(c) there exists an integer v_3 such that for any $m > v_3$ and $m \in \mathcal{J}$:

$$|Q_{T-1}(x_{T-2}^m, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)| < \epsilon/2 \tag{46}$$

Proof: (a) Assume we are given a dual extreme point, π_T, of the problem [$\mathbf{LP}_T$]. By assumption (A4), we can assume $||\pi_T|| < Y$ for some finite positive number Y. The convergence of the sequence $\left\{x_{T-1}^k\right\}_{k\in\mathcal{J}}$ implies that the sequence $\left\{B_{T-1}x_{T-1}^k\right\}_{k\in\mathcal{J}}$ is also convergent. Thus, for any given $\epsilon > 0$, there exists some integer v_1, such that for any $l, m > v_1$ and $l, m \in \mathcal{J}$, we have $||B_{T-1}x^l_{T-1} - B_{T-1}x^m_{T-1}|| < \epsilon/(6Y)$, which implies that, for all $l, m > v_1$ and $l, m \in \mathcal{J}$,

$$|\pi^T_T B_{T-1}(x^l_{T-1} - x^m_{T-1})| \leq ||\pi_T|| \cdot ||B_{T-1}x^l_{T-1} - B_{T-1}x^m_{T-1}|| < \epsilon/6$$

This shows (a).

(b) By Remark 4.6, the function $Q_T(x_{T-1}, \omega_T)$, for any given ω_T, is continuous in x_{T-1}. For any i with $1 \leq i \leq q_T$, consider the function $Q_T(x_{T-1}, \omega_{Ti})$ at point x^0_{T-1}. By continuity of this function, for any $\epsilon > 0$, there exists $\delta_i > 0$ such that, if $||x_{T-1} - x^0_{T-1}|| < \delta_i$, then $|Q_T(x_{T-1}, \omega_{Ti}) - Q_T(x^0_{T-1}, \omega_{Ti})| < \epsilon/12$. On the other hand, since $\left\{x^k_{T-1}\right\}_{k\in\mathcal{J}}$ is convergent, then for a given δ_i, there exists an integer v_{2i}, such that for any $m > v_{2i}$ and $m \in \mathcal{J}$, we have $||x^m_{T-1} - x^0_{T-1}|| < \delta_i$. All this implies (44), which further implies that, for all $l, m > v_{2i}$ and $l, m \in \mathcal{J}$,

$$\begin{aligned} |Q_T(x^l_{T-1}, \omega_{Ti}) - Q_T(x^m_{T-1}, \omega_{Ti})| &\leq |Q_T(x^l_{T-1}, \omega_{Ti}) - Q_T(x^0_{T-1}, \omega_{Ti})| \\ &\quad + |Q_T(x^0_{T-1}, \omega_{Ti}) - Q_T(x^m_{T-1}, \omega_{Ti})| \\ &< \epsilon/12 + \epsilon/12 = \epsilon/6 \end{aligned}$$

Let $v_2 = \max\{v_{21}, v_{22}, ..., v_{2q_T}\}$. Then we have the result (45). This shows (b).

(c) Similarly, the function $Q_{T-1}(x_{T-2}, \omega^0_{T-1})$ is continuous in x_{T-2}. We can use a similar argument to prove (46). ...

Lemma 5.2 For any given infinite subset $\mathcal{K}$ of $\mathcal{N} = \{1, 2, ...\}$, if

(1) $\omega^k_{T-1} = \omega^0_{T-1}$ for some given $\omega^0_{T-1} \in \Omega_{T-1}$ for any $k \in \mathcal{K}$;

(2) the sequence of vectors $\left\{x^k_{T-2}\right\}_{k\in\mathcal{K}}$ converges to some given vector x^0_{T-2},

then there exists an infinite subset $\mathcal{J}$ of $\mathcal{K}$ such that

(a) the sequence $\left\{x^k_{T-1}\right\}_{k\in\mathcal{J}}$ converges to some vector x^0_{T-1};

(b) the sequence $\left\{\hat{Q}^k_{T-1}(x^k_{T-2}, \omega^0_{T-1})\right\}_{k\in\mathcal{J}}$ converges to $Q_{T-1}(x^0_{T-2}, \omega^0_{T-1})$ with probability one;

(c) the sequence $\left\{z_T^k\right\}_{k\in\mathcal{J}}$ converges to $\bar{Q}_T(x_{T-1}^0)$ with probability one.

Proof: For any given infinite subset $\mathcal{K}$ of $\mathcal{N}$, suppose that the conditions (1) and (2) of the lemma are satisfied.

