A MULTICUT ALGORITHM FOR TWO-STAGE STOCHASTIC LINEAR PROGRAMS

John R. Birge
Department of Industrial and
Operations Engineering
The University of Michigan
Ann Arbor, MI 48109 USA

Francois V. Louveaux
Faculté des Sciences Economiques et Sociales
Facultés Universitaires N-D de la Paix
B-5000 Namur
Belgium
Technical Report 85-15

A Multicut Algorithm for Two-Stage Stochastic Linear Programs

John R. Birge*
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109
USA

François V. Louveaux
Faculté des Sciences Economiques et Sociales
Facultés Universitaires N-D de la Paix
B-5000 Namur
BELGIUM

Abstract: Outer linearization methods, such as Van Slyke and Wets's L-shaped method for stochastic linear programs, generally apply a single cut on the nonlinear objective at each major iteration. The structure of stochastic programs allows for several cuts to be placed at once. This paper describes a multicut algorithm to carry out this procedure. It presents experimental and theoretical justification of reductions in major iterations.

Keywords: stochastic programming, outer linearization, cutting plane methods.

*Supported by National Science Foundation Grant No. ECS-8304065.

1. Introduction

Two-stage stochastic linear programs have a deterministic equivalent program with convex objective function that can be solved by a variety of methods. The L-shaped method of Van Slyke and Wets [9] is a cutting plane or outer linearization technique for solving this program when the random variables have finite support. It has been extended to multi-stage stochastic linear and quadratic programs by Birge [1] and Louveaux [7], respectively. Their analyses showed the L-shaped algorithm to be an effective solution technique for a variety of examples. The structure of stochastic programs, however, allows the L-shaped method to be extended to include multiple cuts on the objective in each major iteration. This paper describes this procedure for two-stage stochastic linear programs. A multi-stage version has been proposed by Silverman [8].

In Section 2, we briefly describe the L-shaped algorithm and the problem structure. In Section 3, we present the multicut algorithm and, in Section 4, we discuss its efficiency in terms of bounds on the number of major iterations for general problems. The specific case of simple recourse problems is discussed in Section 5. Section 6 presents results of numerical experiments and the appendices provide illustrative examples of claims made in the text.

2. The L-shaped algorithm

The classical two-stage stochastic linear program with fixed recourse is the problem of finding

$$\min z = cx + E_{\xi} \min q(\omega)\cdot y \tag{1}$$

$$\text{s.t.} \quad Ax = b$$

$$T(\omega)\cdot x + W\cdot y = h(\omega)$$

$$x \geq 0 \qquad y \geq 0$$

where c is a known-vector in R^{n_1}, b a known-vector in R^{m_1}, ξ a random N-vector defined on the probability space $(\Omega, \mathcal{A}, P)$, and A and W are known matrices of size $m_1 \times n_1$ and $m_2 \times n_2$, respectively. W is called the recourse matrix.

For each ω, $T(\omega)$ is $m_2 \times n_1$, $q(\omega) \in R^{n_2}$ and $h(\omega) \in R^{m_2}$. Piecing together the stochastic components of the problem, we get a vector $\xi(\omega) = (q(\omega), h(\omega), T_1(\omega), \ldots, T_{m_2}(\omega))$ with $N = n_2 + m_2 + (n_2 \times n_1)$ components, where $T_i(\omega)$ is the i^{th} row of $T(\omega)$. Transposes have been eliminated for simplicity. E_ξ represents the mathematical expectation with respect to ξ .

Problem 1 is equivalent to the so-called deterministic equivalent program (D.E.P.):

$$\begin{aligned} \min z &= cx + \mathcal{Q}(x) \\ \text{s.t.} \quad Ax &= b \\ x &\geq 0 \end{aligned} \tag{2}$$

where

$$\mathcal{Q}(x) = E_\xi Q(x, \xi(\omega))$$

and

$$Q(x, \xi(\omega)) = \min_y \{ q(\omega)\cdot y \mid W\cdot y = h(\omega) - T(\omega)\cdot x, \ y \geq 0 \}$$

Properties of the D.E.P. have been extensively studied (Wets [10], Garstka and Wets [4]). Of particular interest for computational aspects is the fact that $Q(x, \xi)$ is a convex piecewise linear function of x and that $\mathcal{Q}(x)$ is also piecewise linear convex if ξ has finite support.

