Optimization of Stochastic Planning Models II
Concepts and Theory

R.L. Clay, I.E. Grossman

EDRC 06-186-95

Optimization of Stochastic Planning Models II. Two-stage Successive Disaggregation Algorithm

September 7,1994

R.L. Clay[1] and I.E. Grossmann[2]
Dept of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213 U.S.A.

Abstract

In order to solve the production planning problem given in "Part I\ a successive disaggregation algorithm is developed bated on extensive ieasitivky analysis for guiding the decisions on how to repartition the aggregate probability space. The partitioning algorithm is guaranteed to converge lo the exact solution in a finite number of iterations, and has a highly parallel decomposition and computer implementation. Example problems are presented to demonstrate the solution technique. Results are compared with the alternative solution methods, variants of Benders decomposition schemes tailored to the dynamic staircase LP structure.

Keywords: production planning, stochastic programming, linear programming, aggregate models, successive disaggregation algorithm, approximation method.

1 Introduction

As was shown in "Part I" (Clay and Grossmann, 1994), the optimization of two-stage production networks (such as in Figure 1) with uncertain costs, supplies, and demands can be formulated as the following fixed-recourse stochastic LP:

$$\begin{aligned} \min \quad & mc\wedge + E\wedge clxJ & & (1) \\ \text{s.t.} \quad & A_l x_l = £i & & \\ & 4* + 4*2 = *2 & & \\ & 0 \le x_t \le U_{2_t} & \forall t = 1,\ldots,T, & \end{aligned}$$

where 0_2 is the stochastic parameter vector defined on the (stage-2) probability space (©,?",/>), and &2(0_2) and $c_2(0_2)$ are stochastic linear functions with constant coefficients. The triple *(0,J,P)* defining the probability space is composed of the (non-empty) event space (0), the a-ficld in *9* (J), and the probability measure on *f (P).* As discussed in "Pan I", problem (1) has been extensively studied (e.g., Birge, 1982,1985,1992; Dantzig

---

[1] Email address: ic3l@andiew.cmu.edo (Internet).
[2] Email address: igQo@andrew.cmu.edu (Internet).

and Glynn, 1989; Infangcr, 1991; Wets, 1989; Gassmann, 1990). In this paper, we propose a successive disaggregation approximation method for problem (1), reformulated as the certainty equivalent problem (P0) below.

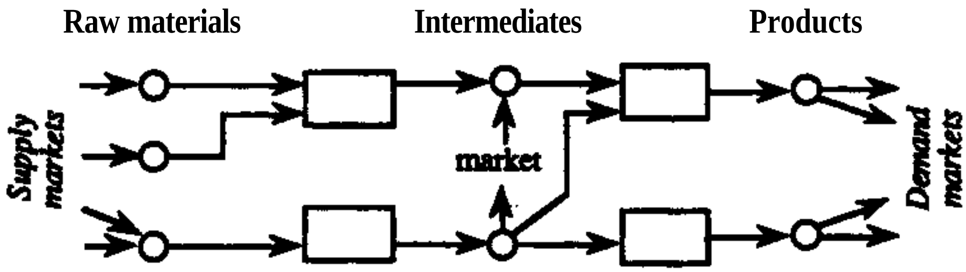

Figure 1. Simple production network with multiple feeds, intermediates, and products, all or part of which depend on uncertain market conditions governing supply, demand, and pricing.

2 Basic Sub-problems

Deterministic Equivalent Problem

Applying the certainty equivalent transfonnation (see Dantzig, 1987) to problem (1) with stochastic parameters c_2 and b_2 expanded over the stage-2 discrete probability index keK_y and rearranging leads to the following formulation.

(PO) min $z = c[x_x + \sum_{k*K} 2*p_n c\backslash_k x_{lk}$ (2a)

s.t. $\backslash x_x = b_x$ (2b)

$B_1 x_1 + A_2 x_{2k} = b_u$ VJfeeAT (2c)

$0 \leq x_1 \leq U_{x_1}$ (2d)

$0 \leq x_{2k} \leq U_{x_2}$ V*eAT, (2c)

where matrices B_x and A^ are fixed (i.e., £,* = ^ and A_u = ^ *VkeK).* The dual variables associated with LP (P0) are given in Table 1.

Table 1. Two-stage MSLP certainty equivalent dual variables.

Constraints	Dual Variables	Dimension
(2b)	μ	m,xl
(2c)	K	(mjXl)x|An
(2d)	σ_L, σ_U	n,xl, n,xl
(2e)	Pu >Puk	(n₂xl)x\K\An₂xl)x\K\

While the certainty equivalent formulation is straightforward, in practice (PO) can be intractable due to the exponential growth in variables and constraints (see Dantzig, 1987). This growth effect is exacerbated as additional time periods art included, making exact solution to die multi-stage equivalent of (PO) generally intractable.

Aggregate Model

As shown in Tart I", the idea behind the disaggregation algorithm is to circumvent the solution of (PO) with an aggregated LP, defined as:

(PA) min ^ - c f t + $\sum_{q \in Q}$ ft-ZL** (3a)

s.t. $A f t = 1$^ (3b)

$$B_1 x_1 + A_2 x_{2q} = \bar{b}_{2q} \quad \forall q \in Q \qquad (3c)$$

$$0 \le x_1 \le U_{x_1} \qquad (3d)$$

$$0 \le x_{2q} \le U_{x_2} \quad \forall q \in Q, \qquad (3e)$$

where $Pu \sim \sum_{k \in K_q} Ptk$ •' Q^Q*' K< C^» (4)

$\bigcup_{q \in}$

^ = X P A / P V (6)

$$\bar{c}_{2q} = \sum p_2 \cdot c_2 \cdot / V \qquad (7)$$

Here $q \in Q$ denotes the set of disjoint partitions containing events $k \in K_q$, whose union comprises the entire stage-2 (discrete) event space K. Additionally, (stage-1) feasibility "cuts" may be added to (PA) constraint set, as discussed below. As discussed in § 4, problem (PA) is solved with increasing resolution in the aggregate probability space to produce approximate solutions for z* and $x\backslash_y$ optimal for (PO).

A critical issue is the test for feasibility and optimality given any proposed solution for the stage-1 activity, x_x. To develop the partitioning algorithm we first consider the feasibility and optimality test problems (PI) and (P5), respectively.

Feasibility Test Problem

We start by examining the problem which occurs when x_x is assumed known (i.e., a value is obtained from the solution to (PA) which we need to test for feasibility). Starting with problem (PO) and assuming x_x known leads to the following formulation, which represents the stage-2 (recourse) component of (PO).

(PI)

$$\min_{x_{2k}} \; z_2 == \sum_{k \in K} p_{2k} c_{2k}^T x_{2k}$$

S.t. ^2*2* = *bit* "" ^1*1 Vk € *K*

$$0 \le x_{2k} \le U_{x_2} \qquad VkeK.$$

Given any proposed stage-1 solution *,, (PI) can be used to evaluate the feasibility with respect to all anticipated future outcomes. Solution to (PI) gives the following results: z_2, *2*> *K** $p_{u\%}$ POT, the *A^{k)* bases, and identification of any stage-2 infeasibilities.

Problem (PI) can be decomposed into N_K «\K\ independent sub-problems, a single-stage UP for each event keK_y greatly simplifying its solution. Furthermore, this decomposition suggests a natural parallel computing scheme with blocks of sub-problems distributed across processors. The objective, z_2, is the sum of the stage-2 objective function terms over all sub-problems *keK.* In a parallel computer this summation can be performed in $\log_2$ time. The overall speedup by parallel solution is expected to be nearly linear in the number of processors (i.e., nearly perfect speedup). Using a computer such as the Connection Machine CM-5 with 2048 processors, we would anticipate better than three orders of magnitude speedup over a (serial) workstation.

While problem (PI) can be decomposed and solved in parallel, the number of UP sub-problems can be exponentially large. We are therefore motivated to develop a variant of the algorithm which tests for feasibility in a reduced space. Given any proposed solution jCj and stage-2 RHS value b_2, the feasibility can be determined by the following sub-problem:

(P2)	min *u*	**duals**
	S.t. $A_2 x_2 = b_2 - B_1 x_1$	$\lambda \in \Re^{m_2}$
	-jt2-u£0	*PL****[1]
	Xt-u-U^ZO	*Pue***[2]*

where a solution u > 0 (u £ 0) implies (PO) is infeasible (feasible). In order to include the most constraining RHS terms, problem (P2) is expanded to the full (PO) constraint space by reformulating it as the following max-min problem which determines the worst potential infeasibility over all RHS realizations:

(P3) max min u

$$
\begin{aligned}
\text{s.t.} \quad & A_2 x_2 = \beta - B_1 x_1 \\
& \beta_L \leq \beta \leq \beta_U \\
& -x_2 - u \leq 0 \\
& x_2 - u - U_{x_2} \leq 0 \\
& \beta \in \{b_{2k} : k \in K\} \\
& x_2 \in \Re^{n_2},
\end{aligned}
$$

where p_L = minfe,}, p_v = max$\{b_{2k}\}$.