(a) In iteration $k \in \mathcal{K}$, for the given $x_{T-2} = x_{T-2}^k$ and $\omega_{T-1}^k = \omega_{T-1}^0$, the algorithm solves the problem $[\mathbf{AP}_{T-1}]$, and gets the solution (x_{T-1}^k, z_T^k). By the assumption (A4), the sequence $\left\{x_{T-1}^k\right\}_{k\in\mathcal{K}}$ is bounded. Thus there must exist an infinite subset $\mathcal{J}$ of $\mathcal{K}$ such that the sequence of vectors $\left\{x_{T-1}^k\right\}_{k\in\mathcal{J}}$ converges to some vector x_{T-1}^0. This proves (a).

(b) By Lemma 5.1, for any given $\epsilon > 0$, there exist integers v_1, v_{2i} for each $1 \le i \le q_T$, and v_3 such that (43), (44), (45) and (46) all hold. Define $v_2 = \max\{v_{21}, v_{22}, ..., v_{2q_T}\}$, and $v = \max\{v_1, v_2\}$.

Partition the set $\mathcal{J}$ into q_T subsets $\mathcal{J}_1$, $\mathcal{J}_2$, ..., $\mathcal{J}_{q_T}$ such that for $i = 1, 2, ..., q_T$, $\omega_T^k = \omega_{Ti}$ for all $k \in \mathcal{J}_i$. In other words, $\mathcal{J}_i$ is the set of iteration indices where the sample ω_{Ti} is drawn out for ω_T. Since $\mathcal{J}$ is infinite, by Remark 4.5, in the iterations $k \in \mathcal{J}$, any particular sample $\omega_{Ti} \in \Omega_T$ can be drawn infinitely many times with probability one (simplified as "wp1" in some places below), which means that every set $\mathcal{J}_i$, for $1 \le i \le q_T$, is infinite wp1.

For each $1 \le i \le q_T$, let r_i be the first element in $\mathcal{J}_i$ that is greater than v. Now consider the algorithm in iterations r_1, r_2, ..., r_{q_T}. In iteration r_i, the algorithm solves the problem $[\mathbf{LP}_T]$ with $x_{T-1} = x_{T-1}^{r_i}$ and $\omega_T = \omega_{Ti}$. This generates the optimal dual solution $\pi_T^{r_i}$ of $[\mathbf{LP}_T]$. So, for $1 \le i \le q_T$,

$$Q_T(x_{T-1}^{r_i}, \omega_{Ti}) = (\pi_T^{r_i})^T (\omega_{Ti} - B_{T-1}x_{T-1}^{r_i}) \tag{47}$$

Let r be the first element in $\mathcal{J}$ that is greater than $\max\{r_1, r_2, ..., r_{q_T}\}$. Clearly, before iteration r, the algorithm has already generated the dual extreme point $\pi_T^{r_i}$ satisfying (47) for all $i = 1, 2, ..., q_T$. After the problem $[\mathbf{LP}_T]$ with $x_{T-1} = x_{T-1}^r$ and some ω_T is solved in iteration r, we get the set $\mathcal{D}_T^r$ that contains the dual extreme points of the problem $[\mathbf{LP}_T]$ generated so far. Then the $(r+1)$-st cut for the function $\bar{Q}_T(x_{T-1})$ is generated and added to the problem $[\mathbf{AP}_{T-1}]$. This cut is given by

$$z_T + \left(\beta_T^{r+1}\right)^T x_{T-1} \ge \alpha_{T,r+1} \tag{48}$$

where the vector β_T^{r+1} and the scalar $\alpha_{T,r+1}$ are defined, respectively, in (31) with $t = T$ and $k = r$, and in (33) with $k = r$, that is,

$$\beta_T^{r+1} = \sum_{i=1}^{q_T} p_{Ti} B_{T-1}^T \pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r) \tag{49}$$

$$\alpha_{T,r+1} = \sum_{i=1}^{q_T} p_{Ti} \left(\omega_{Ti}\right)^T \pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r) \tag{50}$$

where $\pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r)$ is defined by

$$\pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r) = \operatorname{argmax}\left\{\pi_T^T \left(\omega_{Ti} - B_{T-1}x_{T-1}^r\right) \mid \pi_T \in \mathcal{D}_T^r\right\} \tag{51}$$

It is easy to see that $\pi_T^{r_i} \in \mathcal{D}_T^r$ for all $i = 1, 2, ..., q_T$. Thus (51) implies that, for each $i = 1, 2, ..., q_T$,

$$\left(\pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r)\right)^T \left(\omega_{Ti} - B_{T-1}x_{T-1}^r\right) \geq \left(\pi_T^{r_i}\right)^T \left(\omega_{Ti} - B_{T-1}x_{T-1}^r\right) \tag{52}$$