When T is non-stochastic, the original formulation (2) can be replaced by

$$\begin{aligned} \min z &= cx + \Psi(\chi) \\ \text{s.t.} \quad Ax &= b \\ Tx - \chi &= 0 \\ x &\geq 0 \end{aligned} \tag{3}$$

where $\Psi(\chi) = E_\xi \psi(\chi, \xi(\omega))$

and $\psi(\chi, \xi(\omega)) = \min \{q(\omega)\, y \mid Wy = h(\omega) - \chi,\ y \geq 0\}$

This formulation stresses the fact that choosing x corresponds to generating an m_2-dimensional tender $\chi = Tx$ to be "bid" against the outcomes $h(\omega)$ of the random events.

In this paper, we concentrate on algorithms for solving (2) or (3). Excluding algorithms for the specific simple recourse problem (see, e.g., Kall [6], Wets [11]), the basic method for solving (2) is the L-shaped algorithm due to Van Slyke and Wets [9] which is directly related to Benders' decomposition. For more details, see the discussion of algorithmic procedures in Wets [11]. The method consists of solving an approximation of (2) by using an outer linearization of $\mathcal{Q}$. Two types of constraints are sequentially added: (i) feasibility cuts (5) determining $\{x \mid \mathcal{Q}(x) < +\infty\}$ and (ii) optimality cuts (6) which are linear approximations to $\mathcal{Q}$ on its domain of finiteness.

<u>Assumption</u>: The random variable ξ has finite support.

Let $k = 1, \ldots, K$ index its possible realizations and let p_k index their probabilities.

<u>L-shaped algorithm</u>

Step 0. Set $s = t = \nu = 0$.

Step 1. Set $\nu = \nu + 1$. Solve the linear program (4) - (6).

$$\min z = cx + \theta \tag{4}$$

$$\text{s.t.} \quad Ax = b$$

$$D_\ell x \geq d_\ell \qquad \ell = 1, \ldots, s \tag{5}$$

$$E_\ell x + \theta \geq e_\ell \qquad \ell = 1, \ldots, t \tag{6}$$

$$x \geq 0, \quad \theta \in R.$$

Let (x^ν, θ^ν) be an optimal solution. If no constraint (6) is present, θ is set equal to $-\infty$ and is ignored in the computation.

Step 2. For k = 1, ..., K solve the linear program

$$\min \quad w^1 = ev^+ + ev^-$$

$$\text{s.t.} \quad Wy + Iv^+ - Iv^- = h_k - T_k \cdot x^\nu$$

$$y \geq 0 \qquad v^+ \geq 0 \qquad v^- \geq 0$$

where e = (1, ..., 1), until, for some k, the optimal value $w^1 > 0$. Let σ^ν be the associated simplex multipliers and define

$$D_{s+1} = \sigma^\nu T_k$$

and

$$d_{s+1} = \sigma^\nu h_k$$

to generate a feasibility cut of type (5). Set s = s+1 and return to Step 1. If, for all k, $w^1 = 0$, go to Step 3.

Step 3. For k = 1, ..., K solve the linear program

$$\min w^2 = q_k \cdot y \tag{7}$$

$$\text{s.t.} \quad Wy = h_k - T_k \cdot x^\nu$$

$$y \geq 0.$$

Let π_k^ν be the simplex multipliers associated with the optimal solution of Problem k of type (7). Define

$$E_{t+1} = \sum_{k=1}^{N} p_k \cdot \pi_k^\nu T_k \tag{8}$$

and

$$e_{t+1} = \sum_{k=1}^{N} p_k \cdot \pi_k^\nu h_k. \tag{9}$$

Let $w^{2\nu} = e_{t+1} - E_{t+1} x^\nu$.

If $\theta^\nu \geq w^{2\nu}$, stop, x^ν is an optimal solution. Otherwise, set t = t + 1, and return to Step 1.