Again, a solution u>0 (u £ 0) implies (P0) is infeasiblc (feasible). Let /T denote an optimal solution to (P3). In order for /T to be a valid solution to die (P0) feasibility test problem, it must correspond to an event in the probability space, such that:

$$\beta^* \in \{b_{2k} : k \in K\}. \tag{8}$$

Since the "most infeasible" solution to (PI) will correspond to an extreme point (Swaney and Grossmann, 1985), condition (8) is met implicitly when the joint probability space is formed from the intersection of independent probability spaces. That is, when:

$$K = \Theta = \Theta_x \times \cdots_N \times_S \Theta, \tag{9}$$

then $P_i^* = P_u{}^{oT} ft = Pui$ Vf«l$_f$...$_f$ni$_{2i}$

and all P_L I p_v combinations e {fe^ : k € K). Hence when condition (9) applies then P^* e{b_{2k}: keK], satisfying[3] condition (8), and problem (P3) is a valid feasibility test for (P0).

When condition (9) (and hence (8)) is satisfied, problem (P3) can be reformulated to eliminate the nested (max/min) optimization problem by explicitly representing the optimality conditions of the min problem within the max problem (Grossmann and Floudas, 1987). Recasting (P3) in this manner leads to the following mixed integer formulation.

---

[3] If condition (8) is not implicitly satisfied (i.e., non-independent probability space), then problem (P3) can be reformulated by introducing integer variables to explicitly enforce (8). Alternatively, one may utilize the result that (P3) (minus condition (8)) is a relaxation of the feasibility test problem such that a feasible solution to the "reduced" (P3) implies feasibility for (P0). We have restricted our attention to the solution to independent probability space problems.

(P4) max u

$$
\begin{aligned}
\text{s.t.}\quad & A_2 x_2 = \beta - B_1 x_1 \\
& 1 - \sum_{j=1}^{n_2} \rho_{Lj} - \sum_{j=1}^{n_2} \rho_{Uj} = 0 \\
& A_2^T \lambda - \rho_L + \rho_U = 0 \\
& \rho_L - M y_L \le 0 \\
& \rho_U - M y_U \le 0 \\
& s_L = u + x_2 \\
& s_U = U_{x_2} - x_2 + u
\end{aligned}
$$

s_L-U_H(l-y_L)ZO

$$
\begin{aligned}
& s_U - U_{x_2}(1 - y_U) \le 0 \\
& y_L + y_U \le 1 \\
& \beta_L \le \beta \le \beta_U
\end{aligned}
$$

X unrestricted

$$
\begin{aligned}
& \rho_L,\ \rho_U,\ s_L,\ s_U \ge 0 \\
& y_L, y_U \in \{0,1\}^{n_2},
\end{aligned}
$$

where $p_L = \min_{k \in K}$ {fr2ik}, ^ $= \max_{k \in K}$ {b2k],

and conditions (9) and/or (8) hold. Note that evaluating the bounds fi_L and p_v is greatly simplified by the assumption of independent probabilities in (9).

When only the feasibility of a proposed (stage-1) solution x_x is in question (and condition (8) holds implicitly), then the reduced-space problem (P3) or equivalently (P4) can be solved We use this approach to generate the feasibility "cuts" which augment (PA) and enforce feasibility for subsequent aggregate solutions (see also § 4). To generate a feasibility cut for problem (PO) we first examine the (KKT) optimality conditions for problem (P2). The Lagrange function for (P2) is defined as:

$$\mathcal{L} = u + \lambda^T\left[A_2 x_2 - \beta + B_1 x_1\right] + \rho_L^T\left[-x_2 - u\right] + \rho_U^T\left[x_2 - U_{x_2} - u\right],$$

forany fie{b2k: $k \in K\}$.

The stationarity conditions imply that:

$$\frac{\partial \mathcal{L}}{\partial u} = 0 = 1 - \sum_{j=1}^{n_2} \rho_{Lj} - \sum_{j=1}^{n_2} \rho_{Uj},$$

and $$\frac{\partial \mathcal{L}}{\partial x_2} = 0 = A_2^T\lambda - \rho_L + \rho_U.$$

Hence the Lagrange function reduces to:

$$\mathcal{L} = \mathcal{L}(x_1) = \lambda^T[-\beta + B_1x_1] - \rho_U^T U_{x_2} \quad \beta \text{ specified.}$$

Bom constraint feasibility and complementary slackness conditions, we know that $C = u^*$ at some optimal j£. And since feasibility requires that u £ 0, we define the *,-space feasibility cut as:

$$\lambda^T[-\beta + B_1x_1] - \rho_U^T U_{x_2} \le 0.$$

By analyzing the dual solution of (P3) or (P4) we can ascertain the most violating constraints, and hence set /J, A, and p_v for the feasibility cut. As discussed in § 4, problem (PA) is augmented as necessary with the feasibility cuts until (PO) either becomes feasible or is determined to be strictly infeasible.

When an exact upper bound (z_{UB} = z\ + z\) and /or the complete (stage-2) solution are required, problem (PI) must be solved *VkeK.* However, since that is in general prohibitive, alternative methods of generating the upper bound are of interest. Among the alternatives are the methods proposed by Birge and Wallace (1986,1987), Birge and Wets (1985), and Edmundson-Madansky (Madansky, 1960). One might also consider a sampling-based approach to solving (PI), such as that suggested by Dantzig and Glynn (1989) or Infanger (1991). For case-1 or -2 problems (see "Part I" for discussion on classification) we propose another alternative based on a reformulation of problem (PI). When both costs and RHS terms are stochastic (i.e., case-3 problems), the reformulation does not lead to an upper bound on (PI). In the reformulated problem (Rl) an aggregate stage-2 activity vector is used, thus avoiding the \K\ LP sub-problems associated with (PI).

(Rl)
$$\begin{aligned} \max_{\beta}\ \min_{\psi}\ & z'_2 = \sum_{k\in K} p_{2k} c_{2k}^T \psi = \bar{c}_2^T \psi \\ \text{s.t.}\ & A_2\psi = \beta - B_1x_1 \\ & \beta_L \le \beta \le \beta_U \\ & 0 \le \psi \le U_{x_2} \\ & \beta \in [b_{2k} : k \in K) \\ & \psi \in \Re^{n_2}, \end{aligned}$$

where $p_L = \min_{k\in K}\{b_{2k}\}$, ft, $= \max_{k\in K}\{b_{2k}\}$,

jCj known

(PI) is a case-1 problem.

Note that (Rl) can be reformulated as an MILP problem similar to (P4), and then solved with conventional methods. Additionally, we can define the (independent) "event" sub-problems associated with (PI) and (Rl) as follows:

(PU) $$\min \ z_{2k} = p_{2k}c_{2k}^T x_{2k} = p_{2k}\bar{c}_2^T x_{2k}$$

$$\text{s.t.} \quad A_2 x_{2k} = b_{2k} - B_1 x_1$$
$$0 \le x_{2k} \le U_{x_2},$$

where (PI) is a case-1 problem.

(Rl*) $$\min_{\psi} \ z\wedge \ p_{2k}c_{2k}^T \psi = p_{2k}\bar{c}_2^T \psi$$

$$\text{s.t.} \quad A_2 \psi = \beta^* - B_1 x_1$$
$$0 \le \psi \le U_{x_2}$$
$$\psi \in \Re^{n_2},$$

where /T is optimal for (Rl).

The following relationships tie the feasibility test problems with their sub-problems as defined above:

$$z_2^* = \sum_{k\in K} z_{2k}^* \qquad \textbf{(10a)}$$

$$z_2^{\prime *} = \sum_{k\in K} z_{2k}^{\prime *}. \qquad \textbf{(10b)}$$

We note that in practice solving (Rl) by the decomposition to (Rl*) is unnecessary. We present the sub-problem (Rl*) here to facilitate the following discussion.

Theorem 2. Let *i* and $i\wedge^*$ denote feasible / optimal solutions to problems (PI) and (Rl), respectively. For case-1 problems (i.e., fixed cost, stochastic RHS), the solution to (Rl) is a valid upper bound on (PI), such that $z'_2{}^*$ £ z\.

Proof Far both (PI) and (Rl) the cost vector is fixed such that $c_n = c_2 = \bar{c}_2$. Let $z\backslash_k$ and $z_{2k}^{\prime *}$ denote the feasible / optimal solutions to sub-problems (PI*) and (Rl*), respectively.