Now let $s = \max\{v_3, r\}$. Let N be the first element in $\mathcal{J}$ that is greater than s. Consider any iteration n with $n > N$ and $n \in \mathcal{J}$. In iteration n, the algorithm solves the problem $[\mathbf{AP}_{T-1}]$ with $x_{T-2} = x_{T-2}^n$ and $\omega_{T-1} = \omega_{T-1}^0$, and gets the solution value $\hat{Q}_{T-1}^n(x_{T-2}^n, \omega_{T-1}^0)$, and the solution (x_{T-1}^n, z_T^n). Note that, since $n > r$, in iteration n, the cut (48) is already in the problem $[\mathbf{AP}_{T-1}]$. Hence, the solution (x_{T-1}^n, z_T^n) must satisfy (48), that is

$$\begin{aligned}
z_T^n &\geq \alpha_{T,r+1} - \left(\beta_T^{r+1}\right)^T x_{T-1}^n \\
&= \sum_{i=1}^{q_T} p_{Ti} \left[\left(\pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r)\right)^T \left(\omega_{Ti} - B_{T-1}x_{T-1}^n\right)\right] \\
&= \sum_{i=1}^{q_T} p_{Ti} \left[\left(\pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r)\right)^T \left(\omega_{Ti} - B_{T-1}x_{T-1}^r\right)\right] \\
&\quad + \sum_{i=1}^{q_T} p_{Ti} \left(\pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r)\right)^T B_{T-1}(x_{T-1}^r - x_{T-1}^n) \\
&\geq \sum_{i=1}^{q_T} p_{Ti} \left[\left(\pi_T^{r_i}\right)^T \left(\omega_{Ti} - B_{T-1}x_{T-1}^r\right)\right] \\
&\quad + \sum_{i=1}^{q_T} p_{Ti} \left(\pi_T(x_{T-1}^r, \omega_{Ti}, \mathcal{D}_T^r)\right)^T B_{T-1}(x_{T-1}^r - x_{T-1}^n) \quad \text{(by (52))} \\
&= \sum_{i=1}^{q_T} p_{Ti} \left[\left(\pi_T^{r_i}\right)^T \left(\omega_{Ti} - B_{T-1}x_{T-1}^{r_i}\right)\right]
\end{aligned}$$

$$
\begin{aligned}
& + \sum_{i=1}^{q_T} p_{Ti} \left(\pi_T(x^r_{T-1}, \omega_{Ti}, \mathcal{D}^r_T)\right)^T B_{T-1}(x^r_{T-1} - x^n_{T-1}) \quad \text{(defined as } \Delta_1) \\
& + \sum_{i=1}^{q_T} p_{Ti} \left(\pi^{r_i}_T\right)^T B_{T-1}(x^{r_i}_{T-1} - x^r_{T-1}) \quad \text{(defined as } \Delta_2) \\
= & \sum_{i=1}^{q_T} p_{Ti} Q_T(x^{r_i}_{T-1}, \omega_{Ti}) + \Delta_1 + \Delta_2 \quad \text{(by (47))} \\
= & \sum_{i=1}^{q_T} p_{Ti} Q_T(x^{n}_{T-1}, \omega_{Ti}) + \Delta_1 + \Delta_2 \\
& + \sum_{i=1}^{q_T} p_{Ti} \left[Q_T(x^{r_i}_{T-1}, \omega_{Ti}) - Q_T(x^n_{T-1}, \omega_{Ti})\right] \quad \text{(defined as } \Delta_3) \\
= & \bar{Q}_T(x^n_{T-1}) + \Delta_1 + \Delta_2 + \Delta_3
\end{aligned} \tag{53}
$$

The inequality (53) implies that

$$
\hat{Q}^n_{T-1}(x^n_{T-2}, \omega^0_{T-1}) = c^T_{T-1} x^n_{T-1} + z^n_T \geq c^T_{T-1} x^n_{T-1} + \bar{Q}_T(x^n_{T-1}) + \Delta_1 + \Delta_2 + \Delta_3 \tag{54}
$$

It is easy to see that $x_{T-1} = x^n_{T-1}$ is a feasible solution to the problem $[\mathbf{LP}_{T-1}]$ with $x_{T-2} = x^n_{T-2}$ and $\omega_{T-1} = \omega^0_{T-1}$, which means that

$$
Q_{T-1}(x^n_{T-2}, \omega^0_{T-1}) \leq c^T_{T-1} x^n_{T-1} + \bar{Q}_T(x^n_{T-1}) \tag{55}
$$

Combining (54) and (55), we get

$$
\hat{Q}^n_{T-1}(x^n_{T-2}, \omega^0_{T-1}) \geq Q_{T-1}(x^n_{T-2}, \omega^0_{T-1}) + \Delta_1 + \Delta_2 + \Delta_3 \tag{56}
$$

On the other hand, by Lemma 4.4,

$$
\hat{Q}^n_{T-1}(x^n_{T-2}, \omega^0_{T-1}) \leq Q_{T-1}(x^n_{T-2}, \omega^0_{T-1}) \tag{57}
$$

