OPERATIONS RESEARCH CENTER

working paper

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Successive Discretization Procedures for Stochastic Programming with Recourse

by Randall Hiller

OR 146-85 December 1985

Abstract

We address the question of determining good discretizations for stochastic programming problems with recourse. We consider problems in which the second-stage RHS is defined by a vector of independent continuous random variables. Though we consider the problem from a practical viewpoint, some interesting theoretical results come out of the analysis. Two algorithms are described. The first provides a provably good discretization, but is difficult to implement. The second is easily implemented, but may not provide as good a discretization. It is of practical significance that solving the expected value problem is the first step in the discretization procedure. In practice, that is usually the last step. Thus, the results in this paper extend current practice in a simple and natural way.

Introduction

In this paper, we consider the two-stage stochastic programming problem with fixed recourse and random right-hand side. Walkup and Wets [12] showed that if one restricts the stochastic structure of such problems to discrete random variables, there exists a well defined deterministic problem which solves to the same optimal solution and has the same set of optimal first-stage decisions. In this paper, we consider the stochastic programming problem with continuous random variables. We will derive a procedure for approximating the continuous problem by a sequence of discrete problems which can be solved using any of the well-known stochastic programming algorithms (see [2], [4], [5], [11], or, for an excellent overview, [14]).

Olsen [9] showed that general multistage stochastic programming problems containing continuous random variables could be approximated by appropriately chosen problems with discrete random variables. Furthermore, he proved that the sequence of solutions of these discrete approximations would converge to the solution of the continuous problem under some relatively loose assumptions. The question we address in this paper is how best to refine the partition for the simple class of two-stage problems with random right-hand sides. We will develop simple criteria for selecting among a class of discretizations such that the sequence of solutions converge to the optimal solution quickly.

In practice, the expected value problem, that is, the problem with all random elements replaced with their expected value, is generally solved in place of the stochastic programming problem. This approximation is considered necessary since the latter is large and difficult to solve when all the distributions are discrete and is completely intractable when continuous distributions are present. However, researchers and practitioners are becoming aware of the dangers of this single simplistic approximation. Theoretically, one can invent problems in which the errors introduced by this approximation can be arbitrarily large. Furthermore, practical problems have been solved in which the errors have been large enough to justify the additional effort required to solve the stochastic programming problem (e.g. [7]). Though one can determine a bound on the error introduced by solving the expected value problem in place of the stochastic problem (see [1], [3], [6], [8]), it is difficult to get tight bounds. Thus, for most practical applications, one is unable to determine just how bad the expected value approximation is.

We present a method for solving the stochastic programming problem by starting with the solution to the expected value problem. The expected value problem is refined and resolved, and then further refined and resolved, until the error tolerance is acceptable. The analyst must determine the appropriate trade-off between the accuracy of the solution and the cost of its resolution.

By applying the algorithms described in this paper, one not only obtains better approximations to the problem, but also gains insight into the stability of the first stage solution by examining its damped oscillitory behavior from the solution of one discretization to the solution of the next.

Each discretization serves to balance the overall approximation in the following sense. We call a discrete approximation to the stochastic problem balanced if the approximations derived by aggregating any adjacent pair of realizations of the discrete random variables a single realization all have nearly the same optimal value. The algorithms derived in this paper balance the approximation since at each iteration, the next discretization is defined by disaggregating the most out-of-balance element of the problem.

We propose two methods for solving the successive discretization problem. The first method is difficult to apply, but provides a provably good discretization. The second method is extremely easy to apply, but will in general exhibit slower convergence properties than the first. It is our belief that the time/performance tradeoff will generally suggest using the latter method. However, this conjecture has yet to be tested emperically.

The paper is presented in 5 sections. In the first section, we define the problem and the notation. Then, in section 2, we derive the partition selection problem which is the central construct of our analysis. In the third section, we analyze the problem and present our first algorithmic solution procedure. We derive an algorithm for heuristically solving the problem in the fourth section. The algorithm in section 4 is the one that we believe has the most merit from a practical point of view. It is easily implemented and should provide substantial improvement over the current practices of solving the expected value problem or, at best, eyeballing the discretization. Finally, in section 5, we provide a simple numerical example and discuss how the algorithms presented in this paper may be embedded in fast solution procedures for solving larger problems

Section 1: The Problem

The stochastic programming problem with fixed recourse and random right-hand sides can be written as follows:

$$Min\ C_x x + E[C_y y] \qquad (1.1)$$

$$(P) \qquad s.t.\ A_x x \quad = b \qquad (1.2)$$

$$A_y x + B y \ = \xi \qquad (1.3)$$

$$x \geq 0,\ y \geq 0$$

In the continuous stochastic programming problem we wish to analyze, ξ is a vector of continuous random variables, $\xi = \{\xi^1, \xi^2, \ldots, \xi^m\}$, with distribution $F(\xi)$ and probability density function $f(\xi)$ both defined on the support of ξ, denoted by Ξ. We make the following assumptions about the distribution of ξ.

Assumption 1: ξ^i and ξ^j are stochastically independent when $i \neq j$. Thus, the multivariate distribution for ξ is

$$F(\xi) = F_1(\xi^1) \cdot\ F_2(\xi^2) \cdot\ \ldots \cdot\ F_n(\xi^m) \qquad (1.4)$$

and the support of ξ can be written as

$$\Xi = (\Xi^1 \times \Xi^2 \times \ldots \times \Xi^m) \qquad (1.5)$$

Assumption 2: The support of ξ is bounded, ie.$\Xi_i = (\lambda_l^i, \lambda_u^i)$, and $-\propto < \lambda_l^i \leqq \lambda_u^i < +\propto$.

Assumption 3: $f_i(\lambda) > 0$ for all $\lambda_l^i \leqq \lambda \leqq \lambda_u^i$.

Assumption 4: $f_i(\cdot)$ is continuous on $(\lambda_l^i, \lambda_u^i)$.

The approach usually taken in practice to solve (P) is to solve the expected value problem, (EV), defined by replacing ξ in (1.3) by the expected value of ξ, defined as ξ_0.

$$\xi_0^i = \int_{\lambda_l^i}^{\lambda_u^i} \xi dF_i(\xi)$$

with this substitution, the expected value approximation to (P) becomes a deterministic LP.

$$Min\ C_x x + C_y y \tag{1.6}$$

$$(EV) \qquad s.t.\ A_x x \qquad = b \qquad : \pi \tag{1.7}$$

$$A_y x + B y \ = \xi_0 \qquad : \gamma \tag{1.8}$$

$$x \geq 0,\ y \geq 0$$

Problem (EV) can be seen as a trivial discretized approximation to problem (P). The discretization is defined by replacing each random variable by a single element defined as its expected value, and weighting this point by a probability of 1. We wish to refine the trivial discretization of (EV) in an intellegent way.

Before we proceed, we must define what we mean by refining a discretization. Toward this end, we use Olsen's definition of a finite grid [9]:

Definition 1.1: $[\Xi^1;B^1]$ is called a **finite grid** for (P) if Ξ^1 is a finite subset of Ξ, and B^1 is a multifunction, $B^1: \Xi^1 - \blacktriangleright \Xi$, where:

a) $\xi_i \in B^1(\xi_i)$ and $B^1(\xi_i) \in \mathcal{B}(\Xi)$, for all $\xi_i \in \Xi^1$, where $\mathcal{B}(\Xi)$ is the underlying σ-algebra of Borel sets in Ξ.

b) $B^1(\Xi^1) = \Xi$;

c) $B^1(\xi_i) \cap B^1(\xi_j) = \varnothing$ if $\xi_i \neq \xi_j$.

Using this notation, the finite grid $[\Xi^0;B^0]$ which defines (EV) has $\Xi^0 = \{\xi_0\}$, and $B^1(\xi_0) = \Xi$. We shall **refine** the trivial grid, $[\Xi^0;B^0]$, to the grid $[\Xi^1;B^1]$ by partitioning Ξ into two subsets, Ω_1 and Ω_2, each with positive measure. $\Xi^1 = \{\xi_1, \xi_2\}$, where

$$\xi_1 = E\{\xi | \xi \in \Omega_1\} = \int_{\Omega_1} \xi \, dF(\xi)$$

and

$$\xi_2 = E\{\xi | \xi \in \Omega_2\} = \int_{\Omega_2} \xi \, dF(\xi)$$

The deterministic equivalent of the refined stochastic programming problem becomes:

$$(P_1) \qquad \begin{aligned} &Min\ C_x x + C_y(p_1 y_1 + p_2 y_2) && (1.9) \\ &s.t.\ A_x x \qquad\qquad\qquad\quad = b && (1.10) \\ &A_y x \quad + B y_1 \qquad\quad = \xi_1 && (1.11) \\ &A_y x \qquad\qquad + B y_2 = \xi_2 && (1.12) \\ &x \geq 0,\ y \geq 0 \end{aligned}$$

In order to preserve the independence of the stochastic right-hand side elements in our refined problem so that we may reapply the refinement procedure, we must define our partition sets, Ω_1 and Ω_2, carefully. In general, the only way we may partition Ξ into two subsets while preserving independence is by partitioning across a single element of Ξ. This specific type of partition may be uniquely defined by a **partition index**, r, and a **partition cut**, λ, as follows:

$$\Omega_1 = \Xi^1 \times \Xi^2 \times \ldots \times \Xi^{r-1} \times [\lambda_l^r, \lambda] \times \Xi^{r+1} \times \ldots \times \Xi^m \qquad (1.13)$$

$$\Omega_2 = \Xi^1 \times \Xi^2 \times \ldots \times \Xi^{r-1} \times (\lambda, \lambda_u^r] \times \Xi^{r+1} \times \ldots \times \Xi^m \qquad (1.14)$$

Define

$$p_1(r,\lambda) = Pr(\xi \in \Omega_1) = F_r(\lambda), \quad and \qquad (1.15)$$

$$p_2(r,\lambda) = Pr(\xi \in \Omega_2) = 1 - F_r(\lambda) \qquad (1.16)$$

The pair (r,λ) defines the new finite grid $[\Xi^1;B^1]$, which is a refinement of the grid $[\Xi^0;B^0]$.

Once we have solved the problem (P_1), we may proceed to refine $[\Xi^1;B^1]$ in a similar way, generating a new problem (P_2). At each iteration, we are increasing the size of the problem, adding m rows and n columns, where m is the number of recourse constraints and n is the dimension of the recourse decision variable, y. This refinement process may be continued until either 1) the bounds on the optimal value of the original problem are tight enough, 2) the first stage decision variable, x, settles on a fixed value, or 3) the problem becomes intractable. Therefore, given a procedure for determining the initial discretization, we may apply the same procedure iteratively to refine the discretization. We describe this process in detail in section 4.

The problem we wish to analyze in this paper is how to choose the partition. That is, how do we select the r and λ? We assume that we have solved the problem (EV) and have an optimal basis, its corresponding first and second stage solution, (x^*,y^*), and the optimal dual multipliers (π^*,γ^*) on constraints (1.7) and (1.8) respectively.