Improvements on this algorithm have been given in two directions: (i) the study of cases in which Step 2 can be modified to solve only one linear program instead of N and (ii) the study of bunching and sifting procedures to reduce the work in

Step 3 (Garstka and Rutenberg [13]). We again refer to Wets [11] for a detailed account of these improvements.

In this paper, we propose to replace the outer-linearization of $\mathcal{Q}$ used in the L-shaped method by an outer-linearization of all functions

$$Q_k(x) = \min_y \{q_k \cdot y \mid Wy = h_k - T_k \cdot x,\ y \geq 0\}, \qquad (10)$$

of which $\mathcal{Q}(x)$ constitutes the expectation.

3. The Multicut L-shaped algorithm

The multicut L-shaped algorithm is defined as follows:

Step 0. Set $s = \nu = 0$ and $t_k = 0$ for all $k = 1, \ldots, K$.

Step 1. Set $\nu = \nu + 1$. Solve the linear program (11) - (13).

$$\min z = cx + \sum_{k=1}^{K} \theta_k \qquad (11)$$

$$\begin{aligned} \text{s.t.} \quad & Ax = b & & \\ & D_\ell\, x \geq d_\ell, & \ell = 1, \ldots, s & \qquad (12) \\ & E_\ell\, x + \theta_k \geq e_\ell, & \ell = 1, \ldots, t_k & \qquad (13) \\ & & k = 1, \ldots, K. & \end{aligned}$$

Let $(x^\nu, \theta_1^\nu, \ldots, \theta_K^\nu)$ be an optimal solution of (11). If no constraint (13) is present for some k, θ_k^ν is set equal to $-\infty$ and is ignored in the computation.

Step 2. As before.

Step 3. For $k = 1, \ldots, K$ solve the linear program (7).

Let π_k^ν be the simplex multipliers associated with the optimal solution of problem k.

If $\theta_k^\nu < p_k \cdot \pi_k^\nu (h_k - T_k \cdot x^\nu)$ (14)

define

$$E_{t_k+1} = p_k \cdot \pi_k^\nu \cdot T_k \qquad (15)$$

and

$$e_{t_k+1} = p_k \cdot \pi_k^{\nu} \cdot h_k \tag{16}$$

and

$$\text{set } t_k = t_k+1.$$

If (14) does not hold for any $k = 1, \ldots, K$, stop, x^{ν} is an optimal solution. Otherwise, return to Step 1.

We illustrate the differences and similarities between the multicut approach and the standard L-shaped algorithm in the example of Appendix A. The multicut approach is based on the idea that using outer approximations of all $Q_k(x)$ sends more information than a single cut on $\mathcal{Q}(x)$ and that, therefore, fewer iterations are needed.

4. Efficiency and bounds

The following dominance property can be established. Define $K_1 \cap K_2$ to be the constraint set of the stochastic program (2), where

$$K_1 = \{x \mid Ax = b,\ x \geq 0\}$$

and

$$K_2 = \bigcap_{\xi \varepsilon \Xi} \{x \mid \exists\ y \geq 0 \quad \text{s.t. } Wy = h(\xi) - T(\xi)x\}$$

where by assumption Ξ has finite support.

$\mathcal{Q}(x)$ is known to be piecewise linear, hence there exists a polyhedral decomposition of $K_1 \cap K_2$ into a finite collection of closed convex sets C_r, called the cells of the decomposition, such that the intersection of two distinct cells has an empty interior and such that either the function $\mathcal{Q}(x)$ is identically $-\infty$, or on each cell the function $\mathcal{Q}(x)$ is affine.

In the following proposition, we define a <u>major iteration</u> to be the operations performed between returns to Step 1 in both algorithms. <u>Simplex iterations</u> are the number of simplex algorithm pivots performed on any of the linear programs considered by the algorithms.

Proposition: Let $\{x^\nu\}$ be a sequence of points generated by the multicut algorithm and let $\{y^\nu\}$ be a sequence generated by the L-shaped algorithm. Then, if at all major iterations, x^ν and y^ν belong to the same cells of the decomposition, the number of major iterations needed by the multicut algorithm will be less than or equal to the number of major iterations of the L-shaped algorithm.