The RHS vector /T in (Rl) corresponds to the selection of one particular value b_{lk}\ *keK* that defines the "worst" objective function value for (Rl), and correspondingly for (PI) and (Rl*). It therefore follows that 3 x^*_u for each (PI*) *VkeK* such that z^C*^) ≤ ^{z}*ű(V*)-* Inductively applying this inequality over all *keK* along with equation (10) relating the sub-problems to the non-decomposed problems (PI) and (Rl) gives *z* £ z'_2\ //Q.E.D.

Corollary 4. For case-2 problems (i.e., stochastic cost, fixed RHS), the solution to (Rl) is a valid upper bound on (PI), such that z£* £ z\.

Proof Since Afe* =6fe =/*\ problem (Rl) then reduces to an aggregate model (with |Cl = 1) of (PI). By application of Corollary 2 (see "Part I", § 8), it follows that z'^m_2 £ z_2\ //Q£.D.

Optimality Test Problem

We now consider the problem that results from (P0) by assuming x_{lh} known and Xj unknown, leading to the following formulation which represents the stage-1 component of(P0).

(P5) $\min_{*1} \; z_x = c[x_x$

$$
\begin{aligned}
\text{s.t.}\quad & A_1x_1 = b_1 \\
& B_1x_1 = b_{2k} - A_2x_{2k} \qquad\qquad \forall k \in K \\
& 0 \le x_1 \le U_{x_1}
\end{aligned}
$$

Solution to (P5) gives the following results: z_p x_{1f} /i, X_{kf} cx$_L$, $G_{u\%}$ and the *%* basis. Given any proposed stage-2 (feasible) solution x_{lk}> (PS) can be solved to find the stage-1 contribution z, to the objective $z = z_l+z_2$. Furthermore, any feasible solution set z_x and z_2 defines an upper bound on (PO) such that $z_{UB} = z_x+z_2t\ z$\

Since the (P5) constraints are reproduced over all events *keK it* can become intractably large. When the transition matrix £> is fixed (i.e., fl^B, V*€#) the problem can be reformulated to reduce the problem size. The approach (similar to the (P2) reformulation above) comes from the realization that the solution to (P5) will correspond to a vertex or extreme point, dictated by the most constraining RHS terms. The alternative formulation for (P5) follows.

(P6) $\max_{\delta} \min_{x_1} \quad z, = c[x_x$

s.t. $A_l x_l = b_x$

$$B_1 x_1 = \delta$$
$$0 \le x_1 \le U_{x_1}$$
$$\delta_L \le \delta \le \delta_U,$$

where $\delta_L = \min_{k \in K}\{\delta_k\}, \ \delta_U = \max_{k \in K}\{\delta_k\},$

$$\delta_k = b_{2k} - A_2 x_{2k}.$$

Thus the limits on 5 are the minimum and maximum terms over the RHS vector from the set of events *k* e *K*.

In order to avoid having to specify all *x^* Vie € *K* prior to solving (P6), we propose an alternative formulation which circumvents this specification.

(P7) max min $z, = C_,^T JC,$

s.t. $i4,x, = bi$

$$B_1 x_1 = \beta - A_2 \psi$$
$$0 \le x_1 \le U_{x_1}$$
$$0 \le \psi \le U_{x_2}$$
$$\beta_L \le \beta \le \beta_U$$
$$\psi \in \Re^{n_2},$$

where $fi_L = \min_{*c*}\{fc_{2t}\}, \ p_v = \max_{k \in K}\{fc_{2ik}\}.$

Similar to problem (P3) above, (P7) can be reformulated to eliminate the nested (max/min) optimization problem. Recasting (P7) in this manner leads to the following mixed integer formulation.

(P8) max $z,$

s.t. $z, \pounds c,^r x,$

$$A_1 x_1 = b_1$$
$$B_1 x_1 = \beta - A_2 \psi$$
$$c_1 + A_1^T \mu + B_1^T \lambda - a_L + a_V « 0$$
$$\sigma_L - M y_L \le 0$$
$$\sigma_U - M y_U \le 0$$
$$s_L = x_1$$

$$s_U = U_{x_1} - x_1$$
$$s_L - U_{x_1}(1 - y_L) \le 0$$
$$s_U - U_{x_1}(1 - y_U) \le 0$$
$$y_L + y_U \le 1$$
$$0 \le x_1 \le U_{x_1}$$
$$0 \le \psi \le U_{x_3}$$
$$\beta_L \le \beta \le \beta_U$$
/i9 X unrestricted
$$\sigma_L,\ \sigma_U,\ s_L,\ s_U \ge 0$$
$$y_L, y_U \in \{0,1\}^{n_1},$$

where fi_L = minfa*}, p_u = max$\{b_{2k}\}$.

Note that in this case we obtain a new estimate for x_x which can be used to test convergence.

Once we have solved problems (PI) and (PS), or their reformulated equivalents, an upper bound on the exact solution is given by z_{UB}-z^zl^z* or z'_{UB} = z* + %"*z_m ^ z\ if (Rl) is solved in place of (PI). We note that the upper bounding method via solution to (Rl) proposed above for case-1 and -2 problems may result in a non-zero gap between the upper and lower bounds even at the exact solution x[. Consequently, it may be necessary to refine the upper bound provided by (Rl).

3 Sensitivity Analysis Procedure for Partitioning

Central to the successive disaggregation method given in "Part I" (Clay and Grossmann, 1994) is the means by which the aggregate model is repartitioned when further refinement of the solution is required. In this section we consider sensitivity analysis of the aggregate problem (PA) solution in the context of repartitioning. We propose a strategy for repartitioning the aggregate probability space based on sensitivity analysis. The combined sensitivity analysis and repartitioning scheme comprise the successive disaggregation algorithm (SRO) presented in § 4.

Before developing the mathematics underlying the sensitivity analysis and repartitioning scheme, it is helpful to consider the matter at a conceptual level in order to place the development in context. The successive disaggregation algorithm can be viewed as an adaptive strategy which refines the aggregate solution space according to the sensitivities of the aggregate solutions combined with analysis of the potential partition splits along single (stochastic parameter) dimensions. With each solution to the aggregate LP (PA), the sensitivities to RHS and/or cost partition-mean changes are updated from the dual solution. Similarly, the potential partition splits (i.e., disaggregation steps) change according to the evolution of the partitioning. With each subsequent solution to (PA), the information is updated and the combined analysis guides the repartitioning to refine the aggregate probability space so as to add resolution in the area projected to be the most influential (i.e., projected to maximally change the next aggregate solution $\bar{z}^*_Q$).

To illustrate the matching of partition splits (i.e., disaggregation steps) to projected changes in the aggregate solution, $T_{Q\%}$ consider a single dimension of the stochastic vector, 0,, with four discrete values denoted 0_n through 0_{f4}. Assume that $T_Q = z_u <> z^*$ (as per case-1 problems), and that we therefore wish to repartition the 0 probability space so as to maximally increase $\bar{z}^*_Q$. Let the probabilities be such that the partition mean, $\bar{0}$^, lies between the values of events 2 and 3 (i.e., $0_n < \bar{0}_{\%} < 0_{/3}$), as shown in Figure 2. The (ordered) set of 0 events can be delineated according to the values relative to $\bar{0}$^, forming "positive" and "negative" subsets, $K^* = [0_m 0_{I4})$ and $AK = \{0_n, 0_{/2}\}$, respectively. Projecting the split of the partition at the mean, $\bar{0}_k$, leads to new partition "positive" and "negative" means with corresponding differences from the previous mean, denoted $A\bar{0}$£ and $4\bar{0}$£, respectively. From the analysis of the solution to (PA), we know the sensitivity of the aggregate solution, V_Q, to changes in RHS terms and (via the chain rule) to $\bar{0}$. Combining the projected change in $\bar{0}$^ with the sensitivity information leads to a predicted (maximally increasing) change in ?g, denoted Al£ in Figure 2. Making the split accordingly (forming a new partition) and resolving the disaggregated (PA) then leads to results as shown in Figure 3. The sensitivity analysis, repartitioning, and (PA) solution cycle is repeated, selecting partition q (and in general dimension /) so as to maximize $A\bar{z}$^ until the algorithm converges to the exact solution within a predefined tolerance.

As discussed in "Part I", the strategy for solving (PO) by successive disaggregation depends upon the type of uncertain parameters in the particular problem instance. We restrict our attention to problems with uncertainties in cost coefficients and RHS terms, and consider three cases delineated by the type of uncertain parameters. The subscript indicating stage-2 terms has been dropped to simplify the notation in the following discussion. Unless otherwise noted, all terms in the sensitivity analysis denote stage-2 variables and parameters.