Section 2: Derivation of the Partition Selection Problem

Olsen showed that the values of (P_i) will converge to the solution of (P) (in the weak sense of ε-convergence). Moreover, it can be shown that $(EV) \leqq (P_1) \leqq (P_2) \leqq \ldots \leqq (P)$ (e.g. see Hiller, 1985). Therefore, the primary criterion for determining a "good" partition is that it should maximize the rate of convergence. In a local sense, it should maximize the difference between the objective value of (P_1) and (EV).

In practice, the analyst is usually interested in the optimal setting of the first stage decision variable, x. Therefore, a secondary criterion for choosing a "good" partition is that the problem (P_1) resulting from the partition should, if possible, force a change in the first stage decision variable. That is, $x^*_{(EV)}$ should differ from $x^*_{(P1)}$. With this criterion, the analyst may be relatively sure that the discretized problems have selected the optimal value of the first stage decision vector, x, when it stablilizes for several iterations of the procedure.

Our primary objective is met by solving the following problem:

$$Max_{r,\lambda}\, z(r,\lambda) = \{z^*_1 - z^*_0\}$$

$$= Min_{x y_1 y_2}\left[c_x x + p_1(r,\lambda) c_y y_1 + p_2(r,\lambda) c_y y_2\right] - z^*_0 \qquad (2.1)$$

$$(P_1) \qquad s.t.\ A_x x \qquad\qquad = b \qquad (2.2)$$

$$A_y x \quad + B y_1 \qquad = \xi_1(r,\lambda) \qquad (2.3)$$

$$A_y x \qquad\quad + B y_2 = \xi_2(r,\lambda) \qquad (2.4)$$

$$x \geq 0,\ y \geq 0$$

We take the dual of this problem and rescale the dual variables by setting $\gamma_1 = \gamma_1/p_1(r,\lambda)$, and $\gamma_2 = \gamma_2/p_2(r,\lambda)$ to get the equivalent problem:

$$z(r,\lambda) = Max_{\pi,\gamma_1,\gamma_2}\left[\pi b + \gamma_1 p_1(r,\lambda)\xi_1(r,\lambda) + \gamma_2 p_2(r,\lambda)\xi_2(r,\lambda)\right] - (\pi^* b + \gamma^* \xi_0) \qquad (2.5)$$

$$(D_1) \qquad s.t.\ \pi A_x + \gamma_1 p_1(r,\lambda) A_y + \gamma_2 p_2(r,\lambda) A_y \leq c_x \qquad (2.6)$$

$$\gamma_1 B \qquad\qquad \leq c_y \qquad (2.7)$$

$$\gamma_2 B \qquad \leq c_y \tag{2.8}$$

We call problem (D_1) the **modified dual problem** of (P_1).

Dualizing the complicating constraint, (2.6) with the multiplier x gives us the following lagrangean:

$$L(x;r,\lambda) = Max \left\{ \pi(b - A_x x) + \gamma_1 p_1(r,\lambda)[\xi_1(r,\lambda) - A_y x] + \gamma_2 p_2(r,\lambda)[\xi_2(r,\lambda) - A_y x] \right\} \tag{2.9}$$

$$+ \ (c_x x - \pi^* b - \gamma^* \xi_0)$$

$$\pi \ uis \qquad \gamma_1 \in \Gamma \qquad \gamma_2 \in \Gamma \tag{2.10}$$

where we have defined the linear constraint set $\Gamma = \{\gamma | \gamma B \leqq c_y\}$.

We know that $L(x;r,\lambda) \geqq z(r,\lambda)$ in general, and in particular, $L(x^*;r,\lambda) \geqq z(r,\lambda)$ where x^* is the optimal first-stage solution for (EV). Furthermore, $L(x^*;r,\lambda_l{}^r) = L(x^*;r,\lambda_u{}^r) = 0$ for all $1 \leqq r \leqq n$ since (P1) and (EV) are the same problem if λ is set at its lower or upper bounds given our assumption that $f(\xi)$ is continuous. If we fix the lagrangean multiplier at x^*, (2.9)-(2.10) decomposes into

$$\begin{matrix} Max \ \pi(b - A_x x^*) \\ + c_x x^* - \pi^* b \\ s.t. \ \pi \ uis \end{matrix} \ + \ p_1(r,\lambda) \begin{matrix} Max \ \gamma_1[\xi_1(r,\lambda) - A_y x^*] \\ - \gamma^* \xi_0 \\ s.t. \ \gamma_1 \in \Gamma \end{matrix} \ + \ p_2(r,\lambda) \begin{matrix} Max \ \gamma_2[\xi_2(r,\lambda) - A_y x^*] \\ - \gamma^* \xi_0 \\ s.t. \ \gamma_2 \in \Gamma \end{matrix}$$

The first subproblem is solved at π^*, the optimal dual multipliers for the first stage constraints in (EV). The second and third subproblems are exactly the same except for the rth coefficient of the objective function. They may be solved quickly with LP objective function parametric programming from the bfs $\gamma_1 = \gamma_2 = \gamma^*$.

We propose to examine the problem $\text{Max}_{r,\lambda}[L(x^*;r,\lambda)]$ rather than the harder problem, $\text{Max}_{r,\lambda}[z(r,\lambda)]$. We justify this with three observations: first, the latter problem is a nonlinear programming problem with nonlinear constraints, a problem that is difficult to solve at best and completely intractable at worst, while the former problem is a problem with linear constraints which exhibit very special structure, albeit the objective is nonlinear and in general nonconvex. Secondly, in most practical problems, the optimal first stage decision for the (EV) problem is a good estimate for that of the problem (P_1) since the first stage constraints in (EV) are the same and the recourse constraints in (EV) represent a simple aggregation of the recourse constraints of (P_1). Finally, and most importantly, if we solve the problem $\text{Max}_{r,\lambda}[L(x^*;r,\lambda)]$, we are confined to two cases:

Case 1: $\text{Max}_{r,\lambda}[L(x^*;r,\lambda)]$ is only slightly greater than 0. In this case, since

$\mathrm{Max}_{r,\lambda}[L(x^*;r,\lambda)] \geqq \mathrm{Max}_{r,\lambda}[\mathrm{Min}_x[L(x;r,\lambda)]] = Z_{(P1)}-Z_{(EV)}$, we know that no other partition (r,λ) can achieve a greater objective value for (P_1) and our primary criterion for a "good" partition has been met.

Case 2: $\mathrm{Max}_{r,\lambda}[L(x^*;r,\lambda)] \gg 0$. There are two possibilities:

Case 2a: the optimal first stage solution to (P_1) **is** x^* in which case the objective value of $(P_1) = Z_{(EV)} + \mathrm{Max}_{r,\lambda}[L(x^*;r,\lambda)] \gg Z_{(EV)}$ and our primary criterion for a "good" partition has been met.

Case 2b: the optimal first stage solution to (P_1) **is not** x^* in which case the partition (r,λ) will force a change in x and our secondary criterion for a "good" partition has been met.

Thus, regardless of what the outcome of the optimization is, we may make a "good" partition. (We note as an aside that $L(x^*;r,\lambda)$ is bounded due to our assumption of complete recourse.)

Thus, we wish to maximize (2.11) wrt (r,λ). One simple procedure for doing this is to define the 2n functions $g_1(\lambda), g_2(\lambda), \ldots, g_n(\lambda)$ and $h_1(\lambda), h_2(\lambda), \ldots, h_n(\lambda)$ as

$$g_r(\lambda) = \left\{ \begin{array}{c} Max\, \gamma_1[\xi_1(r,\lambda) - A_y x^*] - \gamma^* \xi_0 \\ s.t.\ \ \gamma_1 \in \Gamma \end{array} \right\} \qquad (2.12)$$

$$h_r(\lambda) = \left\{ \begin{array}{c} Max\, \gamma_2[\xi_2(r,\lambda) - A_y x^*] - \gamma^* \xi_0 \\ s.t.\ \ \gamma_2 \in \Gamma \end{array} \right\} \qquad (2.13)$$

For each r, we maximize the one dimensional nonlinear program

$$\begin{array}{l} Max_\lambda\ \{p_1(r,\lambda)g_r(\lambda) + p_2(r,\lambda)h_r(\lambda)\} \\ s.t.\ \ \lambda_l^r \le \lambda \le \lambda_u^r \end{array} \qquad (2.14)$$

For expositional purposes, we define the function $\psi_r(\lambda)$ as

$$\psi_r(\lambda) \equiv p_1(r,\lambda)g_r(\lambda) + p_2(r,\lambda)h_r(\lambda) \qquad (2.15)$$

We call problem (2.14) the partition selection problem since its solution will provide us with an efficient procedure for selecting a good partition to refine the deterministic approximation of the stochastic LP with recourse.Given the solution to (2.14) for each r, we simply choose the r which gives the largest value.

Unfortunately, even for fixed r, (2.15) is neither concave nor convex for general distributions F(ξ). If F(ξ) were uniform, (2.14) would reduce to a set of quadratic programming problems, which could be solved quickly to optimality. We postpone our discussion of this and other special cases and continue with our general discussion in the next section.

Section 3: Analysis and Solution of the Partition Selection Problem

In this section we prove a series of results that will give us more insight into the behavior of problem (2.14). We then propose an algorithm that is guaranteed to solve (2.14) in a finite number of steps. The algorithm is difficult to implement in practice because it requires the inverse of the conditional expected value function. Moreover, it may require a substantial amount of time to solve to optimality. Therefore, we do not recommend the implementation of this algorithm for most practical problems. In the next section we will propose an efficient algorithm which will provide an approximation to the solution.

Figure 3.1 shows a plot of $\psi_r(\lambda)$. The following lemmas reveal the possible variants of figure 3.1.