Proof: The multicut approach sends more information on the function $\mathcal{Q}(x)$ than the L-shaped algorithm. If the L-shaped algorithm stops for optimality, the multicut algorithm will do so also. The reverse is not true. ∎

Unfortunately, whether the iterate points belong to cells that are close to or far from the optimal point is partly a matter of chance. Therefore, the L-shaped method can conceivably do better than the multicut approach (the reverse is obviously also true) in terms of number of major iterations. We illustrate this by the example in Appendix B. Other examples where the multicut approach would do better than the L-shaped method can easily be constructed.

Since none of the methods is superior to the other in all circumstances, the efficiency of the two approaches is measured in terms of worst-case analysis on the number of major iterations.

Definition:

Let $b(\xi)$ represent the maximum number of different slopes of $Q(x, \xi)$ in any direction parallel to one of the axes for a given ξ, i.e. the maximum number of different cells (of the polyhedral decomposition of $K_1 \cap K_2$ relative to $Q(x, \xi)$ for a given ξ) encountered by any ray (parallel to one of the axes) originating at a point arbitrarily chosen in $K_1 \cap K_2$.

Define $b = \max_{\xi \varepsilon \Xi} b(\xi)$ to be the "slope number of the second-stage of (2)."

Theorem:

Let b be the slope number of the second stage of (2). Then, the maximum number of iterations for the multicut algorithm is

$$1 + K \cdot (b^{m_2} - 1) \tag{17}$$

while the maximum number of iterations for the L-shaped algorithm is

$$[1 + K \cdot (b - 1)]^{m_2} \tag{18}$$

where K is the number of different realizations of ξ.

Proof: If b is the slope number of the second stage of (2), in the worst-case $b(\xi) = b$ for all $\xi \varepsilon \Xi$, then the maximal number of different slopes for $\mathcal{Q}(x)$ in one direction is $1 + K \cdot (b - 1)$. The worst-case for the total number of slopes is when the effective number of slopes is equal to $b(\xi)$ in all directions for all ξ(as is the case in a simple recourse problem). Then, the total number of facets of $\mathcal{Q}(x)$ (or cells of the decomposition of $K_1 \cap K_2$) is equal to $[1 + K(b - 1)]^{m_2}$. The worst-case for the L-shaped algorithm is to consider all the facets of $\mathcal{Q}(x)$, which proves (18).
For the multicut approach, the worst-case is to send at each iteration a cut for only one among the K realizations of the random event. For each realization, namely each θ_k, there are at most b^{m_2} facets of $Q_k(x)$; so a total of maximum $K \cdot b^{m_2}$ cuts for all Q_k's are needed to describe $\mathcal{Q}(x)$. The result given in (17) follows from the fact that K cuts are sent at the first iteration. ∎

The maximal number of iterations has an immediate consequence on the size of the first-stage problems to be solved. While problems of smaller size are needed in the first iterations of the L-shaped method (m_1+1 constraints,

n_1+1 variables) as compared to the multicut (m_1+K constraints, n_1+K variables), the above theorem shows that the size of the problem is of the order of $(b-1)^{m_2} K^{m_2}$ in the worst-case for the L-shaped approach and $K \cdot (b^{m_2} - 1)$ for the multicut strategy. One can therefore expect the multicut approach to be especially efficient for problems where m_2 is large and where many cuts are needed.

In the next section, the number of facets for the particular case of simple recourse is given explicitly.

5. The Simple recourse case

The simple recourse problem is a particular case of the formulation (3) with non-stochastic matrix T where the function $\psi(\chi, \xi)$ is separable

$$\psi(\chi, \xi) = \sum_{i=1}^{m_2} \psi_i(\chi_i, \xi_i) \tag{19}$$

and

$$\psi_i(\chi_i, \xi_i) = \min \{q_i^+ y_i^+ + q_i^- y_i^- \mid y_i^+ - y_i^- = h_i - \chi_i,\ y_i^+ \geq 0,\ y_i^- \geq 0 \} \tag{20}$$

where

$$\xi_i = (q_i^+, q_i^-, h_i).$$

Assume that, for each i, ξ_i can take on J different values (where for simplicity of exposition J is assumed to be the same for all i).