From (56) and (57), it is not difficult to show that

$$
|\hat{Q}^n_{T-1}(x^n_{T-2}, \omega^0_{T-1}) - Q_{T-1}(x^n_{T-2}, \omega^0_{T-1})| \leq |\Delta_1| + |\Delta_2| + |\Delta_3| \tag{58}
$$

This gives

$$
|\hat{Q}^n_{T-1}(x^n_{T-2}, \omega^0_{T-1}) - Q_{T-1}(x^0_{T-2}, \omega^0_{T-1})|
$$

$$\leq \quad |\hat{Q}^n_{T-1}(x^n_{T-2}, \omega^0_{T-1}) - Q_{T-1}(x^n_{T-2}, \omega^0_{T-1})| + |Q_{T-1}(x^n_{T-2}, \omega^0_{T-1}) - Q_{T-1}(x^0_{T-2}, \omega^0_{T-1})|$$

$$\leq \quad |\Delta_1| + |\Delta_2| + |\Delta_3| + |Q_{T-1}(x^n_{T-2}, \omega^0_{T-1}) - Q_{T-1}(x^0_{T-2}, \omega^0_{T-1})| \tag{59}$$

It is easy to see that $n > r > r_i$ (all $i = 1, 2, ..., q_T$) $> \max\{v_1, v_2\}$. Hence, by (43) and (45), we have

$$|\Delta_1| < \epsilon/6, \quad |\Delta_2| < \epsilon/6, \quad |\Delta_3| < \epsilon/6 \tag{60}$$

Similarly, since $n > v_3$, by (46), we have

$$|Q_{T-1}(x^n_{T-2}, \omega^0_{T-1}) - Q_{T-1}(x^0_{T-2}, \omega^0_{T-1})| < \epsilon/2$$

Thus (59) and (60) imply that

$$|\hat{Q}^n_{T-1}(x^n_{T-2}, \omega^0_{T-1}) - Q_{T-1}(x^0_{T-2}, \omega^0_{T-1})| < \epsilon/2 + \epsilon/6 + \epsilon/6 + \epsilon/6 = \epsilon$$

This means that the sequence $\left\{\hat{Q}^k_{T-1}(x^k_{T-2}, \omega^0_{T-1})\right\}_{k\in\mathcal{J}}$ converges to $Q_{T-1}(x^0_{T-2}, \omega^0_{T-1})$. Since in the proof we have used the result that each $\mathcal{J}_i$ is infinite wp1, this convergence is with probability one. This shows part (b) of the lemma.

(c) The convergence of the sequence $\left\{z^k_T\right\}_{k\in\mathcal{J}}$ can be proved similarly as follows. The relations (54) and (57) imply that

$$c^T_{T-1}x^n_{T-1} + z^n_T \leq Q_{T-1}(x^n_{T-2}, \omega^0_{T-1})$$

Then, by (55), we have

$$z^n_T \leq \bar{Q}_T(x^n_{T-1})$$

This, together with (53), implies that

$$|z^n_T - \bar{Q}_T(x^n_{T-1})| \leq |\Delta_1| + |\Delta_2| + |\Delta_3| \tag{61}$$

By (44), we have

$$\begin{aligned}
|\bar{Q}_T(x_{T-1}^n) - \bar{Q}_T(x_{T-1}^0)| &= |\sum_{i=1}^{q_T} p_{Ti}\left(Q_T(x_{T-1}^n, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})\right)| \\
&\leq \sum_{i=1}^{q_T} p_{Ti}|Q_T(x_{T-1}^n, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})| \\
&\leq \epsilon/12 < \epsilon/2
\end{aligned}$$

This, together with (61) and (60), implies that

$$\begin{aligned}
|z_T^n - \bar{Q}_T(x_{T-1}^0)| &\leq |z_T^n - \bar{Q}_T(x_{T-1}^n)| + |\bar{Q}_T(x_{T-1}^n) - \bar{Q}_T(x_{T-1}^0)| \\
&< |\Delta_1| + |\Delta_2| + |\Delta_3| + \epsilon/2 < \epsilon
\end{aligned}$$

Hence, the sequence $\left\{z_T^k\right\}_{k\in\mathcal{J}}$ converges to $\bar{Q}_T(x_{T-1}^0)$ wp1. This shows part (c) of the lemma. ...

We now demonstrate convergence of all other stages.