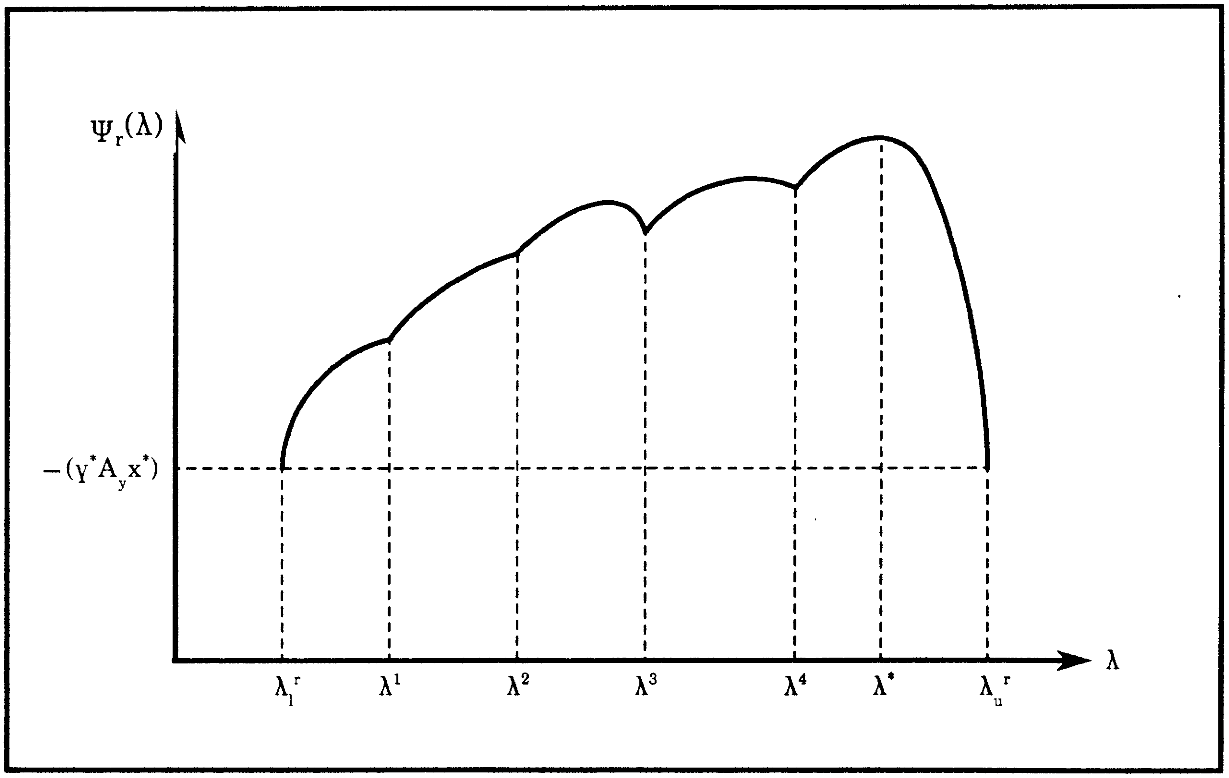


Figure 3.1: Plot of (2.14)

Lemma 3.1: $[\psi_r(\lambda_l^r) \equiv h_r(\lambda_l^r)] = [\psi_r(\lambda_u^r) \equiv g_r(\lambda_u^r)] = -(\gamma^* A_y x^*)$.

Proof: By definition, the components of the vector $\xi_1(r,\lambda)$ are

$$\xi_1(r,\lambda) = (\xi_0^1, \xi_0^2, \ldots, \xi_0^{r-1}, \xi_1^r(\lambda), \xi_0^{r+1}, \ldots, \xi_0^n)$$

Similarly, the components of the vector $\xi_2(r,\lambda)$ are

$$\xi_2(r,\lambda) = (\xi_0^1, \xi_0^2, \ldots, \xi_0^{r-1}, \xi_2^r(\lambda), \xi_0^{r+1}, \ldots, \xi_0^n)$$

The proof is complete by the simple observations that $p_1(r,\lambda_l^r) = 0$, $p_2(r,\lambda_l^r) = 1$, $[\xi_2^r(\lambda_l^r) \equiv E(\xi | \xi \geqq \lambda_l^r)] = \xi_0^r$, and $p_1(r,\lambda_u^r) = 1$, $p_2(r,\lambda_u^r) = 0$, and $[\xi_1^r(\lambda_u^r) \equiv E(\xi | \xi \leqq \lambda_u^r)] = \xi_0^r$. Therefore,

$$\left\{ \begin{array}{c} Max\, \gamma_1[\xi_0 - A_y x^*] - \gamma^* \xi_0 \\ s.t.\ \ \gamma_1 \in \Gamma \end{array} \right\} = \left\{ \begin{array}{c} Max\, \gamma_2[\xi_0 - A_y x^*] - \gamma^* \xi_0 \\ s.t.\ \ \gamma_2 \in \Gamma \end{array} \right\} = -(\gamma^* A_y x^*)$$

☒

Lemma 3.2: $\psi_r(\lambda) \geqq -(\gamma^* A_y x^*)$ for all $\lambda_l^r \leqq \lambda \leqq \lambda_u^r$.

Proof: Figure 3.2 shows a plot of $g_r(\xi_1^r)$ and $h_r(\xi_2^r)$ as functions of $\xi_1^r(\lambda)$ and $\xi_2^r(\lambda)$ (note: <u>not</u> as function of λ). Figure 3.2 can be interpretted in two ways: 1) as a simple LP parametric plot for (EV) as a function of the r^th component of ξ_0, and 2) as a combined plot of $g_r(\xi_1^r)$ and $h_r(\xi_2^r)$ where the domain of $g_r(\xi)$ is $\lambda_l^r \leqq \xi \leqq \xi_0^r$ and the domain of $h_r(\xi)$ is $\xi_0^r \leqq \xi \leqq \lambda_u^r$. Now, define $t_1 = \xi_1^r(\lambda)$, $t_2 = \xi_2^r(\lambda)$, $a = p_1(r,\lambda)$, and $(1-a) = p_2(r,\lambda)$. Our result follows immediately from convexity of LP right-hand side parametrics and the fact that $at_1 + (1-a)t_2 = \xi_0^r$. ☒

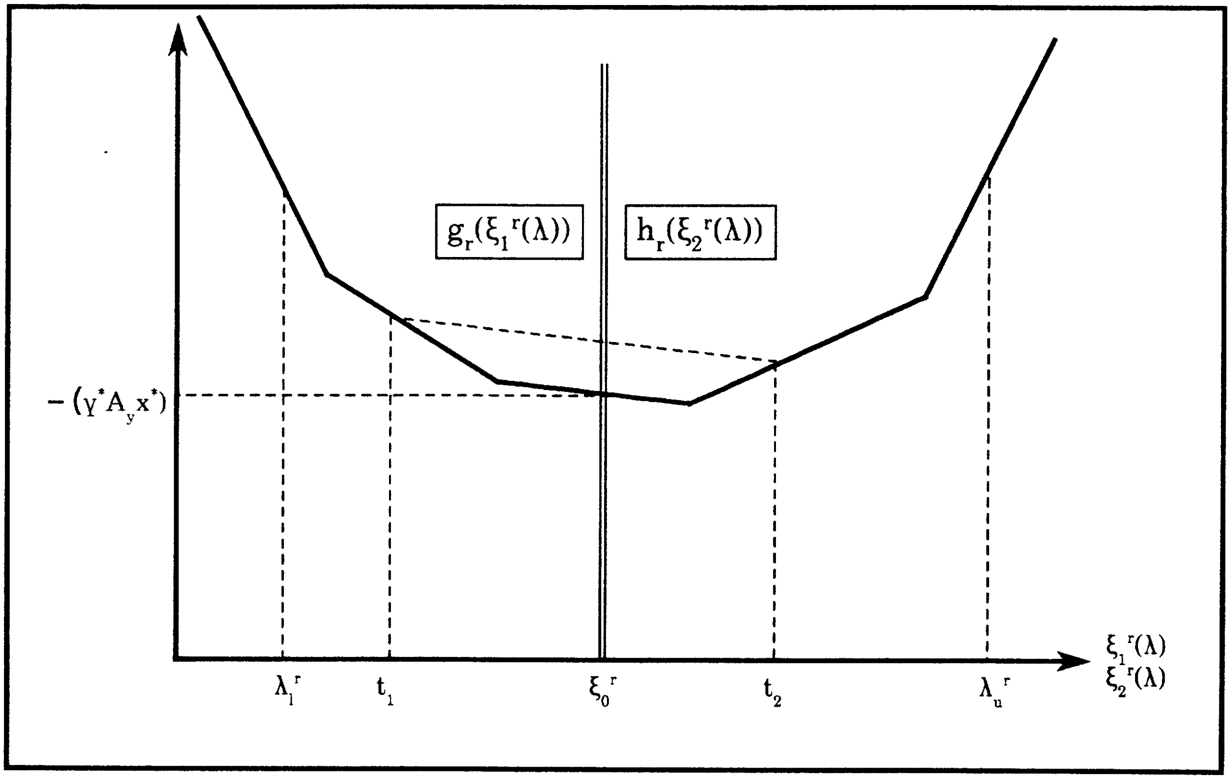


Figure 3.2: RHS Parametric on (EV)

Lemma 3.3: $\psi_r(\lambda)$ is continuous and bounded on $(\lambda_l^r,\lambda_u^r)$.

Proof: $\xi_1^r(\lambda)$ and $\xi_2^r(\lambda)$ are continuous bounded functions of λ in the open interval $(\lambda_l^r,\lambda_u^r)$. Furthermore, $g_r(\xi_1^r)$ is continuous and bounded in ξ_1^r. The same is true for $h_r(\xi_2^r)$. Therefore, $g_r(\lambda)$ and $h_r(\lambda)$ are continuous and bounded. Similarly, $p_1(r,\lambda)$ and $p_2(r,\lambda)$ are continuous by our assumption that $F(\cdot)$ is continuous. Therefore $\psi_r(\lambda) \equiv p_1(r,\lambda)g_r(\lambda) + p_2(r,\lambda)h_r(\lambda)$ is continuous and bounded.

☒

Theorem 3.1: There exists a λ^* such that (2.14) is maximized at λ^* and $\lambda_l^r < \lambda^* < \lambda_u^r$.

Proof: (2.14) is bounded for all λ by our assumptions of fixed recourse and bounded support for ξ. Now, using Lemmas 3.1 and 3.2, we have that $\psi_r(\lambda) \geqq \psi_r(\lambda_l^r) = \psi_r(\lambda_u^r)$ for all $\lambda \in (\lambda_l^r,\lambda_u^r)$. Furthermore, the maximum is attained since $\psi_r(\lambda)$ is continuous and bounded by lemma 3.3.

☒

Now we may continue our analysis of problem (2.14), with the knowledge that an interior solution will exist. (We do not dismiss the possibility that $p_1(r,\lambda)g_r(\lambda) + p_2(r,\lambda)h_r(\lambda) = -(y^*A_yx^*)$ for all $\lambda_l^r \leqq \lambda \leqq \lambda_u^r$.) Although $\psi_r(\lambda)$ is neither concave nor convex on $[\lambda_l^r,\lambda_u^r]$, we do have a convexity result fot $g_r(\lambda)$ and $h_r(\lambda)$.

Theorem 3.2: $g_r(\lambda)$ and $h_r(\lambda)$ are strictly quasiconvex on $[\lambda_l^r, \lambda_u^r]$.