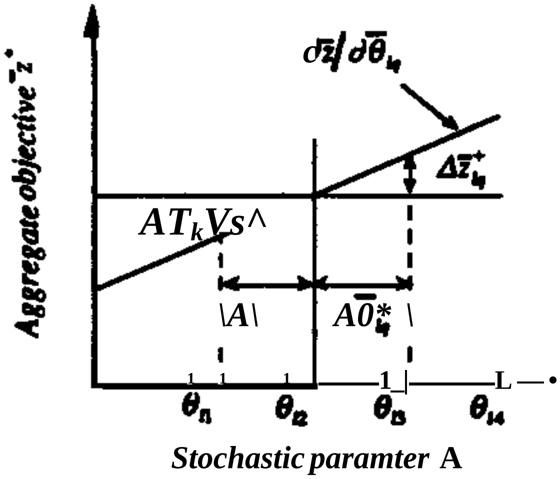

Figure 2. Geometrical interpretation of iepartitioning concept in single stochastic parameter dimension / for single partition *q*.

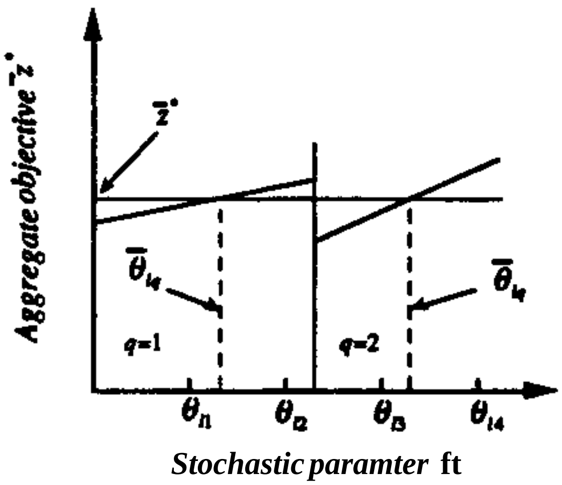

Figure 3. Geometrical interpretation of repartitioning concept in single stochastic parameter dimension / after splitting partition indicated in Figure 2. New partition means are projected from previous analysis, while sensitivities are updated from the solution of (PA) using two partitions.

<u>Case 1: Fixed costs, stochastic RHS</u>

Let z* be the optimal solution to the certainty equivalent (full AT-space) problem (PO). Further, let $N_{\S}$ and N_f denote dim(0) and |Q|, respectively. When costs are fixed the aggregate sub-problems (PA) are convex and thus give a lower bound $T_Q = z_u \leq z^*$ (see Tart I" for discussion on aggregate bounds). We seek the change in partitioning (i.e., the disaggregation step) which induces the maximum increase in z^. We approximate the change in the lower bound *z^* relative to a change in the stochastic parameter vector *0* around a solution point of (PA) as:

$$\Delta z_{LB} = \frac{\partial z_{LB}}{\partial \bar{\theta}} \Delta \bar{\theta} = \sum_{q \in Q} \sum_{i=1}^{N_q} \frac{\partial z_{LB}}{\partial \bar{\theta}_{iq}} \Delta \bar{\theta}_{iq}. \tag{11}$$

For the general case $b = b(9)$ and the RHS mean is defined as:

$$\bar{b}_{iq} = \sum_{*\bullet\ll\bullet} p_k b_{ik}(\theta_k) \Big/ p_q, \tag{12}$$

where P*⁼Z ttff P* **(13)**

and ^C^LL*;-*. **(14)**

Applying die chain rule and expanding over (stage-2) row terms gives:

$$\frac{\partial z_{LB}}{\partial \bar{\theta}} = \sum_{i=1}^{m_2} \frac{\partial z_{LB}}{\partial \bar{b}_i} \frac{\partial \bar{b}_i}{\partial \bar{\theta}} : \bar{\theta} \in \Re^{N_\theta \times N_q}, \ \bar{b}_i \in \Re^{N_q}. \tag{15}$$

The sensitivity of the lower bound to the RHS mean ($Bz_{LM} ld\bar{b}_x$) is available from the solution to (PA). The sensitivity of the RHS mean to die stochastic parameter mean can be obtained from the definitions of the partition means, where:

$$\bar{\theta}_{iq} = \sum p_k \theta_{ik} \Big/ p_q. \tag{16}$$

The corresponding sensitivity of the stochastic parameter mean to the discrete event values is:

$\frac{\partial \bar{\theta}_{iq}}{\partial \theta}$ = £*-:*€*,. **(17)**

When the stochastic function $b_t(0)$ is linear with constant coefficients we have:

b^aje, **(18)**

where[4] *b e* 9T*, *0 e* 9i$^{N'}$+^{x} f and *a e ftw*"*.* Expanding terms gives:

fy = T^cttf^/ Vi = l,..M m2» **(19)**

1*0

[4] **The stochastic vector *0* is augmented by 1 to account for the constant term, such that 8_{0k} • 1 and a_{i0} is the constant for t$_a$ *VkeK.***

and hence $$\frac{\partial b_{ik}}{\partial \theta_{lk}} = a_{il} = \frac{\partial b_i}{\partial \theta_l} = constant \quad \forall k \in K. \tag{20}$$

Substituting (19) into (12) gives:

$$\bar{b}_{iq} = \sum_{k\in K_q} p_k \sum_{l=0}^{N_q} \alpha_{il}\theta_{lk} \Big/ p_q = \sum_{l=0}^{N_q} \alpha_{il} \sum_{k\in K_q} p_k \theta_{lk} \Big/ p_q = \sum_{l=0}^{N_q} \alpha_{il}\bar{\theta}_{lq}, \tag{21}$$

where $\theta_{0k} = 1$ **and** a_{i0} is the constant term for (stage-2) row i. Hence from (21) we have:

$$\frac{\partial \bar{b}_{iq}}{\partial \bar{\theta}_{lq}} = a_{il} = \frac{\partial b_i}{\partial \theta_l} = constant \quad \forall q \in Q,\ i = 1,\dots,m_2,\ l = 1,\dots,N_q. \tag{22}$$

And thus for linear $b(\theta)$ with constant coefficients the second term from (15) is constant. Combining (15) and (22) gives the formula for computing the sensitivities to the aggregate solution required in (11) as follows:

$$\frac{\partial z_{LB}}{\partial \bar{\theta}_{lq}} = \sum_{i=1}^{m_2} \frac{\partial z_{LB}}{\partial \bar{b}_{iq}} \frac{\partial b_i}{\partial \theta_l} = \sum_{i=1}^{m_2} \frac{\partial z_{LB}}{\partial \bar{b}_{iq}} \alpha_{il}. \tag{23}$$

<u>Case 2: Stochastic costs, fixed RHS</u>

For case-2 problems the aggregate sub-problems are concave and thus give an upper bound $T_Q = z_{UB} \geq z^*$ (see "Part I" for discussion). While case-2 problems can be reformulated and solved as case-1 problems, we examine the cost sensitivities in order to develop the complete sensitivity analysis which will be used for the general problem discussed below. Similar in spirit to the case-1 analysis, we seek the change in partitioning which induces the maximum decrease in z_{UB}. We approximate the change in the upper bound z_{UB} relative to a change in the stochastic parameter vector around a solution point of (PA) as:

$$\Delta z_{UB} \approx \frac{\partial z_{UB}}{\partial \bar{\theta}} \Delta\bar{\theta} = \sum_{q\in Q} \sum_{l=1}^{N_q} \frac{\partial z_{UB}}{\partial \bar{\theta}_{lq}} \Delta\bar{\theta}_{lq}. \tag{24}$$

The development parallels that for case 1 above. For the general case $c = c(\theta)$ and the cost mean is defined as:

$$\bar{c}_{jq} = \sum_{k\in K_q} p_k c_{jk}(\theta_k) \Big/ p_q, \tag{25}$$

Expanding terms and applying the chain rule to (3a) gives:

$$\frac{\partial z_{UB}}{\partial \bar{\theta}_{kq}} = \text{Si\^{}ifH\^{}} \qquad (26)$$

taking $\bar{c}_f = /(\bar{0}_f)$ and $x_f = g\{\bar{6},)$.