Lemma 5.3 For any given τ, $1 \leq \tau \leq T-2$, and any given infinite subset $\mathcal{K}$ of $\mathcal{N} = \{1, 2, ...\}$, if

(1) $\omega_\tau^k = \omega_\tau^0$ for some given $\omega_\tau^0 \in \Omega_\tau$, for any $k \in \mathcal{K}$;

(2) the sequence of vectors $\left\{x_{\tau-1}^k\right\}_{k\in\mathcal{K}}$ converges to some given vector $x_{\tau-1}^0$,

then there exists an infinite subset $\mathcal{J}$ of $\mathcal{K}$ such that

(a) the sequence $\left\{x_\tau^k\right\}_{k\in\mathcal{J}}$ converges to some vector x_τ^0;

(b) the sequence $\left\{\hat{Q}_\tau^k(x_{\tau-1}^k, \omega_\tau^0)\right\}_{k\in\mathcal{J}}$ converges to $Q_\tau(x_{\tau-1}^0, \omega_\tau^0)$ with probability one;

(c) the sequence $\left\{z_{\tau+1}^k\right\}_{k\in\mathcal{J}}$ converges to $\bar{Q}_{\tau+1}(x_\tau^0)$ with probability one.

Proof: We prove the lemma by induction on τ. When $\tau = T-1$, this lemma is exactly Lemma 5.1 and hence holds. Suppose that this lemma holds when $\tau = t$. We need to prove that it also holds when $\tau = t-1$. The proof technique is similar to that of Lemma 5.2. Thus we only provide a sketch of the proof here.

For any given infinite subset $\mathcal{K}$ of $\mathcal{N}$, suppose that when $\tau = t-1$, conditions (1) and (2) of the lemma are satisfied.

(a) In iteration $k \in \mathcal{K}$, the algorithm solves the problem $[\mathbf{AP}_{t-1}]$ with $x_{t-2} = x_{t-2}^k$ and $\omega_{t-1} = \omega_{t-1}^0$, and gets the solution (x_{t-1}^k, z_t^k). By the assumption (A4), the sequence $\left\{x_{t-1}^k\right\}_{k\in\mathcal{K}}$ is bounded. Thus, there must exist an infinite subset $\mathcal{L}$ of $\mathcal{K}$ such that the sequence of vectors $\left\{x_{t-1}^k\right\}_{k\in\mathcal{L}}$ converges to some vector x_{t-1}^0.

Partition the set $\mathcal{L}$ into q_t subsets $\mathcal{L}_1$, $\mathcal{L}_2$, ..., $\mathcal{L}_{q_t}$ such that for $i = 1, 2, ..., q_t$, $\omega_t^k = \omega_{ti}$ for all $k \in \mathcal{L}_i$. In other words, $\mathcal{L}_i$ is the set of iteration indices where the sample ω_{ti} is drawn out for ω_t. By Remark 5.1, it is easy to see that every set $\mathcal{L}_i$, for $1 \le i \le q_t$, is infinite wp1.

For each $i = 1, 2, ..., q_t$, by the induction assumption that the lemma holds for $\tau = t$ and by the fact that $\omega_t^k = \omega_{ti}$ for all $k \in \mathcal{L}_i$, and that the sequence $\left\{x_{t-1}^k\right\}_{k\in\mathcal{L}}$ (and hence the sequence $\left\{x_{t-1}^k\right\}_{k\in\mathcal{L}_i}$) converges to some vector x_{t-1}^0, there must exist a subset $\mathcal{J}_i$ of $\mathcal{L}_i$, for each $i = 1, 2, ..., q_t$, such that the sequence $\left\{\hat{Q}_t^k(x_{t-1}^k, \omega_{ti})\right\}_{k\in\mathcal{J}_i}$ converges to $Q_t(x_{t-1}^0, \omega_{ti})$, wp1. Thus, it is easy to show that for any given $\epsilon > 0$, there exists an integer u_i, such that, wp1,

$$|\hat{Q}_t^m(x_{t-1}^m, \omega_{ti}) - Q_t(x_{t-1}^0, \omega_{ti})| < \epsilon/12, \quad \text{for all } m > u_i \text{ and } m \in \mathcal{J}_i \tag{62}$$

Define $\mathcal{J} = \bigcup_{i=1}^{q_t} \mathcal{J}_i$. Clearly, $\mathcal{J} \subseteq \mathcal{L}$. Hence, the sequence $\left\{x_{t-1}^k\right\}_{k\in\mathcal{J}}$ converges to x_{t-1}^0. This shows that part (a) of the lemma holds when $\tau = t-1$.