Proof: Recall the definition of strict convexity: $g_r(\lambda)$ is strictly convex on a convex set Ξ^r if for all $\lambda_0 \in \Xi^r$, the following holds: for any $\lambda \in \Xi^r$, $g_r(\lambda) < g_r(\lambda_0)$ implies that $g_r((1-\alpha)\lambda_0 + \alpha\lambda) < g_r(\lambda_0)$ for all $0 < \alpha < 1$.
We prove the theorem for $g_r(\lambda)$ by contradiction. The proof is exactly the same for $h_r(\lambda)$. Assume that there exists a $\lambda_0 \in [\lambda_l^r, \lambda_u^r]$, a $\lambda \in [\lambda_l^r, \lambda_u^r]$, and an $\alpha \in (0,1)$ with $g_r(\lambda) < g_r(\lambda_0)$ and $g_r((1-\alpha)\lambda_0 + \alpha\lambda) \geqq g_r(\lambda_0)$. Now, we define the function $\xi_1^r(\lambda)$ mapping the "cut", λ, into the value $\xi_1^r = E[\xi | \xi \leqq \lambda]$ (figure 2.2 is a plot of $g_r(\xi_1^r(\cdot))$). $\xi_1^r(\lambda)$ is a continuous 1-1 increasing function because of our assumption that $f_r(\xi)$ is continuous and positive on Ξ^r. Therefore, there exists a $\beta \in (0,1)$ such that
$\xi_1^r((1-\alpha)\lambda_0 + \alpha\lambda) = (1-\beta)\xi_1^r(\lambda_0) + \beta\xi_1^r(\lambda)$.

$$\begin{aligned} g_r[(1-\alpha)\lambda_0 + \alpha\lambda] &= g_r[\xi_1^r((1-\alpha)\lambda_0 + \alpha\lambda)] \\ &= g_r[(1-\beta)\xi_1^r(\lambda_0) + \beta\xi_1^r(\lambda)] \end{aligned}$$

$$
\begin{aligned}
&\leqq (1-\beta)g_r[\xi_1{}^r(\lambda_0)] + \beta g_r[\xi_1{}^r(\lambda)] \text{ by convexity of } g_r[\xi_1{}^r]\\
&= (1-\beta)g_r[\lambda_0] + \beta g_r[\lambda]\\
&< (1-\beta)g_r[\lambda_0] + \beta g_r[\lambda_0] \text{ since } \beta \neq 0 \text{ and } g_r(\lambda) < g_r(\lambda_0)\\
&= g_r[\lambda_0]
\end{aligned}
$$

Hence, $g_r[(1-\alpha)\lambda_0 + \alpha\lambda] < g_r[\lambda_0]$, which contradicts our assumption to the contrary.

☒

Further analysis must be completed on the function $\psi_r(\lambda)$ in order to render problem (2.14) tractible. The following theorems provide us with a convenient method for solving (2.14). We prove that $(\lambda_l{}^r,\lambda_u{}^r)$ can be partitioned into a finite number of intervals, $(\lambda_0 \equiv \lambda_l{}^r,\lambda_1)$, (λ_1,λ_2), ..., $(\lambda_{N-1},\lambda_N \equiv \lambda_u{}^r)$, such that $\psi_r(\lambda)$ is continuously differentiable on each. Moreover, on each interval, the funtion has at most one local maximum that is not an extreme point of one of these intervals. Therefore, we may restrict our search for λ^* to a finite set. To solve (2.14), we need only compare the value at each endpoint, λ_i, (except the two endpoints $\lambda_l{}^r$ and $\lambda_u{}^r$ since we know an interior solution exists) and possibly one other interior point. Thus, we need compute and compare at most N values of $p_1(r,\lambda)g_r(\lambda) + p_2(r,\lambda)h_r(\lambda)$. It is of practical significance that N is usually not too large.

Theorem 3.3: $\psi_r(\lambda)$ is continuously differentiable on $(\lambda_{i-1},\lambda_i{}^r$. Furthermore, the nondifferentiable points, λ_0, λ_1, λ_2, λ_N, correspond to vertices of the graph of figure (3.2).

Proof: First note that $\xi_1(r,\lambda)$ and $\xi_2(r,\lambda)$ are strictly increasing functions in $\mathbb{C}^1$ on $(\lambda_l{}^r,\lambda_u{}^r)$ since $f(\lambda)>0$. Furthermore, $g_r(\xi_1{}^r)$ and $h_r(\xi_2{}^r)$ are piecewise linear functions of $\xi_1{}^r$ and $\xi_2{}^r$ as figure 3.2 shows. Therefore, $g_r(\lambda)$ and $h_r(\lambda)$ are piecewise continuously differentiable everywhere except where $\xi_1{}^r(\lambda)$ is at a breakpoint of $g_r(\xi_1{}^r)$ or where $\xi_2{}^r(\lambda)$ is at a breakpoint of $h_r(\xi_2{}^r)$. However, there are a finite number of such breakpoints and $\xi_1{}^r(\cdot)$ and $\xi_2{}^r(\cdot)$ are one-to-one, hence, $g_r(\lambda)$ and $h_r(\lambda)$ are continuously differentiable in $\{(\lambda_l{}^r,\lambda_u{}^r) - \{\lambda_1,\lambda_2,\ldots,\lambda_{N-1}\}\}$. $p_1(r,\lambda)$ and $p_2(r,\lambda)$ are continuous and continuously differentiable everywhere on $(\lambda_l{}^r,\lambda_u{}^r)$, so we conclude that $\psi_r(\lambda)$ is continuously differentiable on $(\lambda_0 \equiv \lambda_l{}^r,\lambda_1)$, (λ_1,λ_2), ..., and $(\lambda_{N-1},\lambda_N \equiv \lambda_u{}^r)$.

☒

Theorem 3.4: There exists at most one critical point of (2.14) on $(\lambda_l^r, \lambda_u^r)$, and if $f(\xi)$ is differentiable, then the critical point must be a local maximum.

Proof: Consider some interval, $(\lambda_{i-1}, \lambda_i)$. There exists a unique vector $\gamma_1^i \equiv [\gamma_{11}^i, \gamma_{12}^i, \ldots, \gamma_{1m}^i]^T$ which solves (2.12), and a unique vector $\gamma_2^i \equiv [\gamma_{21}^i, \gamma_{22}^i, \ldots, \gamma_{2m}^i]^T$ which solves (2.13). Hence,

$$\psi_r(\lambda) = \gamma_1^i \int_{\lambda_l^r}^{\lambda} (\xi - A_y x^*) f(\xi)\, d\xi + \gamma_2^i \int_{\lambda}^{\lambda_u^r} (\xi - A_y x^*) f(\xi)\, d\xi - \gamma^* \xi_0$$

By theorem 3.3, this function is differentiable wrt λ, and its derivative is

$$\frac{d\psi_r(\lambda)}{d\lambda} = f(\lambda)\left[(\lambda - A_y^r x^*)(\gamma_{1r}^i - \gamma_{2r}^i)\right] \tag{3.1}$$

Now, since $f(\lambda) > 0$ on $(\lambda_{i-1}, \lambda_i)$, there is no critical point if $\gamma_{1r}^i = \gamma_{2r}^i$, in which case, $\psi_r(\lambda)$ is linear, and one critical point at

$$\lambda'^* \equiv A_y^r x^* \tag{3.2}$$

if $\gamma_{1r}^i \neq \gamma_{2r}^i$ and if the λ'^* so defined falls within the segment $(\lambda_{i-1}, \lambda_i)$. (3.2) is independent of the segment $(\lambda_{i-1}, \lambda_i)$ selected. Hence, it λ'^* is the same for all intervals and it lies in the interior of at most one segment. If $f(\lambda)$ is differentiable, then

$$\left.\frac{d^2\psi_r(\lambda)}{d\lambda^2}\right|_{\lambda'^*} = f'(\lambda'^*)\left[(\lambda'^* - A_y^r x^*)(\gamma_{1r}^i - \gamma_{2r}^i)\right] + f(\lambda'^*)(\gamma_{1r}^i - \gamma_{2r}^i)$$

$$= f(\lambda'^*)(\gamma_{1r}^i - \gamma_{2r}^i)$$

But $\gamma_{1r}^i \leqq \gamma_{2r}^i$ since $\xi_1^r(\lambda) < \xi_2^r(\lambda)$, $\xi_1^j(\lambda) = \xi_2^j(\lambda) = \xi_0^j j()$ for all $j \neq r$, and (2.12) and (2.13) are LP maximization problems. Furthermore, if there is a critical point, then we know that $\gamma_{1r}^i \neq \gamma_{2r}^i$ and hence, $\gamma_{1r}^i < \gamma_{2r}^i$. Therefore, $d^2\psi_r(\lambda)/d\lambda^2 |_{\lambda'^*} < 0$, so λ'^* must be a local maximum.

☒

Theorems 3.3 and 3.4 suggest a method for solving (2.14). We know that the optimal solution is in the finite set $\{\lambda_1, \lambda_2, \ldots, \lambda_{N-1}, \lambda'^*\}$. Therefore, we need only compute $\psi_r(\lambda_i)$ for $i = 1,2,\ldots,N$, and $\psi_r(\lambda'^*)$ to find the maximizing λ^*. One algorithm for doing this is the following.

ALGORITHM 1

Step 0: Initialize r to the first stochastic index. Set $\alpha = 0$ and $s = 0$.

Step 1: To compute λ_i and λ'^* for the current partition index, r, we solve a parametric programming problem to obtain figure 3.2 for the range $\lambda_l^r \leqq \lambda \leqq \lambda_u^r$. This gives us the set of pivot points, $\{\xi^1, \xi^2, \ldots, \xi^{N-1}\}$.

Step 2: Define $\lambda(\xi^i)$ as the inverse of the conditional expectation function.

$$\lambda(\xi^i) : \xi^i \rightarrow \lambda \quad s.t. \begin{cases} \displaystyle\int_{\lambda_l^r}^{\lambda} \xi f(\xi)\, d\xi = \xi^i \int_{\lambda_l^r}^{\lambda} f(\xi)\, d\xi & \text{if } \xi^i \leq \xi_0^r \\ \displaystyle\int_{\lambda}^{\lambda_u^r} \xi f(\xi)\, d\xi = \xi^i \int_{\lambda}^{\lambda_u^r} f(\xi)\, d\xi & \text{if } \xi^i \leq \xi_0^r \end{cases}$$

Calculate $\lambda_i = \lambda(\xi^i)$ for all $i = 1,2,\ldots,N-1$ and sort all the values to obtain the ordered set $\{\lambda_1, \lambda_2, \ldots, \lambda_{N-1}\}$.

Step 3: Calculate $\lambda'^* = A_y^r x^*$, and discard the value if it does not lie within the range $\lambda_l^r \leqq \lambda'^* \leqq \lambda_u^r$.

Step 4: Steps 1 – 3 provide us with the finite set $\{\lambda_1, \lambda_2, \ldots, \lambda_{N-1}, \lambda'^*\}$ some member of which we know maximizes $\psi_r(\lambda)$. We simply compute $\psi_r(\lambda)$ for each element and choose the largest to locate the optimal partition cut λ^r for the candidate partition index, r.

Step 5: If $\psi_r(\lambda^r) > \alpha$, then set $\alpha = \psi_r(\lambda^r)$, $s = r$, and $\lambda^* = \lambda^r$.

Step 6: If all the candidate partition indices have been processed, then STOP, the optimal partition is (s, λ^*). Otherwise, set $r = r + 1$, and go to step 1.