The sensitivity of the upper bound to die cost mean $(<h_{UM}/d?j)$ is obtained from the aggregate problem (PA) objective function definition (3a), giving:

$$\frac{\partial z_{UB}}{\partial \bar{c}_{jq}} = p_q x_q. \qquad (27)$$

The stage-2 activity vector jc_f = JC^ is obtained from the solution to (PA). And when $c(8)$ is linear witfi constant coefficients we have:

$$c_y = yj0, \qquad (28)$$

where[5] C€9T\ *0eft"'**,and* $y_€3l^{(A'«^{+I}>K"}$». Expandingtennsgives:

$$c_j = \sum_{1-0}^{N_\theta} \gamma_{jl}\theta_l \quad y/si,...,\wedge, \qquad (29)$$

and hence $$\frac{\partial \bar{c}_{jq}}{\partial \bar{\theta}_{kq}} = \frac{\partial c_j}{\partial \theta_l} = \gamma_{jl} = \frac{\partial c_{jk}}{\partial \theta_{lk}} = constant. \qquad (30)$$

The sensitivity of the upper bound T_Q to the aggregate solution x_f is provided by the dual solution of (PA). The sensitivity of x_f to $\bar{0}_f$ is not readily available, but can be estimated. For the purposes of the case-3 example problems presented in § 7 the authors used a differencing method to compute the partials $dx_k fdl)_k$. Combining expressions gives the formula for computing the sensitivities to the aggregate solution required in (24) as follows:

$$\frac{\partial z_{UB}}{\partial \bar{\theta}_{kq}} = \sum_{j=1}^{n_2} p_q x_{jq}\gamma_{jl} + \sum_{j=1}^{n_2} \frac{\partial z_{UB}}{\partial x_{jq}} \frac{\partial x_{jq}}{\partial \bar{\theta}_{kq}}. \qquad (31)$$

[5] The stochastic vector 0 is augmented by 1 to account for the constant term, such that 0_{0k} • 1 and p_{i0} is the constant for $c^\wedge$ $VkeK$.

Case 3: Stochastic costs, stochastic RHS

When both costs and RHS are stochastic the problem is convex in the constraints and concave in the objective function. The sensitivities to changes in the stochastic parameters **are** obtained by combining the results of cases 1 **and** 2 above. However, the aggregate problem no longer has the property of simple convexity or concavity. Thus the bounding properties (as for cases 1 and 2) no longer apply and the repartitioning strategy must change accordingly. The sensitivity for projecting changes in the aggregate problem based on changes in the stochastic parameter is similar in principle to cases 1 and 2 above, which when combined give:

4?-§45 = ££-§-45,, **(32)**

and $$\frac{\partial \bar{z}}{\partial \bar{\theta}_{lq}} = \sum_{i=1}^{m} \frac{\partial \bar{z}}{\partial \bar{b}_{iq}} \alpha_{il} + \sum v^* r_j ££ \frac{\partial x_{jq}}{\partial \bar{\theta}_{lq}}, \qquad (33)$$

assuming *b(0)* and *c(6)* are stochastic linear functions with constant coefficients defined by (18) and (28) above.

Projecting min **and** max *A0:*

We now consider repartitioning the (stage-2) event space *K* in order to improve the approximation of the exact solution *i* • provided by the aggregate solution V_Q. Each of the three cases defined above has unique bounding properties, namely: (0 V_Q = i_u < *t* *z*• for case-1, *iix) IQ* = z_m £ z* for case-2, and (iII) T_Q does not bound z* for case-3 problems. While the repartitioning strategy necessarily differs for each case, the essential goal of finding the min or max change in *0* via a repartitioning of the event space is common to all three cases.

To explore the potential changes in the partition means, we consider deviations from the means resulting from splitting partitions along single dimensions, thus retaining the rectangular structure of the partitioned even space. For each partition *qeQ* axis / defined by the parameter 0, has mean *0*^. Deviations from the mean for each partition *qeQ* and event *k e* K_q are defined as follows:

$\Delta\bar{\theta}_{lk}$ = 0»-6|t:*cAv (34)

The cumulative positive and negative changes from the means are defined, respectively, as:

$$\Delta\bar{\theta}_{lq}^{+} = \sum_{k\in K_q^+} p_k\left(\theta_{lk} - \bar{\theta}_{lq}\right) \Big/ \sum_{k\in K_q^+} p_k = \sum_{k\in K_q^+} p_k\theta_{lk} \Big/ \sum_{k\in K_q^+} p_k - \bar{\theta}_{lq} \tag{35}$$

and

$$\Delta\bar{\theta}_{lq}^{-} = \sum_{k\in K_q^-} p_k\left(\theta_{lk} - \bar{\theta}_{lq}\right) \Big/ \sum p_k = \sum p_k\theta_{lk} \,/\, \sum p_k - \bar{\theta}_{lq}, \tag{36}$$

where

$$K_q^+ \subseteq K_q : \theta_{lk} \geq \bar{\theta}_{lq} \tag{37}$$

and

$$K_q^- = K_q \setminus K_q^+. \tag{38}$$

The authors note that alternative criterion to (37) exist The choice of (37) can be viewed as a weighted-average approach, as opposed to the 1^ norm, for example.

Projecting min **and** max *Āz:*

By combining the results from the above analysis we can project the rcpartitioning which is projected to produce the desired maximal change in the aggregate solution V_Q, per (11), (24), or (32) for casc-1, -2 or -3 problems, respectively. The corresponding sensitivities are computed from (23), (31), or (33), respectively. The change in the partition means is independent of problem case, and is determined from (35) and (36), using criterion (37) to delineate events into positive and negative influence subsets, corresponding to the projected partition split given any pair *(14).*

For the most general case, the projected positive and negative changes in I for any partition $q \in Q$ and stochastic parameter index $l = 1,...,N_\theta$ are defined, respectively, as:

$$\Delta\bar{z}_{lq}^{+} = \max_{\Delta\bar{\theta}_{lq}^{\pm}}\left\{\left(\sum_{i=1}^{m_1}\frac{\partial\bar{z}}{\partial\bar{b}_{iq}}\alpha_{il} + \sum_{j=1}^{n_2} p_q x_{jq}\gamma_{jl} + \sum_{j=1}^{n_2}\frac{\partial\bar{z}}{\partial x_{jq}}\frac{\partial x_{jq}}{\partial\bar{\theta}_{lq}}\right)\Delta\bar{\theta}_{lq}^{\pm}\right\} \tag{39}$$

and

$$\Delta\bar{z}_{lq}^{-} = \min_{\Delta\bar{\theta}_{lq}^{\pm}}\left\{\left(\sum_{i=1}^{m_1}\frac{\partial\bar{z}}{\partial\bar{b}_{iq}}\alpha_{il} + \sum_{j=1}^{n_2} p_q x_{jq}\gamma_{jl} + \sum_{j=1}^{n_2}\frac{\partial\bar{z}}{\partial x_{jq}}\frac{\partial x_{jq}}{\partial\bar{\theta}_{lq}}\right)\Delta\bar{\theta}_{lq}^{\pm}\right\}, \tag{40}$$

where *b(0)* and c(0) are stochastic linear functions defined by (18) and (28) above. Let the maximum and minimum taken over all indices $l = 1,...,N_\theta$ and $q \in Q$ *be* defined as:

$$\Delta\bar{z}^{*+} = \max_{l,q}\left\{\Delta\bar{z}_{lq}^{+}\right\} \tag{41}$$

and

$$\Delta\bar{z}^{*-} = \min_{l,q}\left\{\Delta\bar{z}_{lq}^{-}\right\}. \tag{42}$$

Let *AT* denote the optimal choice from (41) or (42) for each of the three problem cases. When (stage-2) costs are fixed and RHS's stochastic (i.e., case-1 problems), the aggregate problems are convex in the constraints. As shown in Tart I", successive disaggrtgation of (PA) provides a monotonically increasing lower bound z_u on z^*. The algorithm correspondingly selects $AT = AS^{**}$ in order to maximize the increase in the lower bound z_u^z^. Case-2 problems are reformulated and solved as case-1 problems (see Tart I" for discussion). For the most general case with both costs and RHS's stochastic (i.e., case-3), the aggregate problem is convex in the constraints and concave in the objective function. Consequently, the aggregate problem solution provides neither a valid lower or upper bound on z\ Furthermore, it can be shown that successive disaggregation does not necessarily provide a monotonic decrease in the difference $|z^*_Q - z^*|$. The algorithm seeks the disaggregation steps projected to maximize changes in the aggregate solution T_Q. This approach is based on the assumption that maximal changes in T_Q will correspond to different bases for the true solution, and that grouping (via successive disaggregation) these bases into probability-weighted subspaces according to their impact on I^*_Q will give an efficient approximation to the full-space problem (PO).

4 Successive Disaggregation Algorithm

With the above definitions and preliminaries aside, we now consider the question of how to solve (PO) using successively refined aggregate problems (PA). The objective is to successively disaggregate the problem in as few steps as possible until the aggregate solution T_Q approaches z^* within some predetermined tolerance.