(b) First, using (62) and the logic similar to that in the proof of Lemma 5.1, we can get the following results. For any given $\epsilon > 0$,

(i) there exists an integer v_1 such that for any dual extreme point, (π_t, ρ_t), of the problem $[\mathbf{AP}_t]$,

$$|\pi_t^T B_{t-1}(x_{t-1}^l - x_{t-1}^m)| < \epsilon/6, \quad \text{for any } l, m > v_1 \text{ and } l, m \in \mathcal{J} \tag{63}$$

(ii) there exists an integer v_{2i}, for every $1 \le i \le q_t$, such that for

$$|Q_t(x_{t-1}^m, \omega_{ti}) - Q_t(x_{t-1}^0, \omega_{ti})| < \epsilon/12, \quad \text{for any } l, m > v_{2i} \text{ and } l, m \in \mathcal{J}, \tag{64}$$

and furthermore, let $v_2 = \max\{u_1, u_2, ..., u_{q_t}, v_{21}, v_{22}, ..., v_{2q_t}\}$, we have

$$|\hat{Q}_t^l(x_{t-1}^l, \omega_{ti}) - Q_t(x_{t-1}^m, \omega_{ti})| < \epsilon/6, \text{ for any } l, m > v_2 \text{ and } l, m \in \mathcal{J}, \text{ and any } 1 \le i \le q_t, \tag{65}$$

(iii) there exists an integer v_3 such that

$$|Q_{t-1}(x_{t-2}^m, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| < \epsilon/2, \quad \text{for any } m > v_3 \text{ and } m \in \mathcal{J}. \tag{66}$$

Let us define v, r_i (for each $1 \le i \le q_t$), r, s, and N exactly the same way as in the proof of Lemma 5.2. Consider the algorithm in iterations r_i and r. Similar to (47) and (52), the following relations hold for each $i = 1, 2, ..., q_t$,

$$\hat{Q}_t^{r_i}(x_{t-1}^{r_i}, \omega_{ti}) = (\pi_t^{r_i})^T (\omega_{ti} - B_{t-1}x_{t-1}^{r_i}) + (\rho_t^{r_i})^T \alpha_{t+1}^{r_i} \tag{67}$$

and

$$\begin{aligned} &(\pi_t(x_{t-1}^r, \omega_{ti}, \mathcal{D}_t^r))^T (\omega_{ti} - B_{t-1}x_{t-1}^r) + (\rho_t(x_{t-1}^r, \omega_{ti}, \mathcal{D}_t^r))^T \alpha_{t+1}^r \\ &\ge (\pi_t^{r_i})^T (\omega_{ti} - B_{t-1}x_{t-1}^r) + (\rho_t^{r_i})^T \alpha_{t+1}^r \end{aligned} \tag{68}$$

Now consider any iteration n with $n > N$ and $n \in \mathcal{J}$. In iteration n, the algorithm solves the problem $[\mathbf{AP}_{t-1}]$ with given $x_{t-2} = x_{t-2}^n$ and $\omega_{t-1} = \omega_{t-1}^0$, and gets the solution value $\hat{Q}_{t-1}^n(x_{t-2}^n, \omega_{t-1}^0)$, and the solution (x_{t-1}^n, z_t^n). Since $n > r$, in iteration n, the $(r+1)$-st cut (with coefficients β_t^{r+1} and $\alpha_{t,r+1}$ defined by (31) and (32) respectively) is already in the problem $[\mathbf{AP}_{t-1}]$. Hence, the solution (x_{t-1}^n, z_t^n) must satisfy that cut, that is

$$\begin{aligned} z_t^n &\ge \alpha_{t,r+1} - \left(\beta_t^{r+1}\right)^T x_{t-1}^n \\ &= \sum_{i=1}^{q_t} p_{ti} \left[(\omega_{ti} - B_{t-1}x_{t-1}^n)^T \pi_t(x_{t-1}^r, \omega_{ti}, \mathcal{D}_t^r) + (\alpha_{t+1}^r)^T \rho_t(x_{t-1}^r, \omega_{ti}, \mathcal{D}_t^r) \right] \end{aligned}$$

By (67) and (68), we can get a similar result to (53) as follows

$$z_t^n \ge \bar{Q}_t(x_{t-1}^n) + \Delta_1 + \Delta_2 + \Delta_3 \tag{69}$$

where

$$\Delta_1 = \sum_{i=1}^{q_t} p_{ti} \left(\pi_t(x_{t-1}^r, \omega_{ti}, \mathcal{D}_t^r)\right)^T B_{t-1}(x_{t-1}^r - x_{t-1}^n)$$

$$\Delta_2 = \sum_{i=1}^{q_t} p_{ti} \left(\pi_t^{r_i}\right)^T B_{t-1}(x_{t-1}^{r_i} - x_{t-1}^r)$$

$$\Delta_3 = \sum_{i=1}^{q_t} p_{ti} \left[\hat{Q}_t^{r_i}(x_{t-1}^{r_i}, \omega_{ti}) - Q_t(x_{t-1}^n, \omega_{ti})\right]$$

Using the same argument as in the proof of Lemma 5.2, we can then show the following result that is similar to (59),

$$|\hat{Q}_{t-1}^n(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| \le |Q_{t-1}(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| + |\Delta_1| + |\Delta_2| + |\Delta_3| \quad (70)$$

By (63) and (65), we have

$$|\Delta_1| < \epsilon/6, \quad |\Delta_2| < \epsilon/6, \quad |\Delta_3| < \epsilon/6 \quad (71)$$

Similarly, by (66), we have

$$|Q_{t-1}(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{T-1}^0)| < \epsilon/2$$

Thus, (70) implies that

$$|\hat{Q}_{t-1}^n(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| < \epsilon$$

This means that the sequence $\left\{\hat{Q}_{t-1}^k(x_{t-2}^k, \omega_{t-1}^0)\right\}_{k \in \mathcal{J}}$ converges to $Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)$. Since in the proof we have used some results that are true wp1, thus this convergence is with probability one. This shows that part (b) of the lemma holds when $\tau = t - 1$.