Although algorithm 1 will solve our partition selection problem to optimality, it may require an inordinate amount of time to do so. Even though it is guaranteed to solve the problem in a finite number of steps, we must execute the body of algorithm 1 (steps 1-4) for each candidate r. That is, we must execute it m times. Moreover, we require the inverse of the conditional expectation function to complete step 2 of the algorithm. This is in general not available. Therefore, though we can in theory determine the optimal partition with algorithm 1, we must look for a more applicable algorithm to solve practical problems. We address this issue in the next section.

Section 4: Solving the Partition Selection Problem Locally

In the last section we developed an algorithm guaranteed to solve (2.14) in finite time. In practice, however, we wish to obtain an adequate partition using as little time as possible. In this section, we develop a fast operational procedure for selecting the partition (r,λ).

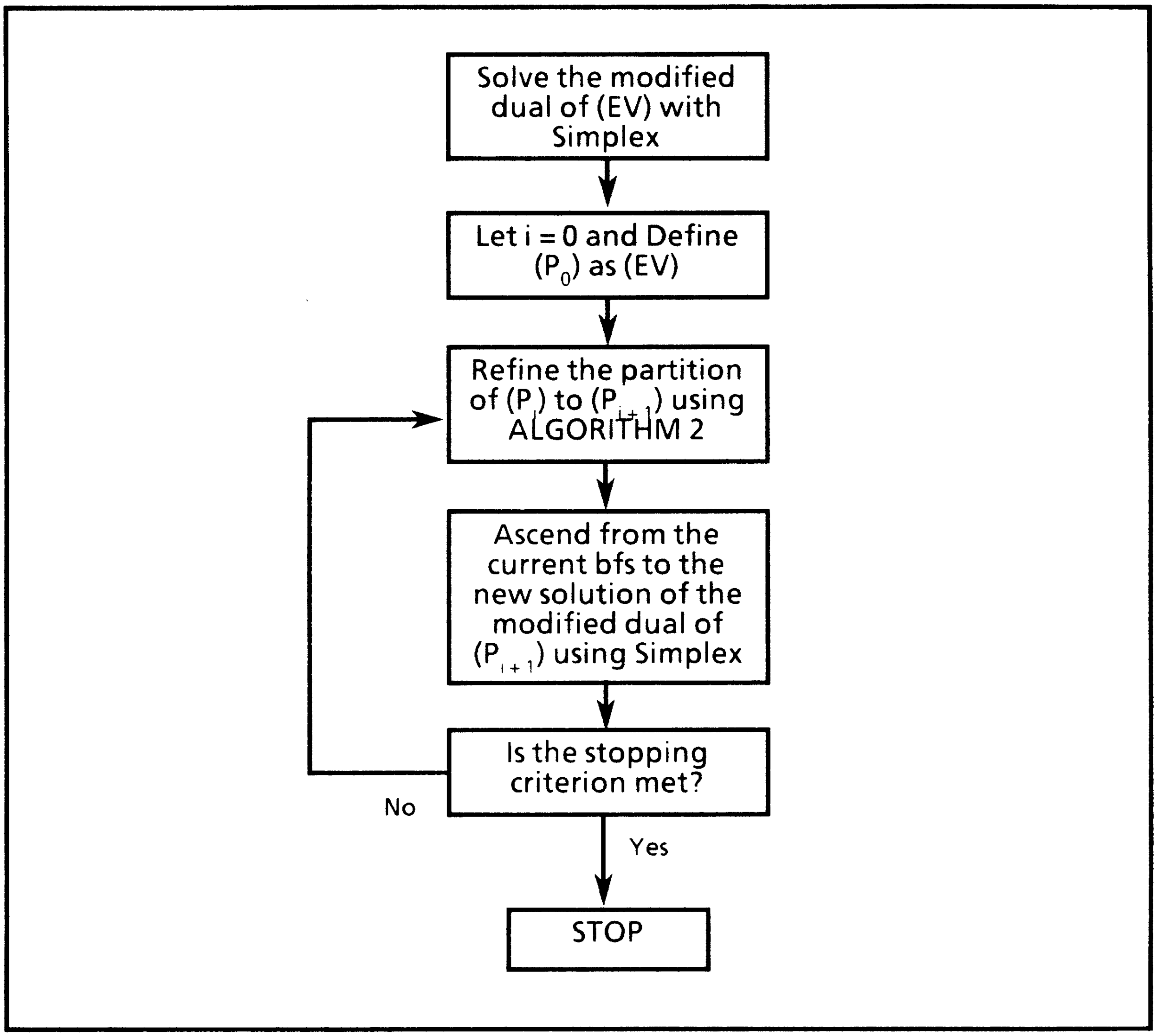
In place of solving (2.14), we propose to select the partition cut, λ^r, to give us the largest subgradient of simplex ascent in (2.14), at the initial solution of $\gamma_1 = \gamma_2 = \gamma^*$. This is justified by noticing that with λ fixed (2.14) decomposes into a weighted sum of the two LP problems, (2.12) and (2.13), with initial solutions of $\gamma_1 = \gamma^*$ and $\gamma_2 = \gamma^*$ respectively. These initial solutions are nearly optimal and if either is nonoptimal, the optimal solution is only a few pivots away.

This procedure, which for obvious reasons we call Algorithm 2, can be embedded in the overall solution procedure for two-stage stochastic programming problems with fixed recourse (P) as shown in figure 4.1. In this case, the selection criterion for λ proposed in this section will give us the largest reduced cost for the first pivot of the ascent from the solution of (D_i) to the solution of (D_{i+1}). This selection criteria at first seems very myopic. We are not choosing λ to maximize (2.14) globally. We are not even choosing it to maximize (2.14) locally. We are simply setting λ to give us the steepest direction of steepest ascent for (2.14). Nevertheless, the results we obtain in applying this rule are intuitively appealing and are very easy to calculate in practice. No numerical integration is required on $f(\xi)$. Hence, we may apply the method to problems with very complex density functions. Moreover, unlike algorithm 1, algorithm 2 uses information about the optimal solution to the current discretization in determining the next discretization. This type of feedback is extremely appealing from a practical standpoint since the algorithm can direct itself into the more promising regions as it discovers their existence.

Given r and λ, the partition selection problem (2.14) decomposes into

$$\left\{\begin{array}{l} Max\ \gamma_1 c_1(r,\lambda) \\ s.t.\ \ \gamma_1 B \le c_y \end{array}\right\} + \left\{\begin{array}{l} Max\ \gamma_2 c_2(r,\lambda) \\ s.t.\ \ \gamma_2 B \le c_y \end{array}\right\} \tag{4.1}$$

where $c_1(r,\lambda) = F_r(\lambda)[\xi_1(r,\lambda) - A_y x^*]$, and $c_2(r,\lambda) = (1 - F_r(\lambda))[\xi_2(r,\lambda) - A_y x^*]$. For each subproblem we have a good initial bfs from the solution to the modified dual of (EV). (Note: The modified dual and the dual of (EV) are exactly the same. It is only the latter problems (P_i) where the modified dual is different than the dual.)



Solve the modified dual of (EV) with Simplex
Let i = 0 and Define (P_0) as (EV)
Refine the partition of (P_i) to (P_{i+1}) using ALGORITHM 2
Ascend from the current bfs to the new solution of the modified dual of (P_{i+1}) using Simplex
Is the stopping criterion met?
No
Yes
STOP

Figure 4.1: Procedure for Solving (P)

Before we proceed, we define the following notation:

$[A_B|B_B]$ is the optimal basis for the modified dual of (EV).

$c_{1B}(r,\lambda)$ is the vector of elements of $c_1(r,\lambda)$ corresponding to the basis B_B.

$c_{2B}(r,\lambda)$ is the vector of elements of $c_2(r,\lambda)$ corresponding to the basis B_B.

e_r is the rth unit vector, [0,0,...,0,1,0,...,0].

y^* is the optimal recourse vector for (EV).

y_1 is the vector of dual prices for the first set of constraints in (4.1).

y_2 is the vector of dual prices for the second set of constraints in (4.1).

ç represents the reduced cost of the element with objective coefficient c. $(A_y)_i$ is the ith row of A_y.

In the following analysis, we derive results for the first subproblem of (4.1) only. The results are exactly analagous for the second subproblem unless stated otherwise.

We wish to derive an expression for the reduced cost of element $\gamma_1{}^i$ as a function of r and λ. This reduced cost element, $\text{ç}_1{}^i(r,\lambda)$ is

$$\text{ç}_1^i(r,\lambda) = c_1^i(r,\lambda) - y_1 B \tag{4.2}$$

where, y_1, the vector of simplex multipliers for (4.1) at the bfs $\gamma_1 = \gamma_2 = \gamma^*$, is

$$y_1 = c_{1B}(r,\lambda)\, B_B^{-1} \tag{4.3}$$

$$= \begin{cases} F_r(\lambda)\, y^* & \textit{if r is nonbasic} \\ F_r(\lambda)\, \{y^* + e_r[\xi_1^r(r,\lambda) - \xi_0^r]\, B_B^{-1}]\} & \textit{if r is basic} \end{cases} \tag{4.4}$$

Thus, we have two cases:

Case 1: r is basic.

$$\text{ç}_1(r,\lambda) = F_r(\lambda)\left\{(\xi_0 - A_y x^*) + e_r(\xi_1^r(r,\lambda) - \xi_0^r) - \left[y^* + e_r[\xi_1^r(r,\lambda) - \xi_0^r]\, B_B^{-1}\right] B\right\} \tag{4.5}$$

Case 1a: **i is nonbasic.** $B_B{}^{-1}B_i = B'_i$ where B'_i is a column of the optimal tableau for the modified dual of (EV), and B'_{ir} is the element corresponding to $\gamma_1{}^r$.

$$\text{ç}_1^i(r,\lambda) = F_r(\lambda)\left\{(\xi_0^i - (A_y)_i\, x^*) - y^* B_i + [\xi_1^r(r,\lambda) - \xi_0^r]\, B'_{ir}\right\} \tag{4.6}$$

But $\xi_0{}^i - (A_v)_i x^* - y^* B_i = 0$, so

$$\text{ç}_1^i(r,\lambda) = F_r(\lambda)\left\{[\xi_1^r(r,\lambda) - \xi_0^r)]B'_{ir}\right\} \tag{4.7}$$

Similarly, we can derive the reduced cost for $\gamma_2{}^i$.

$$\text{ç}_2^i(r,\lambda) = (1 - F_r(\lambda))\left\{[\xi_2^r(r,\lambda) - \xi_0^r]\, B'_{ir}\right\} \tag{4.8}$$

Setting $(d\text{ç}_1{}^i(r,\lambda)/d\lambda) = 0$ gives necessary conditions for the maximum (if the resulting $\lambda^* \in [\lambda_l{}^r, \lambda_u{}^r]$). To do this, rewrite $\text{ç}_1{}^i(r,\lambda)$ as

$$\text{ç}_1^i(r,\lambda) = \int_{\lambda_l^r}^{\lambda} (\xi - \xi_0^r) B'_{ir}\, f_r(\xi)\, d\xi \tag{4.9}$$

The derivative of which is easily evaluated as $(\lambda - \xi_0{}^r)B'_{ir} f_r(\lambda)$. For this to be zero, either a) $f_r(\lambda) = 0$, b) $B'_{ir} = 0$, or c) $\lambda = \xi_0{}^r$. a) is impossible due to our initial assumption that $f_r(\cdot) > 0$. Thus, if $B'_{ir} \neq 0$, $\lambda^* = \xi_0{}^r$ maximizes or minimizes $\varsigma_1{}^i(r,\lambda)$.