Let the (stage-2) discrete probability space *(&>T,P)* have a rectangular support defined by the intersection of independent stochastic parameter spaces. If we consider only single axis cuts for splitting any partition into two sub-sections, then the rectangular structure is retained through all subsequent repartitioning. Algorithm SRO is designed to successively improve the estimate T_Q of z* through repeated solution of increasingly disaggregated LFs approximating the full-space *CET* problem (PO). For cases 1 and 2 which are convex and concave, respectively, each subsequent solution to the partition approximation LFs provides a monotonic improvement in the estimate T_Q of z^*. For all three cases solution to the aggregate problem (PA) provides updated sensitivity information which is applied to project the repartitioning which is predicted to minimize the difference $|z^*_Q - z^*|$. The general form of the successive disaggregation algorithm SRO is now presented. A high-level flowchart of the algorithm is provided as Figure 4.

Algorithm SRQ

Step 1 **Initialization.**

Define problem parameters and discrete probability space. Initialize $\partial b_i/\partial\theta_t = \alpha_{it}$, $\partial c_j/\partial e$, = Yjir G « - { i } . ' (^ 4 - 0 .

Step 2 **Aggregate LP solution.**

Given *Q* and the set of feasibility "cuts", formulate and solve the aggregate LP (PA) => £, %9 df/db^ ,and x^.

Step 3 **Full-space (PI) feasibility sub-problem.**

Solve the MILP(P4) using x , = j ^ as fixed input =^ u\ f}\ duals, and slacks.

Step 4 **Full-space (PI) feasibility test**

If $u^* > 0$, then (PI) is infeasible for given x_l. A feasibility cut is added to aggregate model (PA). The term /T indicates the most violating constraints with corresponding dual A. The feasibility cut ^ [- / T + ^ j e J - p f t / ^ SO (see § 2 for discussion) augments the (PA) constraints for all subsequent solutions. Return to *step 2* with *Q* unchanged.

Otherwise u^* £ 0, and (PO) is feasible using x_x = I*.

5rep 5 **Stage-2 solution decision (optional).**

If the stage-2 solution is required, go to *step* 9.

Otherwise, the stage-2 solution is not required. Continue on to *step 6.*

Step 6 **Project changes in $\bar{z}$ from partition splits.**

Compute $\bar{0}_V$ $A\bar{d}$^ $A\bar{6}$^ from (16), (35), and (36), respectively, V/ = 1,...,N,; *qeQ*. Compute *Az*^ and *AS*^ from (39) and (40) V/ = l,...,Af#; *qeQ*. Compute *Az£* and *AS£* from (41) and (42). Select *AT* according to problem type (see § 3 discussion) and determine the corresponding (/,?) pair representing the partition *q* to be split along axis /.

Step 7 **Aggregate model convergence test**

If $A\bar{z}^*$ £ to/j,. t then no further (significant) improvement is projected for dis-aggregating. Stage-1 approximation solution is given by *z s *I* and $x^*_x = \bar{x}\{$.**

Otherwise, the projected maximal change $A\bar{z}^* > tol$^, and disaggrcgation as determined in *step 6* proceeds to refine the aggregate solution.

Step8 Repartition the probability space.

Let q_t denote the partition to be split, and q_f the partition formed by the split. Using (/,?) pair corresponding to $A\bar{z}^*$ from *step 6,* split partition *q,* along the selected axis / at the axis mean according to (34)-(36). Update q_t and q_f partition mean terms. Augment the partition sec Q«- $Qyjq_r$ *Go to step 2.*

Step 9 Stagc-2 (PI) sub-problem (optional)

Solve full-space (PI) using $x_x = \bar{x}$^ as fixed input => Zj, *x*^* and duals.

Step 10 Full-space optimality test (optional).

Solve stage-1 MEP sub-problem (P8) =^ z, = u\ $z_{UB} = z_x + z_2$, Xj, and duals.

Step 11 Full-space convergence test (case-1 or case-2 problems)[6] (optional).

Compute the gap $(\Delta z = z_{UB} - \bar{z})$ between the upper and lower bounds. If gap > tolerance, return to *step 6;* otherwise stop with the approximate solution given by $z^* \cong \bar{z}^*$, $x_1^* \cong \bar{x}_1^*$, and $x_{2k}^* \cong x_{2k}$.

The algorithm can be modified by substituting solution to (Rl) in place of (PI) in *step 3,* and changing the upper bound computation in *step 10* by replacing $z_{UB} = z_x + z_2$ with $z'_{UB} = z_x + z'_2$. In the authors* implementation of algorithm SRO feasibility is tested "initially" (i.e., $Q = 1$) and "finally" (i.e., $Q = Q^*)_9$ skipping the feasibility test (i.e., *steps 3-4-5)* intermediately for initially feasible problems. Our computational results indicate that all example problems remained feasible once an initial feasible solution was found. Furthermore, all problems we tested[7] had converged to the exact solution x_1^* when the $A\bar{z}^*$ £ *tol*^. condition was true.

---

[6] When either RHS terms (case-1) or cost coefficients (case-2) arr stochastic, but not both (case-3), the aggregate problem can be solved to provide a lower boond. Solution to (PI) and (PS) provides an upper bound. The objective bound gap test is applicable for case-1 and case-2 problems, but not for case-3 problems. For case-3 problems the aggregate convergence test can be used to terminate the algorithm, unless another means of finding the lower bound is employed.

[7] Only problems under 35k rows by 35k columns were solved exactly, due to limitations in the LP code implementation and computer memory.

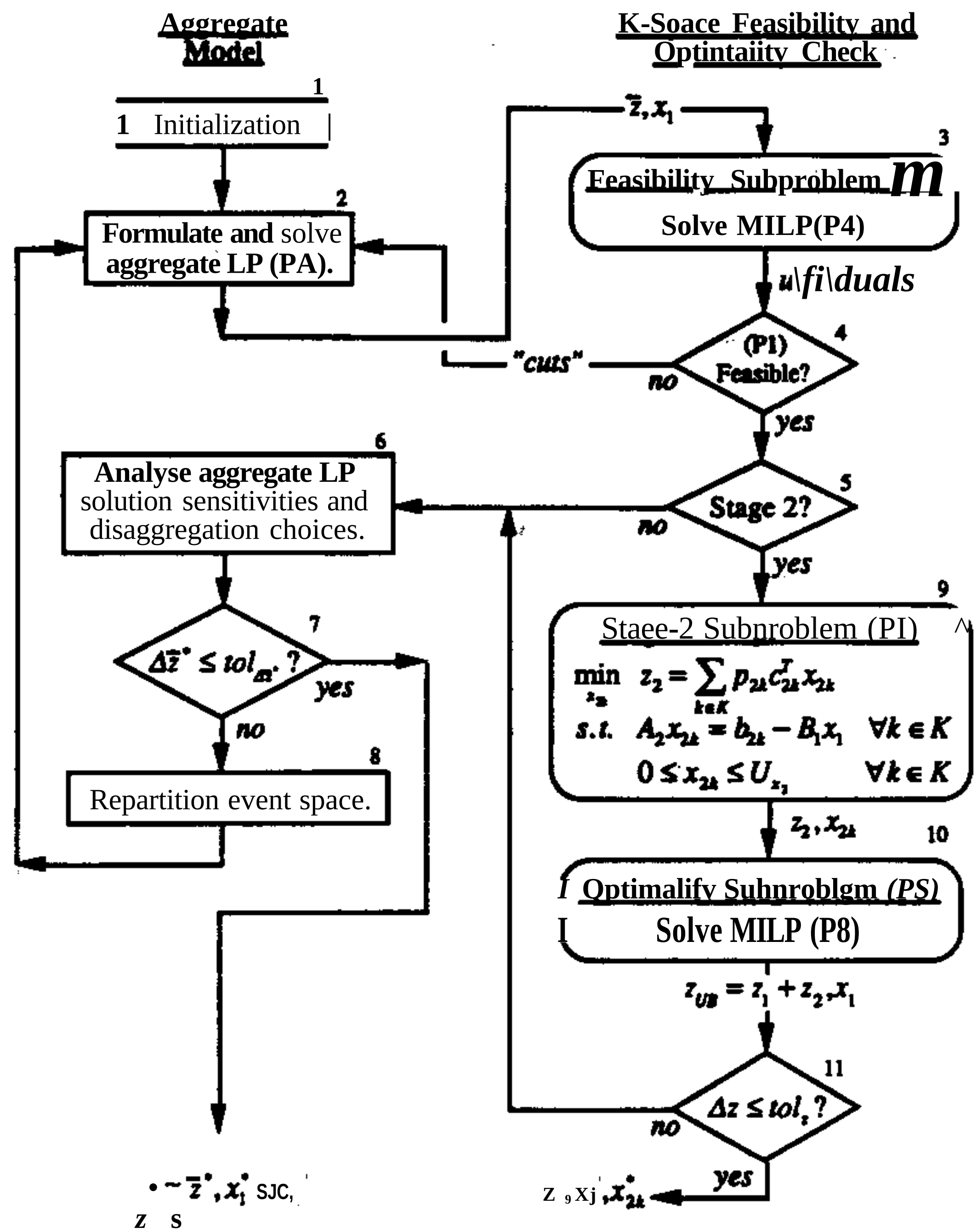

Figure 4. Overview flowchart of the two-stage successive disaggregation algorithm (SRO), based on problems (PO) and (PA). Algorithm *"steps*" are indicated by the numbers outside of the boxes.