Then, using the multicut approach consists of approximating the recourse function $\Psi(\chi)$ by the outer-linearization

$$\sum_{i=1}^{m_2} \sum_{j=1}^{J} \theta_{ij} \tag{21}$$

Due to the simple recourse property, only two cuts of the type (13) can be generated for each θ_{ij}, namely

$$\theta_{ij} \geq p_{ij} \cdot q_{ij}^- \ (\chi_i - h_{ij}) \tag{22}$$

and

$$\theta_{ij} \geq p_{ij} \cdot q^+_{ij} \; (h_{ij} - \chi_i) \qquad (23)$$

where p_{ij} denotes the probability of the j^{th} realization of ξ_i.

Introducing the slack variable u_{ij} in constraint (22),

$$p_{ij} \cdot q^-_{ij} \; (\chi_i - h_{ij}) + u_{ij} = \theta_{ij} \qquad (24)$$

and substituting θ_{ij} from (24) into (21) — (23), the simple recourse problem (19), (20) is equivalent to

$$\min \; cx + \sum_{i=1}^{m_2} \sum_{j=1}^{J} p_{ij} \cdot q^-_{ij}(\chi_i - h_{ij}) + \sum_{i=1}^{m_2} \sum_{j=1}^{J} u_{ij} \qquad (25)$$

$$\begin{aligned} \text{s.t.} \quad & Ax = b \\ & Tx - \chi = 0 \\ & u_{ij} \geq p_{ij} \cdot q_{ij}(h_{ij} - \chi_i) \qquad i = 1, \ldots, m_2, \quad j = 1, \ldots, J \\ & u_{ij} \geq 0 \qquad i = 1, \ldots, m_2, \quad j = 1, \ldots, J \end{aligned}$$

where $q_{ij} = q^+_{ij} + q^-_{ij}$.

From (25), we can derive the following algorithm.

Multicut algorithm for simple recourse problem.

Step 0. Set $\nu = t = 0$

Step 1. Set $\nu = \nu + 1$. Solve the linear program

$$\min \; z = cx + \sum_{i=1}^{m_2} \sum_{j=1}^{J} p_{ij} \cdot q^-_{ij} \cdot T_i \cdot x + \sum_{\ell=1}^{t} u_\ell \qquad (26)$$

$$\begin{aligned} \text{s.t.} \quad & Ax = b \\ & u_\ell \geq e_\ell - E_\ell \, x \qquad \ell = 1, \ldots, t \\ & u_\ell \geq 0 \qquad \ell = 1, \ldots, t \end{aligned}$$

Let (x^ν, u^ν) be an optimal solution. If $t = 0$, then u is ignored in the computation.

Step 2. For each $i = 1, \ldots, m_2$, and $j = 1, \ldots, J$, if the constraint

$$0 \geq p_{ij} \cdot q_{ij}(h_{ij} - T_i \cdot x^\nu) \tag{27}$$

is violated, define

$$E_{t+1} = p_{ij} \cdot q_{ij} \cdot T_i$$

and

$$e_{t+1} = p_{ij} \cdot q_{ij} \cdot h_{ij}$$

and set $t = t+1$.

The initial problem (27) involves m_1 constraints and n_1 variables. For this problem, the worst-case situation is when at each iteration, only one constraint (27) is violated in Step 2. Then, the maximal number of iterations is $J \cdot m_2+1$. To compare with the maximal number of iterations an L-shaped type algorithm would require to solve the same problem, note that for each i, the function $\Psi_i(\chi_i) = E\psi_i(\chi_i, \xi_i)$ contains J+1 facets and since $\Psi(\chi)$ is separable in i, it contains at most $(J+1)^{m_2}$ facets. This is precisely the worst-case upper bound on the number of iterations for an L-shaped type algorithm as in the Theorem of Section 4.