Now, notice that $(d\varsigma_2{}^i(r,\lambda)/d\lambda)|_{\lambda*} = (d\varsigma_1{}^i(r,\lambda)/d\lambda)|_{\lambda*} = 0$ and furthermore, that at λ^*, either $\varsigma_1{}^i(r,\lambda)$ is maximized and $\varsigma_2{}^i(r,\lambda)$ is minimized or $\varsigma_2{}^i(r,\lambda)$ is maximized and $\varsigma_1{}^i(r,\lambda)$ is minimized. Thus, a partition at λ^* will maximize the next subgradient along all directions of simplex ascent.

Case 1b: **i is basic and i = r.** In this case, the analysis is the same, but $B'_{ir} = 1$.

Case 1c: **i is basic and i ≠ r.** Here, $\varsigma_1{}^i(r,\lambda) = \varsigma_2{}^i(r,\lambda) = 0$ for all λ since $\xi_0 - (A_y)_i x^* - y^* B = 0$ by primal feasibility of the (EV) solution.

Case 2: **r is nonbasic.**

$$\varsigma_1(r,\lambda) = F_r(\lambda)\left\{(\xi_0 - A_y x^*) + e_r[\xi_1^r(r,\lambda) - \xi_0^r] - y^* B\right\} \tag{4.10}$$

But $\xi_0 - (A_v)_i x^* - y^* B = 0$, so (4.10) simplifies to

$$\varsigma_1(r,\lambda) = F_r(\lambda)\left\{e_r[\xi_1^r(r,\lambda) - \xi_0^r]\right\} \tag{4.11}$$

which is nonzero only if i = r. In this case, we use the same analysis as above (with B'_{ir} replaced by 1). The result is the same, $\lambda^* = \xi_0{}^r$ will maximize $\varsigma_1{}^i(r,\lambda)$ and minimize $\varsigma_2{}^i(r,\lambda)$ or vice versa.

Regardless of the case we are in, if we are partitioning along dimension r, we should set the partition cut, λ^r, at the current expected value, $\xi_0{}^r$ to maximize the simplex subgradient.

Now we want to use this information to develop an efficient procedure for determining the best partition index. To be consistent with our analysis so far, we should select the r that maximizes the maximum simplex subgradient. That is, we should calculate $\varsigma_1{}^i(r,\xi_0{}^r)$ for each (r,i) combination and select the r^* such that

$$r^* = argmax_r\left\{max_i\left[max\,[\,\varsigma_1^i(r,\xi_0^r),\, \varsigma_2^i(r,\xi_0^r)\,]\right]\right\} \tag{4.12}$$

We need to calculate $\varsigma_1{}^i(r,\xi_0{}^r)$ if r is basic and i is nonbasic, or if r = i. Therefore, there are $2(m(n-m)+n)$ calculations. If $f_r(\xi)$ is a general density function, each calculation requires a numerical integration as in (4.9). This could conceivably require a substantial amount of time. Therefore, as an approximation, we take the first-order expansion of $F_r(\xi_0{}^r)$ and $F_r(\xi_0{}^r)\xi_1{}^r$ to perform our analysis (note: the approximation is exact if $F_r(\cdot)$ is the uniform distribution).

$$F_r(\xi_0^r) = \int_{\lambda_l^r}^{\xi_0^r} f_r(\xi)\,d\xi \approx (\xi_0^r - \lambda_l^r)\,\frac{f_r(\xi_0^r) + f_r(\lambda_l^r)}{2} \tag{4.13}$$

$$F_r(\xi_0^r)\,\xi_1^r = \int_{\lambda_l^r}^{\xi_0^r} \xi f_r(\xi)\,d\xi \approx (\xi_0^r - \lambda_l^r)\,\frac{\xi_0^r f_r(\xi_0^r) + \lambda_l^r f_r(\lambda_l^r)}{2} \tag{4.14}$$

With this approximation,

$$\begin{aligned} \varsigma_1^i(r,\lambda) &= -\tfrac{1}{2}\cdot f_r(\lambda_l^r)\cdot B_{ir}^{'}\cdot (\xi_0^r - \lambda_l^r)^2 \\ \varsigma_2^i(r,\lambda) &= +\tfrac{1}{2}\cdot f_r(\lambda_u^r)\cdot B_{ir}^{'}\cdot (\lambda_u^r - \xi_0^r)^2 \end{aligned} \tag{4.15}$$

and , though we still have $2(m(n-m)+n)$ calculations, each one consists of simple arithmetic operations with values readily available.

We are now prepared to state algorithm 2 formally:

ALGORITHM 2

Step 0: Initialize r to the first stochastic index, and set $\alpha = 0$, $s = 0$.

Step 1: Calculate the two values of (4.15).

Step 2: If either value is greater than α, then set $\alpha = \max[\varsigma_1^i(r,\lambda), \varsigma_2^i(r,\lambda)]$ and set $s = r$.

Step 3: If all stochastic elements have been processed, then STOP, the selected partition is (s,ξ_0^s). Otherwise, set $r = r+1$, and go to step 1.

After performing this analysis, we have a partition index, call it s, and a partition cut, ξ_0^s. After solving the resulting problem (P_1), we may again perform this analysis with the lower and upper bounds on ξ^s for each of the two recourse subproblems set at $[\lambda_l^s, \xi_0^s]$ and $[\xi_0^s, \lambda_u^s]$ respectively.

Figure 4.2 shows a graphical representation of a possible partition sequence for a problem with 2 stochastic RHS elements. Figure 4.3 shows the resulting discretization after the fourth partition. At this stage, problem (P_4) has 5n recourse variables and 5m recourse constraints. In this example, the partition sequence may have been $\{(1,\xi_0^1),(2_1,\xi_0^2),(1_2,\xi_1^1),(2_{22},\xi_0^2)\}$.

Algorithm 2 will automaticall refine components of the grid in "promising" areas. That is, it will eliminate those elements of the deterministic equivalent problem that serve to make the LP large without adding to the structure of the

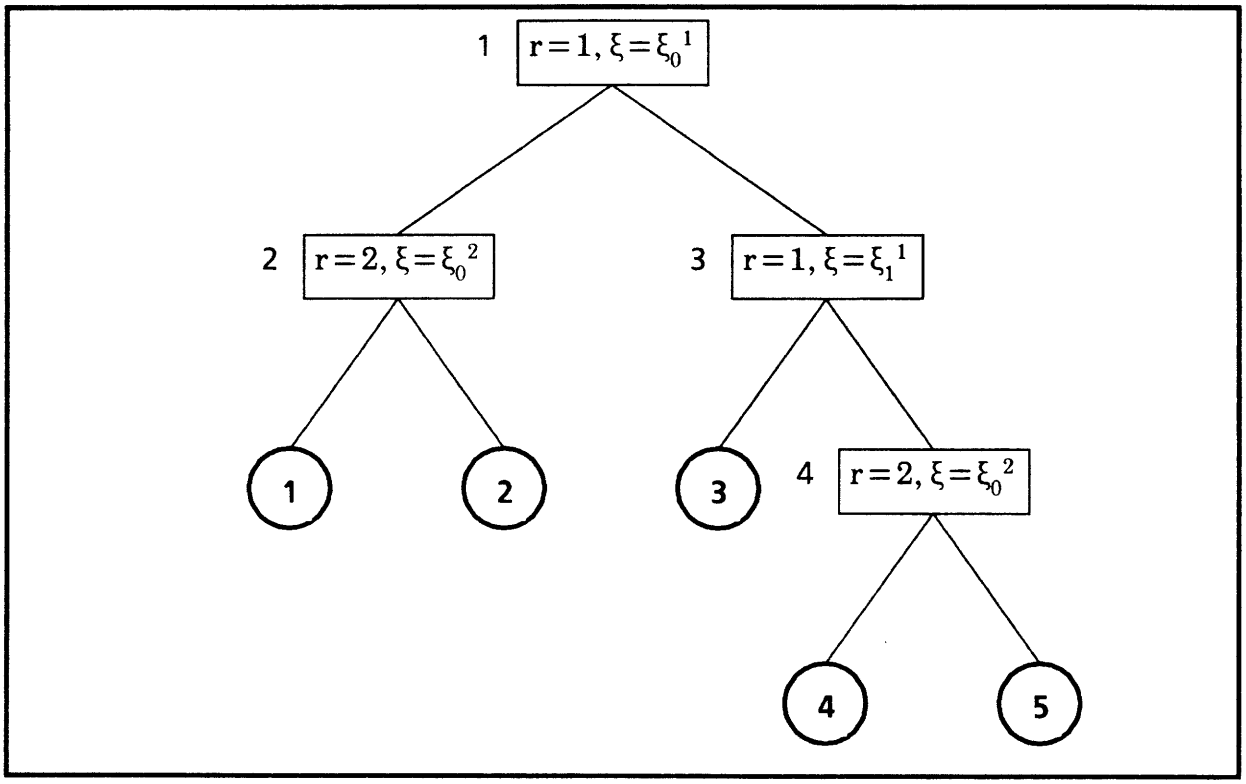


Figure 4.2: A Sample Partition Sequence

problem. It is our belief that algorithm 2 provides a good balance between ease of implementation, speed of execution and outcome of results. Furthermore, the result of our analysis in this section serve to both support and extend ones intuition. It is intuitively appealing that the result is to partition at the conditional expectation point. However, ones intuition could not be relied upon to determine the partition dimension. For that, one must rely on the analytical results above.