5 Illustrative Example

Example problem (EX2P) is a simplified planning LP of the form (PO) which we use to demonstrate the basic procedure for algorithm (SRO). We omit the slacks and retain the inequalities to give the following certainty equivalent formulation.

(EX2P) min z = a^4· $\sum_{UK} p_{2k} x_{2k}$

s.t. jc1+2x2ik£du $\forall k \in K$

$2x_1 + x_{2k} \geq d_{2k}$ $\forall k \in K$

$x_1 \geq 0$

$x_{2k} \geq 0$ $\forall k \in K,$

where dl(0«{U,3}, p^(i) = {0.2,0.5,0.3},

<*,(/) = {1,2,3}, *PdtU)* = {0.2,0.6,0.2},

$p_{2k} = p_{d_1}(i) \cdot p_{d_2}(j),$

$x_1 \in \Re^1,$

$x_{2k} \in \Re^1$ VJfceAT,

$i \in I,\ j \in J,\ K = I \times J.$

Here JC24 are continuous stage-2 variables expanded out over the discrete probability space given by crossing demand 1 and 2 (independent) probability spaces, indexed by ï € / and *j* € /, respectively. The first stage variable is x,. The cost coefficients are 1.0 for stage-1 and stage-2 variables. Solving (EX2P) as formulated above gives the exact solution of z* = 1.6333 and JÇ* = 0.6667, feasible for all nine possible events.

In order to illustrate algorithm (SR0), we now outline the solution method on (EX2P). The aggregate sub-problems (PA) as defined in § 2 are given by the specification of the probability space and the set of disjoint partitions *Q* conforming to (14). With the set *Q* initially a single partition, the aggregate solution is V_Q = z^ = 1.3667 and $\bar{x}_1^*$ = 0.6333, feasible for all events *ke K.* Analyzing the sensitivities (summarized in Table 2) and projecting partition splits according to § 3 indicates the best projected repartitioning is given by *0>qY* = (2,1) and *A*z̄* = 0.3333, where *At* = *AI*** is the predicted maximal increase in the lower bound. We note that for (EX2P) there is a 1-to-l mapping of $\bar{0}$^ to $\bar{b}$^ such that:

$$\partial\bar{\theta}_{iq}/\partial\bar{b}_{iq} = \alpha_{il} = 1,\ \partial\bar{z}/\partial\bar{b}_{iq} = \partial\bar{z}/\partial\bar{\theta}_{iq} = \lambda_{iq}\ \ \forall i = l$$

and $$\partial\bar{\theta}_{iq}/\partial\bar{b}_{iq} = \alpha_{il} = 0\ \ \forall i \neq l.$$

Splitting the single partition along probability dimension *1 = 2,* updating the partition means, reformulating and solving (PA) yields the improved solution T_Q = 1.5333 and $\bar{x}$[= 0.4666. Repeating the cycle gives a continuing reduction in the predicted increase in the lower bound and a monotonic increase in the computed lower bound. The complete results are summarized in Table 3 and Figure 5.

Table 2. (EX2P) initial cycle sensitivity analysis summary ($N_q = q = 1$).

$i = l$	λ_{iq}	A6£	A6^	K	4z;
1	0.333	0.9	-0386	0.300	-0.129
2	0.333	1.0	-0.250	0.333	-0.083

Table 3. **(EX2P) results summary.**

N_q	my	$\Delta \bar{z}^*_{predicted}$	$\Delta \bar{z}^*_{observed}$	$\bar{z}^*_Q$	z_{UB}
1	2,1	0.333	**0.167**	1.3667	1.6476
2	1.1	0.300	**0.026**	1.5333	1.6476
3	1.1	0.061	**0.048**	1.5591	1.6476
4	2.1	0.033	0.023	1.6067	1.6366
5	2,4	0.003	0.003	1.6300	1.6366
6	n/a	0	0	1.6333	1.6333

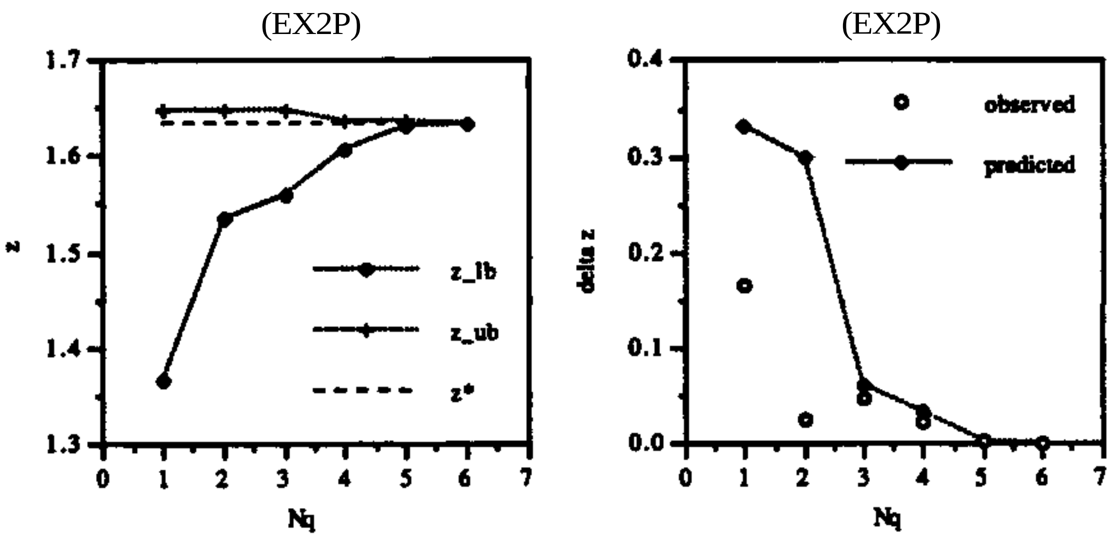

Figure 5a/b. (a) Monotonic improvement in (PA) approximation to the (P0) objective as aggregate probability space is successively disaggregated (Nq « # partitions). Exact solution is reached when 6 partitions are used, (b) In all cases the predicted maximal change in the aggregate objective (based on sensitivity and repartilioning analysis) is greater or equal to that observed. Furthermore, the predicted change monotonically decreases, and goes to zero when the exact solution has been reached.

The upper bounds generated from solving (P0) using fixed $\bar{x}_i^*$ from the (PA) solutions are reported in the last column of Table 3. These bounds were generated for completeness, and were not used in the original solution to the problem. That is, the exact optimal solution was found by successively disaggregated solutions to (PA), terminating when no further improvements in the lower bound were projected (i.e., after forming 6 partitions). The monotonic increase in the lower bound with successive disaggregation is shown in Figure 5a. The exact solution ($\hat{z} \cdot V_Q = z_u$ and $x^f = X$) is reached when 6 (of 9) partitions are used. As shown in Figure 5b, the predicted maximal change in the aggregate objective (based on sensitivity and repartitioning analysis) is greater or equal to that observed at every step. Furthermore, the predicted change monotonically decreases with disaggregation, and goes to zero when the exact solution has been reached. Similar behavior has been observed for all case-1 (and case-2) example problems, although the fraction of expansion necessary to get the exact solution tends to decrease significantly as the cardinality of the probability space increases, as might be expected intuitively. The disaggregation sequence is shown in Figure 6.