6. Numerical experimentation and conclusions

The L-shaped algorithm and the multicut method have been coded in FORTRAN in the codes NDREG and NDSEP respectively. NDREG is a two-stage version of the multi-stage code developed by Birge in [1] and described in [2]. NDSEP uses the same subroutines for linear program solutions, constraint generation and constraint elimination as NDREG. The subroutines to control where cuts are placed and to determine optimality have been modified in NDSEP to reflect the differences between the standard L-shaped method and the multicut approach.

The set of test problems and their size characteristics appear in Table I. The first four problems are small energy examples with varying objectives and constraints and the last example is a stochastic two-stage version of one of Ho and Loute's [5] staircase problems. These examples were chosen because of their applicability and the facet structure of their recourse functions.

The problems were solved using the FORTRAN-G compiler at The University of Michigan on an Amdahl 5860. The number of major iterations, simplex iterations and CPU seconds are given for each problem in Table II. Both NDREG and NDSEP used the bunching approach (Wets [11]) for solving second-period problems. They also both included the deletion of slack cuts which resulted in savings of up to twenty percent in CPU times.

The results in Table II illustrate the effectiveness of the multicut approach and some of its shortcomings. In each example, the number of major iterations is reduced. This is due to the passing of more information on each major iteration as noted above. A difficulty arises, however, because of the increased size of (11) - (13) over (4) - (6). Although (4) - (6) in the worst-case may have many more constraints than (11) - (13), program (11) - (13) is initially larger and, hence, requires more time to solve. This leads to the increased time in solving NRG4 by NDSEP. NDSEP, in fact, spends 2.8 more CPUs solving (11) - (13) than NDREG spends solving (4) - (6) on NRG4. This problem is an especially bad case because the original problem is so small that the addition of 27 extra constraints increases its size nine-fold and has a significant slowing effect.

These examples suggest that the multicut approach can lead to significant reductions in the number of major iterations. As indicated above, the worst-case advantage of the multicut approach in limiting major iterations is enhanced as m_2 increases in size,and the experiments show that the multicut approach is most effective when the number of realizations K is not significantly larger than the number of first period constraints n. When K is large relative to n, it may be advantageous to use a hybrid approach in which subsets of the realizations are grouped together to form a reduced number of combination cuts. The worth of this and other strategies is, however, problem dependent and should be demonstrated through experimentation in different and varied application areas.

Table 1. Problem parameters

Problem	Period 1 (A)			Period 2 (W)			Realizations
	n_1	m_1	ρ*	n_2	m_2	ρ*	K
NRG1	7	3	1.000	20	8	.375	3
NRG2	7	3	1.000	20	8	.375	3
NRG3	7	3	1.000	20	8	.375	9
NRG4	7	3	1.000	20	8	.375	27
SCAGR7.S2	36	16	.191	79	39	.092	8

*fraction of elements (excluding slack variable elements) which are nonzero

Table 2. Experimental results

Problem	NDREG			NDSEP		
	Major Iterations	Simplex Iterations	CPUs	Major Iterations	Simplex Iterations	CPUs
NRG1	10	117	.34	6	64	.23
NRG2	13	163	.49	9	92	.35
NRG3	14	196	1.26	8	121	1.11
NRG4	14	207	3.19	7	166	5.66
SCAGR7.S2	10	138	1.66	7	108	1.40

Appendix A

Assume that $Q(x, \xi) = \begin{cases} \xi - x & \text{if } x \leq \xi \\ x - \xi & x \geq \xi \end{cases}$

and that ξ can take on the values 1, 2, and 4, each with probability 1/3. Assume also $c \cdot x = 0$ and $0 \leq x \leq 10$.

Figure 1 represents the functions $Q_1(x)$, $Q_2(x)$, $Q_3(x)$, and $\mathcal{Q}(x)$. Since the first-stage objective cx is zero, Q(x) is also the function z(x) to be minimized. Assume the starting point is $x^1 = 0$. The sequence of iterations for the L-shaped method would be:

Iteration 1:

x^1 is not optimal; send the cut

$$\theta \geq 7/3 - x.$$

Iteration 2:

$x^2 = 10$, $\theta^2 = -23/3$ is not optimal; send the cut

$$\theta \geq x - 7/3.$$

Iteration 3:

$x^3 = 7/3$, $\theta^3 = 0$ is not optimal; send the cut

$$\theta \geq \frac{x+1}{3}.$$

Iteration 4:

$x^4 = 1.5$- $\theta^4 = 2.5/3$ is not optimal; send the cut

$$\theta \geq \frac{5-x}{3}.$$

Iteration 5:

$x^5 = 2$, $\theta^5 = 1$, which is the optimal solution.