(c) The convergence of the sequence $\left\{z_t^k\right\}_{k \in \mathcal{J}}$ can be proved similarly to part (c) of Lemma 5.2. We do not give any details here.

Therefore, by induction, we have proved the lemma. ...

Theorem 5.4 The sequence of the solution values $\left\{\hat{Q}_1^k\right\}_{k\in\mathcal{N}}$ of the problem $[\mathbf{AP}_1]$ converges to the optimal value Q_1, wp1.

Proof: In the approximated problem $[\mathbf{AP}_1]$, we can view the constraint "$A_1x_1 = b_1$" as "$A_1x_1 = \omega_1 - B_0x_0$" with $\omega_1 \equiv b_0$, $x_0 \equiv 0$, and any given B_0, and view the value $\hat{Q}_1^k$ as the function $\hat{Q}_1^k(x_0, \omega_1)$. Thus, when $\tau = 1$ and $\mathcal{K} = \mathcal{N}$, the conditions (1) and (2) of Lemma 5.3 are certainly satisfied. Applying Lemma 5.3, we have that there exists an infinite subset $\mathcal{J}$ of $\mathcal{N}$, such that the sequence $\left\{\hat{Q}_1^k\right\}_{k\in\mathcal{J}}$ converges to $Q_1(b_0, 0) \equiv Q_1$, wp1.

On the other hand, by Lemma 4.4, the sequence $\left\{\hat{Q}_1^k\right\}_{k\in\mathcal{N}}$ is nondecreasing. We know that if a monotone sequence has a convergent subsequence that converges to some value, then the whole sequence must converge to that value. Therefore, the sequence $\left\{\hat{Q}_1^k\right\}_{k\in\mathcal{N}}$ converges to Q_1, wp1. ...

Theorem 5.5 Any accumulation point of the sequence $\left\{x_1^k\right\}_{k\in\mathcal{N}}$ is, wp1, an optimal solution of the problem $[\mathbf{LP}_1]$.

Proof: To prove this, we need only to show that any convergent subsequence of the sequence $\left\{x_1^k\right\}_{k\in\mathcal{N}}$ converges, wp1, to an optimal solution of the problem $[\mathbf{LP}_1]$.

Consider a subsequence $\mathcal{K}$ of $\mathcal{N}$ such that the sequence $\left\{x_1^k\right\}_{k\in\mathcal{K}}$ converges to some vector x_1^0. With the identification of $b_0 = \omega_1 - B_0x_0$ as in the proof of Theorem 5.4, applying Lemma 5.3, we can show that there exists a subsequence $\mathcal{J}$ of $\mathcal{K}$, such that the sequence $\left\{z_2^k\right\}_{k\in\mathcal{J}}$ converges to $\bar{Q}_2(x_1^0)$ wp1.

On the other hand, we know that $\hat{Q}_1^k = c_1^T x_1^k + z_2^k$ for all $k \in \mathcal{N}$. Thus,

$$z_2^k = \hat{Q}_1^k - c_1^T x_1^k, \quad \text{for all } k \in \mathcal{N} \tag{72}$$

Now in the set $\mathcal{K}$, we take the limit on both sides of (72). Since the sequence $\left\{x_1^k\right\}_{k\in\mathcal{K}}$ converges to x_1^0, and by Theorem 5.4, the sequence $\left\{\hat{Q}_1^k\right\}_{k\in\mathcal{K}}$ converges to Q_1 wp1, then the sequence $\left\{z_2^k\right\}_{k\in\mathcal{K}}$ converges to $Q_1 - c_1^T x_1^0$, wp1. We know that if a convergent sequence has a subsequence that converges to some value, then the whole sequence converges to that value, so the following must be true

$$\bar{Q}_2(x_1^0) = Q_1 - c_1^T x_1^0$$

Hence

$$Q_1 = c_1^T x_1^0 + \bar{Q}_2(x_1^0) \tag{73}$$

Since, for any $k \in \mathcal{K}$, x_1^k is a feasible solution to the problem [$\mathbf{LP}_1$], then by assumption (A4), the limit of the sequence $\left\{x_1^k\right\}_{k\in\mathcal{K}}$, x_1^0, must be a feasible solution to the problem [$\mathbf{LP}_1$]. Therefore, (73) implies that the solution x_1^0 is actually optimal to the problem [$\mathbf{LP}_1$]. This shows the theorem. ...