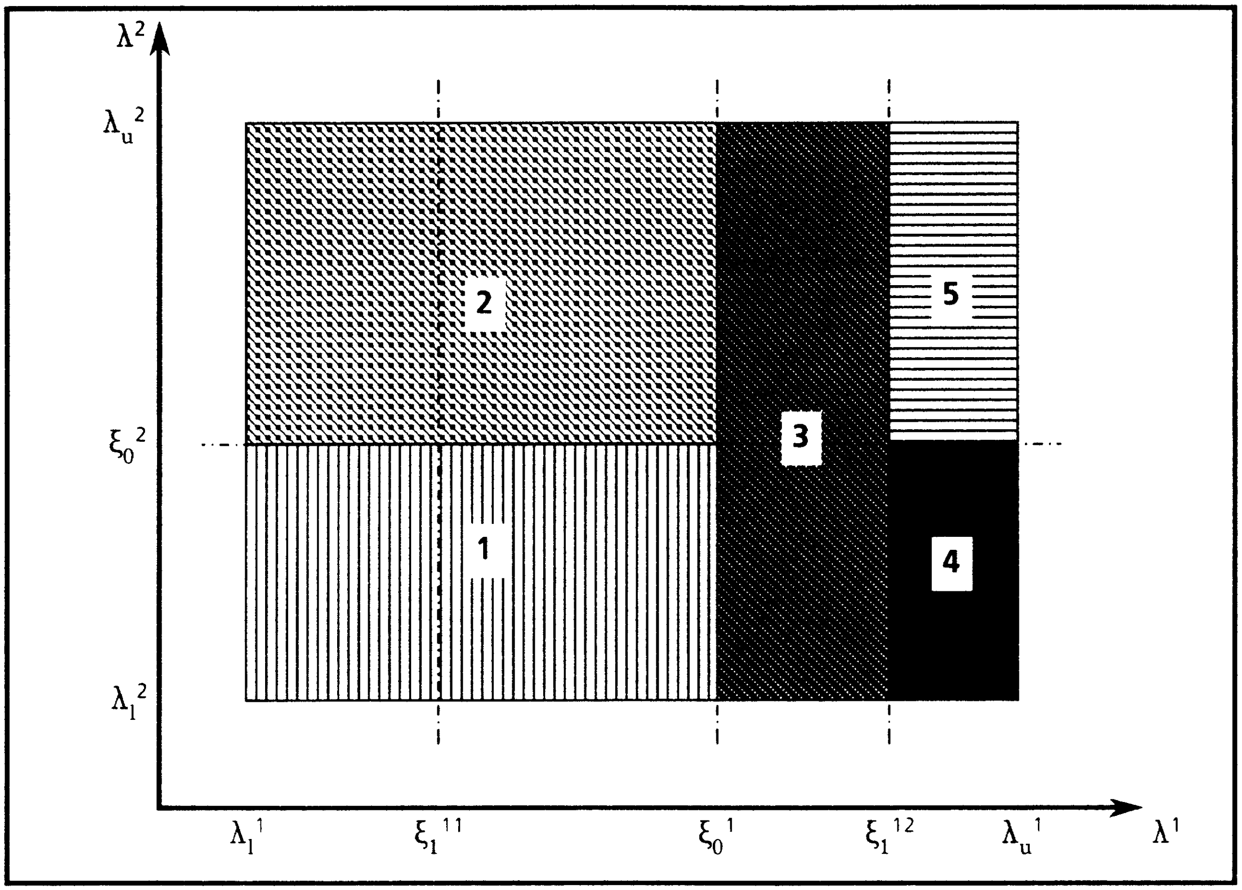


Figure 4.3: The discretization after the fourth partition

Section 5: A Numerical Example

In this section, we illustrate the application of the algorithms to a small problem. The problem we consider has simple recourse structure. That is, the recourse constraints are of the form $A_y x + Iy^{+} - Iy^{-} = \xi$, where I is the identity matrix. This type of stochastic programming problem has many practical applications. Moreover, Wets has developed an extremely efficient algorithm for solving such problems with discrete random variables using a variant of Dantzig and Beale's generalized upper bounding algorithm [15]. For these simple problems, there is great potential practical value for combining Wets algorithm with our discretization scheme.

Our sample problem is derived from the following scenario. A manufacturing plant is preparing to produce two products for the coming demand season. The production process requires two machines. There are 250 manufacturing hours available to product the required inventory. Producing the first product requires 2 hours on machine 1 and 2.5 hours on machine 2. Producing the second product requires 3 hours and 2.5 hours on each machine respectively. The total manufacturing costs are 500 dollars for product 1 and 700 dollars for product 2. In the event that demand exceeds inventory next season, the plant can manufacture the products during the demand season at a unit cost of 1000 dollars and 1500 dollars respectively. If inventory exceeds demand, the excess inventory goes to waste at a cost of 1200 dollars per unit.

The model of our sample problem is:

$$Min\ 5x_1 + 7x_2 + E_{\xi}[\,10y_1^{+} + 15y_2^{+} + 12y_1^{-} + 12y_2^{-}\,]$$

$$\begin{aligned} s.t.\quad & 2x_1 + 3x_2 \le 250 \\ & x_1 + x_2 \le 100 \\ & x_1 \quad + y_1^{+} \quad - y_1^{-} = \xi_1 \\ & \quad x_2 \quad + y_2^{+} \quad - y_2^{-} = \xi_2 \\ & x_1 \ge 0, x_2 \ge 0, y_i^{+} \ge 0, y_i^{-} \ge 0 \end{aligned}$$

The demand is modelled by the independent random variables, ξ_1 and ξ_2, which are in turn defined by the probability density functions shown in figure 5.1.

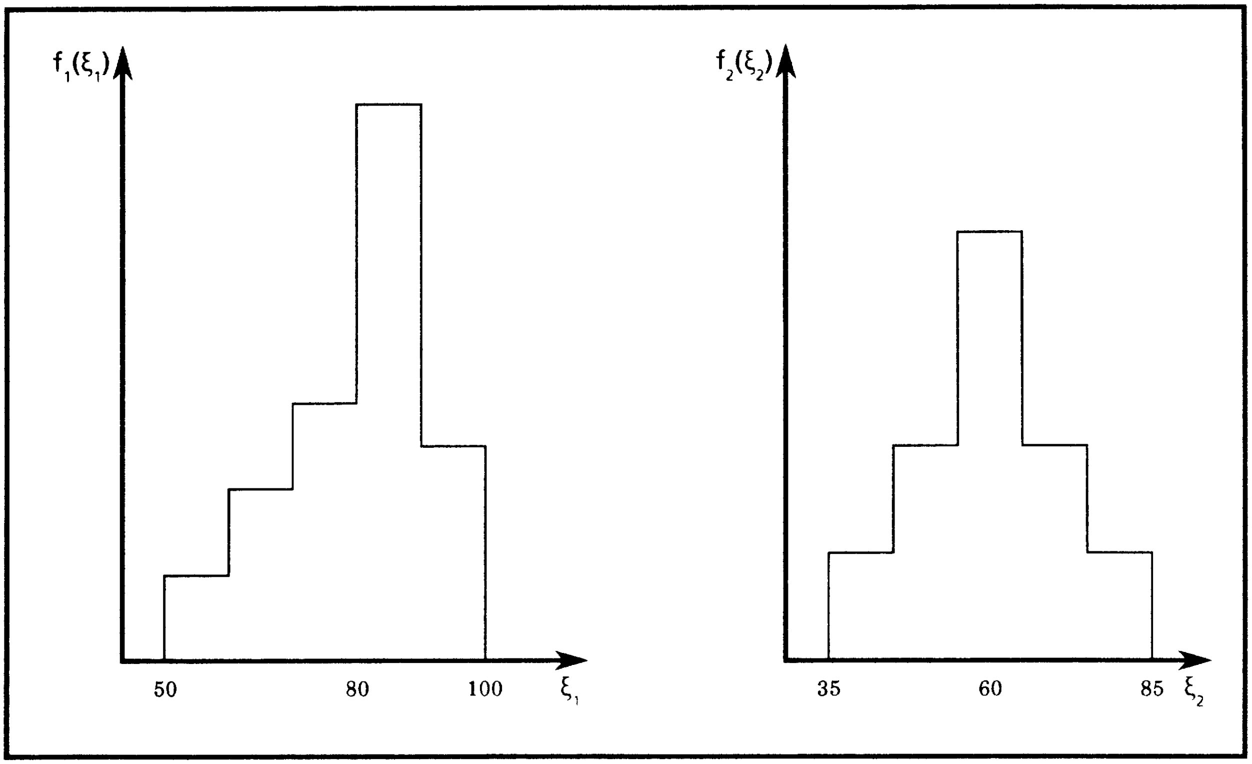


Figure 5.1: Demand Distribution Functions

These distribution functions may be written as:

$$F_1(\xi_1) = \frac{2}{300}(\xi_1-50)U_{50}^{\xi_1} + \frac{2}{300}(\xi_1-60)U_{60}^{\xi_1} + \frac{2}{300}(\xi_1-70)U_{70}^{\xi_1}$$

$$+ \frac{7}{300}(\xi_1-80)U_{80}^{\xi_1} - \frac{8}{300}(\xi_1-90)U_{90}^{\xi_1} - \frac{5}{300}(\xi_1-100)U_{100}^{\xi_1}$$

$$F_2(\xi_2) = \frac{1}{100}(\xi_2-35)U_{35}^{\xi_2} + \frac{1}{100}(\xi_2-45)U_{45}^{\xi_2} + \frac{2}{100}(\xi_2-55)U_{55}^{\xi_2}$$

$$- \frac{2}{100}(\xi_2-65)U_{65}^{\xi_2} - \frac{1}{100}(\xi_2-75)U_{75}^{\xi_2} - \frac{1}{100}(\xi_2-85)U_{85}^{\xi_2}$$

where $U_d^{\xi} = \{0 \text{ if } \xi < d \text{ and } 1 \text{ if } \xi \geqq d\}$.

The first step of algorithm 1 is to solve the expected value problem. $E_\xi(\xi_1) = 80$ and $E_\xi(\xi_2) = 60$. In practice we solve the modified dual of the problem, which, we recall, is exactly the dual problem in the first iteration. The primal and dual solution of the expected value problem is: $Z_{(EV)} = 1045$, $x^* = (35,60)$, $\pi^* = (2.5,0)$, $\gamma^* = (10,14.5)$. (Note that if we implemented this solution, we would produce 35 units of product 1 and 60 units of product 2, and the expected cost would be $\{5(35) + 7(60) + E_\xi[10(\xi_1 - 35)^+ + 12(\xi_1 - 35)^- + 15(\xi_2 - 60)^+ + 12(\xi_2 - 60)^-]\}*100 = \$125{,}600$.