Starting from $x^1 = 0$, the multicut approach would yield the following sequences:

Iteration 1: x^1 is not optimal; send the cuts $\theta_1 \geq \frac{4-x}{3}$, $\theta_2 \geq \frac{2-x}{3}$ and $\theta_3 \geq \frac{1-x}{3}$.

Iteration 2: $x^2 = 10$, $\theta_1^2 = -2$, $\theta_2^2 = -8/3$, $\theta_3^2 = -3$ is not optimal;

send the cuts $\theta_1 \geq \frac{x-4}{3}$, $\theta_2 \geq \frac{x-2}{3}$, $\theta_3 \geq \frac{x-1}{3}$.

Iteration 3: $x^3 = 2$, $\theta_1^3 = 2/3$, $\theta_2^3 = 0$, $\theta_3^3 = 1/3$ is the optimal solution.

Therefore, by sending separate cuts on $Q_1(x)$, $Q_2(x)$, and $Q_3(x)$, the full description of $\mathcal{Q}(x)$ is obtained in two iterations.

Appendix B

Let $n_1 = 1$, $K = 2$ be the realizations of with equal probability 1/2. For the first value of ξ, $Q(x, \xi)$ has two pieces, such that

$$Q_1(x) = \begin{cases} -x - 1 & \text{if } x \le -1 \\ 0 & \text{if } x \ge -1. \end{cases}$$

While for the second value of ξ, $Q(x, \xi)$ has four pieces such that

$$Q_2(x) = \begin{cases} -1.5\,x & \text{if } x \le 0 \\ 0 & \text{if } 0 \le x \le 2 \\ 2/7(x-2) & \text{if } 2 \le x \le 9 \\ x - 7 & \text{if } x \ge 9. \end{cases}$$

Assume also that x is bounded by $-20 \le x \le 20$ and $c = 0$. Starting from any initial point $x^1 < -1$, one obtains the following sequence of iterate points and cuts for the L-shaped method.

Iteration 1: $x^1 = -2$, θ^1 is omitted; new cut $\theta \ge -0.5 - 1.25\,x$.

Iteration 2: $x^2 = +20$, $\theta^2 = -25.5$; new cut $\theta \ge 0.5\,x - 3.5$.

Iteration 3: $x^3 = 12/7$, $\theta^3 = -37/14$; new cut $\theta \ge 0$.

Iteration 4: $x^4 \in [-2/5, 7]$, $\theta^4 = 0$.

If x^4 is chosen to be any value in [0,2] then the algorithm terminates at Iteration 4.

The multicut approach would generate the following sequence.

Iteration 1: $x^1 = -2$, θ_1^1 and θ_2^1 omitted; new cuts $\theta_1 \ge -0.5\,x - 0.5$.

$\theta_2 \ge -3/4\,x$.

Iteration 2: $x^2 = 20$, $\theta_1^2 = -10.5$, $\theta_2^2 = -15$; new cuts $\theta_1 \ge 0$. $\theta_2 \ge 0.5\,x - 3.5$.

Iteration 3: $x^3 = 2.8$, $\theta_1^3 = 0$, $\theta_2^3 = -2.1$; new cut $\theta_2 \ge 2/7\,(x - 2)$.

Iteration 4: $x^4 = 0.552$, $\theta_1^4 = 0$, $\theta_2^4 = -0.414$; new cut $\theta_2 \ge 0$.

Iteration 5: $x^5 = 0$, $\theta_1^5 = \theta_2^5 = 0$, STOP.