6 Conclusions

In this paper, we have proposed the CUPPS algorithm, a sampling-based algorithm, for solving multistage stochastic linear programs. We have proved that the algorithm is convergent with probability one.

We believe that multistage stochastic linear programs are much harder than two-stage ones. It is unlikely that a scenario-based algorithm is capable of solving a multistage problem with the sample space in each stage containing 1000 samples. For such a problem with T as small as 3, there are 10^6 scenarios. The standard methods, such as the diagonal quadratic approximation method of Mulvey and Ruszczynski [15], and the augmented Lagrangian decomposition method of Rosa and Ruszczynski [20], that reformulate the stochastic problems as a deterministic equivalence, are certainly incapable of dealing with such a problem. We also doubt that the nested Benders decomposition algorithm of Birge [1] can handle such a problem because one iteration alone involves solving at least 3000 linear programs.

Unfortunately, these algorithms do not have an obvious version of approximation schemes inherent in them. By contrast, we can stop the CUPPS algorithm after a desired number of iterations and obtain an approximate solution.

Acknowledgement

This research was supported in part by grant AFOSR-F49620-93-1-0098 from the Air Force Office of Scientific Research. The authors are especially grateful for the comments of one referee who helped refine the presentation.

References

[1] J.R. Birge. Decomposition and partitioning techniques for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007, 1985.

[2] J.R. Birge. Stochastic programming computation and applications. *INFORMS Journal on Computing*, 9(2):111–133, 1997.

[3] J.R. Birge and J.M. Mulvey. Stochastic programming in industrial engineering. Technical report, Department of Industrial and Operations Engineering, The University of Michigan Mathematical Programming, 1996.

[4] R. K.-M. Cheung and W.B. Powell. A stochastic hybrid approximation procedure, with an application to dynamic networks. Technical Report SOR-94-02, Department of Civil Engineering and Operations Research, Princeton University, 1994.

[5] K.L. Chung. *A Course in Probability Theory.* Academic Press, 1974.

[6] J.-C. Culioli and G. Cohen. Decomposition/coordination algorithms in stochastic optimization. *SIAM Journal of Control and Optimization*, 28:1372–1403, November 1990.

[7] Y. Ermoliev. Stochastic quasigradient methods and their application to system optimization. *Stochastics*, 9:1–36, 1983.

[8] L.F. Escudero, P.V. Kamesam, A.J. King, and R. J.-B. Wets. Production planning via scenario modeling. *Annals of Operations Research*, 43:311–335, 1993.

[9] H.I. Gassman. Mslip: A computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423, 1990.

[10] A. M. Gupal and L. G. Bazhenov. A stochastic method of linearization. *Cybernetics*, pages 482–484, 1972.

[11] J.L. Higle and S. Sen. Stochastic decomposition: An algorithm for two stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669, 1991.

[12] G. Infanger. *Planning under Uncertainty: Solving Large-scale Stochastic Linear Programs.* The Scientific Press Series, Boyd & Fraser, New York, 1994.

[13] G. Infanger and D.P. Morton. Cut sharing for multistage stochastic linear programs with interstate dependency. *Mathematical Programming*, 75:241–256, 1996.

[14] D.P. Morton. An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling. *Annals of Operations Research*, 64:211–235, 1996.

[15] J. M. Mulvey and A. Ruszczynski. A diagonal quadratic approximation method for large scale linear programs. *Operations Research Letters*, 12:205–215, 1991.

[16] J. M. Mulvey and H. Vladimirou. Stochastic network programming for financial planning problems. *Management Science*, 38(8):1642–1664, 1992.

[17] W. B. Powell, Patrick Jaillet, and Amedeo Odoni. Stochastic and dynamic networks and routing. In C. Monma, T. Magnanti, and M. Ball, editors, Handbook in Operations Research and Management Science, *Volume on* Networks, pages 141–295. North Holland, 1995.

[18] S. M. Robinson. Analysis of sample path optimization. *Mathematics of Operations Research*, 21(3):1–528, 1996.

[19] T.R. Rockafellar and R.J. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[20] C. Rosa and A. Ruszczynski. On augmented lagrangian decomposition methods for multistage stochastic programs. *Annals of Operations Research*, 64:289–309, 1996.

[21] R. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics*, 17:638–663, 1969.

[22] R. Wets. Programming under uncertainty: The solution set. *SIAM Journal of Applied Mathematics*, 14:1143–1151, 1966.

[23] R.J. Wittrock. Advances in a nested decomposition algorithm for solving staircase linear programs. Technical Report SOL-83-2, Stanford University, 1983.