We now proceed with algorithm 1. First set $r = 1$, and calculate $g_1(\xi_1(\lambda))$ and $h_1(\xi_1(\lambda))$ using parametric programming.

$$g_1(\xi_1(\lambda)) = \begin{array}{l} Max\, \gamma_1(\xi_1 - 35) + \gamma_2(60 - 60) - (10{,}15)\cdot\ \ (80{,}60)^T \\ s.t.\ \ -12 \le \gamma_1 \le 10, and\ -12 \le \gamma_2 \le 15 \end{array}$$

$$g_2(\xi_2(\lambda)) = \begin{array}{l} Max\, \gamma_1(80 - 35) + \gamma_2(\xi_2 - 60) - (10{,}15)\cdot\ \ (80{,}60)^T \\ s.t.\ \ -12 \le \gamma_1 \le 10, and\ -12 \le \gamma_2 \le 15 \end{array}$$

Figure 5.2a shows the plot of $g_1(\xi_1(\lambda))$ and $h_1(\xi_1(\lambda))$ as a function of $\xi_1(\lambda)$. Note that there are no vertices in the range $50 \leqq \xi_1 \leqq 100$. This implies that $\psi_1(\lambda) = -1250$ for all λ. We next check all possible cuts for the second partition index, $r = 2$. Figure 5.2b shows $g_2(\xi_2(\lambda))$ and $h_2(\xi_2(\lambda))$. There is one pivot point at $\xi_2 = 60$. This point corresponds to the expected value of ξ_2, so we need not execute step 2 of algorithm 1. Continuing with step 3, we calculate $\lambda'^* = 60$, and since $35 < 60 < 85$, this value corresponds to the only valid partition at this iteration. Thus, we apply it to obtain

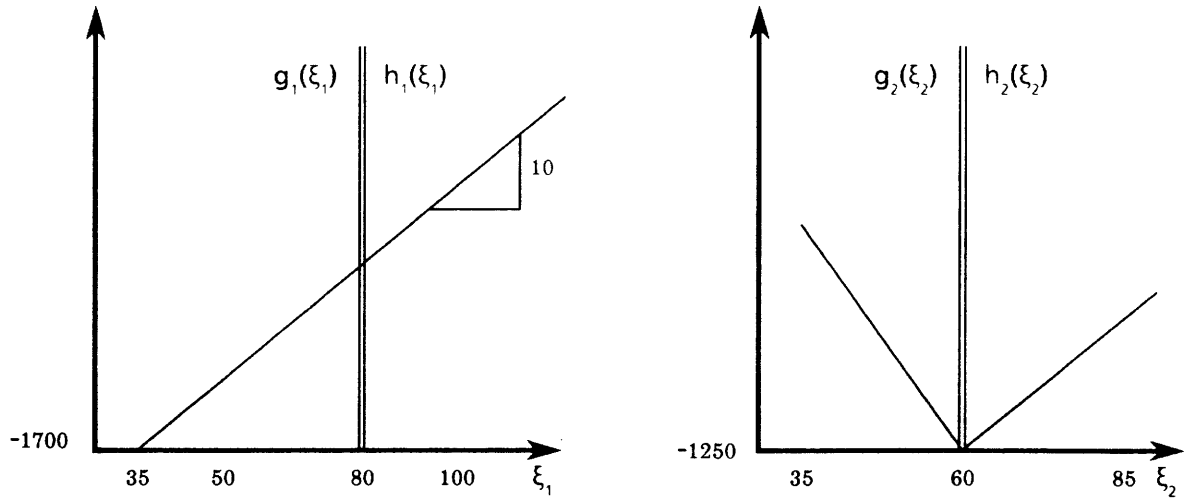


Figure 5.2a

Figure 5.2b

Figure 5.2: Plots of g_r and h_r

our second approximation. To generate our second problem, we must calculate $E_\xi[\xi_2|\xi_2 \leqq 60] = 51$, $p_1 = .5$, and $E_\xi[\xi_2|\xi_2 \geqq 60] = 69$, $p_2 = .5$.

The modified dual of our new approximation is:

$$Max\ -250\pi_1 - 250\pi_2 + (.5)80\gamma_1^1 + (.5)51\gamma_2^1 + (.5)80\gamma_1^2 + (.5)69\gamma_2^2$$

$$s.t.\quad -2\pi_1 - 2.5\pi_2 + (.5)\gamma_1^1 + (.5)\gamma_1^2 \leq 5$$

$$-3\pi_1 - 2.5\pi_2 + (.5)\gamma_2^1 + (.5)\gamma_2^2 \leq 7$$

$$\gamma^1 \epsilon \Gamma,\ \gamma^2 \epsilon \Gamma,\quad where\, \Gamma \equiv \{(\gamma_1, \gamma_2)|\ -12 \leq \gamma_1 \leq 10,\ -12 \leq \gamma_2 \leq 15\}$$

$$\pi_1 \geq 0,\ \pi_2 \geq 0$$

The solution to this problem is $Z = 1049.5$, $x^* = (48.5, 51.0)$, $\pi^* = (2.5,0)$, $\gamma^{1*} = (10,14)$, $\gamma^{2*} = (10,15)$. If we implemented this solution, the expected cost would be $115,200, an 8% saving over the expected value solution.

We define the functions $g_r^j(\lambda)$ as $g_r(\lambda)$ for the jth partition block. At this point, we have 2 partition blocks: block $1 = \{(\xi_1,\xi_2)|50 \leqq \xi_1 \leqq 100$, and $35 \leqq \xi_2 \leqq 60\}$, and block $2 = \{(\xi_1,\xi_2)|50 \leqq \xi_1 \leqq 100$, and $60 \leqq \xi_2 \leqq 85\}$. These functions are defined using equations (2.12) and (2.13). We continue solving the sequence of refined problems. Table 5.1 shows the sequence of solutions generated. Our final solution of (52.8,48.2) is within 0.2% of the optimal. This solution was obtained with the five partitions shown in figure 5.3.

First Stage Solution	Value	Expected Cost	partition (blk, indx,cut)
(35.0,60.0)	1045.0	1256.0	(1,2,60.0)
(48.5,51.0)	1049.5	1151.9	(1,2,51.0)
(55.6,44.4)	1066.9	1123.5	(1,2,44.4)
(52.2,47.8)	1077.0	1082.3	(4,1,52.2)
(52.8,47.2)	1077.5	1079.7	

Table 5.1: The Sequence of Solutions Generated with Algorithm 1

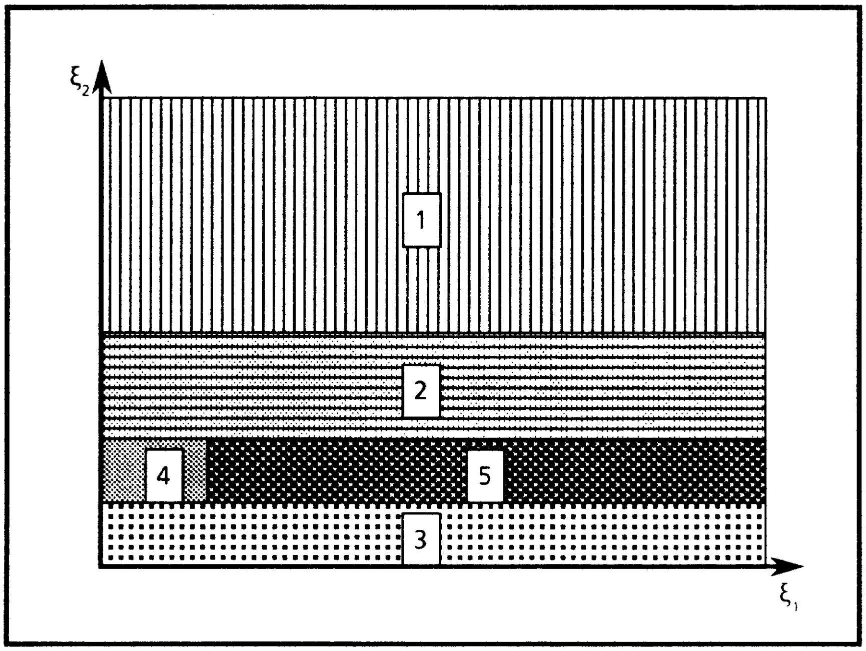


Figure 5.3: The Final Partition Structure

Conclusion

We analyzed discretization procedures for two-stage stochastic programming problems with fixed recourse and continuous stochastic right-hand sides. By assuming independence of the stochastic elements, we were able to develop two algorithms for successively refining a discretization of the problem. The deterministic equivalent of this discretized version of the problem can be solved using any algorithm for stochastic programming with recourse.

With the results presented here, it may no longer be advantageous to require that all the random variables to be discrete. The analyst has no way of knowing *a priori* which stochastic elements are most critical to his problem. Thus, he/she is unable to determine which elements should have a fine grid and which should have a coarse grid in his/her discretization. By allowing the algorithm to determine which elements to refine as it goes, we may use information gleaned during the solution process as it becomes available. In this way, we are assured of having a balanced grid structure throughout the entire procedure.

The first algorithm presented will determine a provably optimal discretization in finite time at each stage. However, it is difficult to implement for a general class of problems and it may require an excessive amount of time to solve. Thus, its value is theoretical for the most part. However, it does provide us some good insights into the discretization problem as well as providing a firm foundation upon which other heuristics may be built.

The second algorithm, on the other hand, is intuitively appealing, trivial to implement, and requires little time to solve. The partition derived from this algorithm will maximize the next dual simplex subgradient. Thus, it is natural to embed it in a dual simplex procedure for solving the two stage stochastic programming problem with random right-hand sides. Moreover, with minor modifications, the algorithm could be implemented for the general multistage stochastic programming problem.

References

[1] Birge, J.R., "The Value of the Stochastic Solution in Stochastic LP's with Fixed Recourse," Mathematical Programming 24 (1982) 314 – 325.

[2] Birge, J.R., and Louveaux, F.V., "A Multicut Algorithm for Two-Stage Stochastic Linear Programs," University of Michigan Department of Industrial and Operations Engineering Technical Report 85-15 (1985).

[3] Birge, J.R., and Wets, R.J.B., "Generating Upper Bounds on the Expected Value of a Convex Function with Applications to Stochastic Programming," University of Michigan Department of Industrial and Operations Engineering Technical Report 85-14 (1985).

[4] Garstka, S.J. and Rutenberg, D., "Computations in Discrete Stochastic Programs with Recourse," Operations Research 21 (1973) 112 – 122.

[5] Hiller, R. "Practical Approaches to Solving Stochastic Programming Problems with Complete Recourse and Random Right-Hand Sides," MIT OR Center (1985).

[6] Huang, C.C., Vertinsky, I., and Ziemba, W.T., "Sharp Bounds on the Value of Perfect Information," Operations Research 25 (1977) 128 – 139.

[7] Kallberg, J.G., White, R.W., and Ziemba, W.T., "Short Term Financial Planning Under Uncertainty," Management Science 28 (1982) 670 – 682.

[8] Madansky, A., "Inequalities for Stochastic Linear Programming Problems," Management Science 6 (1960) 197 – 204.

[9] Olsen, P., "Discretizations of Multistage Stochastic Programming Problems," Mathematical Programming Study 6 (1976) 111 – 124.

[10] Rockafellar, R.T. and Wets, R.J.B., "Continuous versus Measurable Recourse in N-stage Stochastic Programming," Journal of Mathematic Analysis and Applications 48 (1974) 836 – 859.

[11] Van Slyke, R. and Wets, R.J.B., "L-Shaped Linear Programs with Application to Optimal Control and Stochastic Programming," SIAM Journal on Applied Mathematics 17 (1969) 638 – 663.

[12] Walkup, D.W., and Wets, R.J.B., "Stochastic Programs with Recourse," SIAM Journal on Applied Mathematics 15 (1967) 1299 – 1314.

[13] Wets, R.J.B., "Stochastic Programming with Fixed Recourse: The Equivalent Deterministic Problems," SIAM Review 16 (1974) 309 – 339.

[14] Wets, R.J.B., "Stochastic Programming: Solution Techniques and Approximation Schemes," in A. Bachem, M. Groetschel, B. Korte, eds., Mathematical Programming: State-of-the-Art 1982, Springer-Verlag, Berlin (1983) 506 – 603.

[15] Wets, R.J.B., "Solving Stochastic Programs with Simple Recourse," Stochastics 10. (1983) 219 – 242.