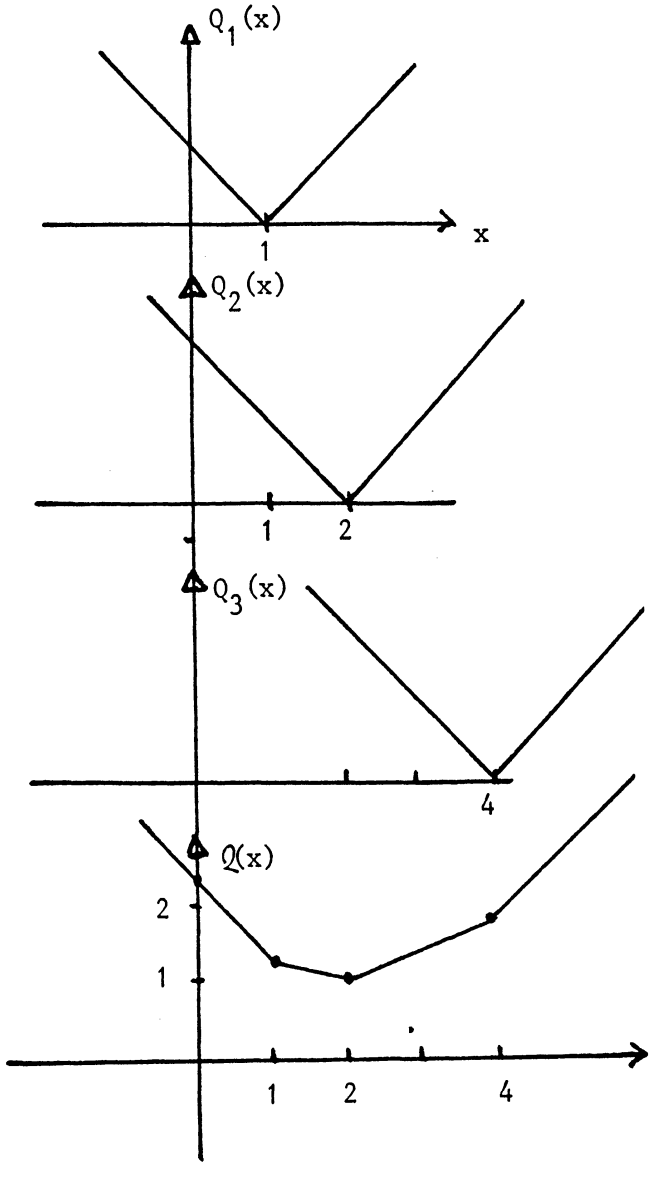

Figure 1. Recourse Functions

References

[1] J.R. Birge, "Decomposition and partitioning methods for multi-stage stochastic linear programs", Technical Report 82-6, Department of Industrial and Operations Engineering, The University of Michigan (Ann Arbor, MI, 1982); also to appear in Operations Research.

[2] J.R. Birge, "An L-shaped method computer code for multi-stage stochastic linear programs", in: Y. Ermoliev and R. J-B. Wets, eds., Numerical methods in stochastic programming, to appear.

[3] S.J. Garstka and D.P. Rutenberg, "Computation in discrete stochastic programs with recourse", Operations Research 21(1973) 112-122.

[4] S.J. Garstka and R. J-B. Wets, "On decision rules in stochastic programming", Mathematical Programming 7(1974) 117-143.

[5] J.K. Ho and E. Loute, "A set of staircase linear programming test problems", Mathematical Programming 20(1981) 245-250.

[6] P. Kall, "Computational methods for solving two-stage stochastic linear programming problems", Zeitschrift fuer angewandte Mathematik und Physik 30(1979) 261-271.

[7] F.V. Louveaux, "A solution method for multistage stochastic programs with application to an energy investment problem", Operations Research 28(1980) 889-902.

[8] J. Silverman, "An accelerated nested decomposition algorithm for the solution of multi-period multi-scenario LPs", unpublished notes, Department of Operations Research, Stanford University (Stanford, CA, 1983).

[9] R. Van Slyke and R. J-B. Wets, "L-shaped linear programs with application to optimal control and stochastic programming", SIAM Journal on Applied Mathematics 17(1969)638-663.

[10] R. J-B. Wets, "Characterization theorems for stochastic programs", Mathematical Programming 2(1972) 166-175.

[11] R. J-B. Wets, "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) pp. 506-603.