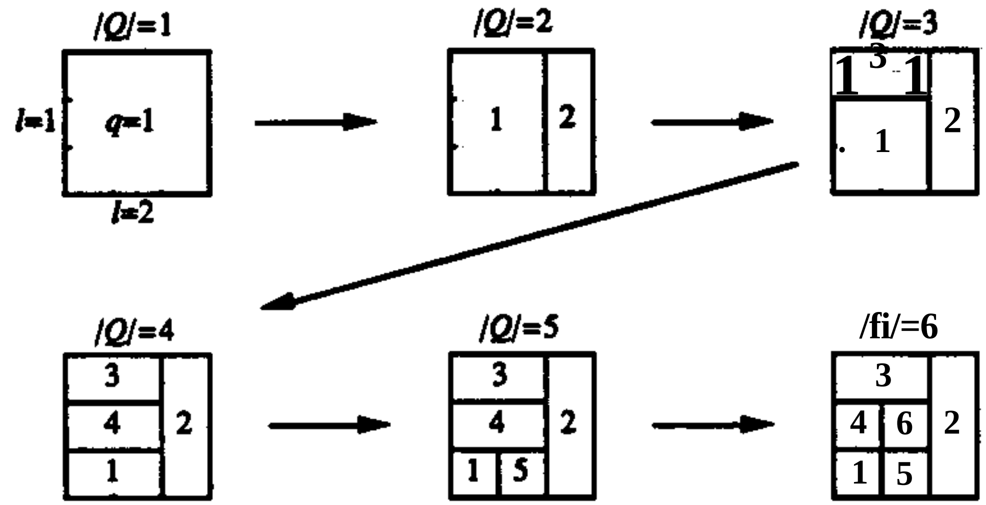

Figure 6. Partitioning sequence for solution to (EX2P) via algorithm (SRO).

6 Planning Example

The second example problem (EX11) is more representative of industrial planning models, and is a variation of the refinery planning example in Edgar and Himmelblau (1989, p. 254). A simplified representation of the refinery production network is shown in Figure 7. We compare the performance of the partitioning algorithm with other methods (namely, Benders decomposition, certainty equivalent transformation, and single-point mean value approximation). The problem considers the purchasing, processing, storage, and sales for a single refinery over two time periods. Five crude oils are available for purchase subject to the supply limitations. The crude oils can be stored over time subject to the tank inventory limits. The production model is a linear conversion of feeds to products according to a fixed yield matrix, with limits on the production capacities. Four products are manufactured and available for sale according to market demands, each of which is considered random with four states possible for each demand Thus there are 256 possible events depending on the state of each demand. Products can also be sent to tankage for temporary storage. A restriction is applied fencing a net balance of inventory in both feed and product tanks over the time horizon, with accumulation allowed in intermediate stages. Hence, all materials purchased must ultimately be processed and then sold as products. The objective function is cost minimization which is formulated as the sum of crude purchasing and operating costs minus the revenue from product sales. A 0.99 discount factor is applied to all stage-2 costs and revenues. Stage-2 demands axe on average lower than those of the first stage.

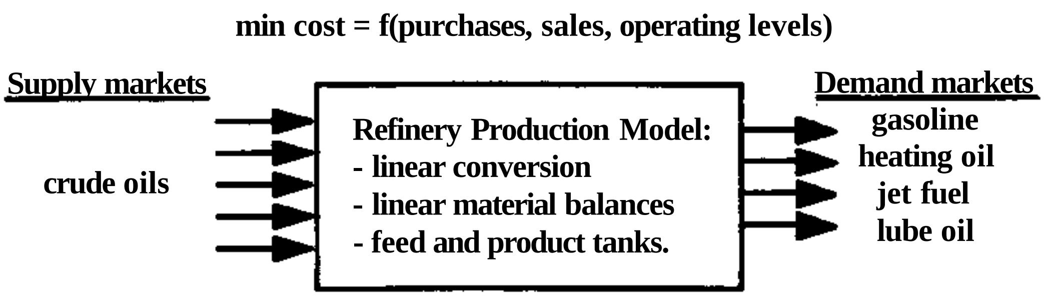

Figure 7. Refinery production model for example problem (EX11).

The basic dimensions of the LP arc as follows. For the deterministic case (i.e., no probabilistic expansion) there are 13 stage-1 constraints, 22 stage-2 constraints, and 1 constraint defining the objective function, giving a total of 36 constraints. With the probabilistic expansion over all 256 events, the stage-2 constraints are reproduced once for each event leading to 5632 stage-2 constraints, and a total of 5646 constraints. In the deterministic case there are 55 variables, and 141 non-zero elements in the LP. In the probabilistic case each stage-2 variable is expanded over the event space, leading to a total of 6940 variables and 20286 non-zero elements in the LP.

We now consider the solution via the partitioning algorithm and compare the results with the solution via alternate methods. The results of this problem solution are summarized in Tables 4 and 5 below. The terms CET and SRO refer to certainty equivalent transformation and partitioning algorithm, respectively. For the Benders decomposition results we report the major iterations. A variation on algorithm (SRO) was used, whereby the partition splits were not restricted to one split per partition. Thus the algorithm progressed directly from N_f « 1 -» 4 -» 8 -> 16 partitions for (PA). In this manner fewer intermediate solutions were required. Each repartitioning is recorded as an "iteration" for algorithm (SRO).

Note in Table 4 that with die SRO method, 16 partitions arc required to converge to the exact optimum. The CET version was solved with OSL on an IBM/6000 using simplex and interior point methods. In both cases the problem was initialized with ail activities set to zero, except the inventory terms which were set at their starting values (Le., t*0). As shown in Table 5 die simplex method took nearly 3000 iterations and about 2 minutes. Solution using the interior point method took 21 iterations and about 30 minutes. Solution by Benders decomposition took between 11 and 14 major iterations, depending on the level of precision required Timing data for Benders are not reported since oor implementation of die Benders method does not take advantage of die decoupled stage-2 LP sub-problems. Each major iteration via Benders requires the solution of N_K LP sub-problems. Solution to these sub-problems can be prohibitive, as previously discussed. By comparison, using a sensitivity-based successive disaggregation approach, the partitioning algorithm found the exact solution to machine tolerance in 4 iterations (i.e., 4 different parritionings). The maximum expansion of the variables and constraints for the SRO algorithm is in this case Af_e = 16, or 0.0625 of that required by GET or Benders methods. This reduction in variable and constraint expansion is critical in making the problem more tractable.

Table 4. Partition algorithm solution to refinery planning example problem (EX11).

Method	# partitions	z_u	£	Zj»	gap(%)
CET	#, = 256	n/a	-6522.8	n/a	n/a
SRO	1	-6536.1	-6536.1	-6468.6	1.03
SRO	4	-6529.5	-6529.5	-6517.7	0.18
SRO	8	-6527.0	-6527.0	-6521.8	0.08
SRO	16	-6525.8	-6525.8	-6525.8	0

Table 5. Alternative algorithm solution to refinery planning example problem (EX11).

Method	Comment	CPU (sec)	Iterations	gap (%)
CET	OSL/Simplex	121	2831	n/a
CET	OSL/Interior point	1872	21	n/a
Benders	Complicating: x,	n/a	11	0.1
Benders	Complicating: x,	n/a	13	0.01
Benders	Complicating: x,	n/a	14	0.001

7 Computational Results and Analysis

We now present the results of several computational experiments conducted to examine the behavior of algorithm SRO on two-stage planning problems. Problem (EX11) is a variant of die Edgar and Himmelblau (1989) as described in die previous section. The number of events for each independent probability dimensions (i.e., stochastic demands) is parameterized so as to change the total number of events in the joint probability space. Increasing the total number of events, IATI, results in a higher resolution definition of die probability space. As shown in Table 6 and Figure 8 the ratio of expansicm for algorithm SRO as compared to the GET solution is small, and decreases as 1X1 increases. For the largest problem instance (exl Ij) less than 300 partitions are needed to find the epsilon-exact solution, versus 160,000 for the GET equivalent problem. Correspondingly, the maximum sized SRO LP is 7,450 rows by 7,750 columns versus 4,160,000 rows by 4,320,000 columns for the GET equivalent problem.

In Table 6b the solution times reported correspond to solution of the problem on an alpha workstation with 64 MB RAM memory (athena.nectar.cs.cmu.edu). It should be noted that our implementation of algorithm SRO is relatively unsophisticated from a performance standpoint. Reading and writing files to disk consumes substantial time, and could be greatly reduced by using memory reads and writes. Also, no storting basis is used for each new LP for successively disaggregated problems (PA), resulting in a "cold" start to each sub-problem. Modifying the previous bases for each subsequent solution would no doubt greatly improve the efficiency of the implementation. Another obvious extension to the implementation would be to include a Benders-type solution to (PA) when the LP sub-problems get large.

Table 6a. Partition algorithm solution to refinery planning example problem (EX11).

Problem	Case	101	10,1	\K\	N_l	$N'_Q/\|K\|$
exile	1	4	4	256	16	0.06250
exile	1	4	6	1296	36	0.02777
exl If	1	4	8	4096	56	0.01367
exllg	1	4	10	10000	90	0.00900
cxllh	1	4	12	20736	103	0.00497
exlli	1	4	16	65536	208	0.00317
exllj	1	4	20	160000	286	0.00179

Table 6b. Expansion and timing for solution to planning example problem (EX11).

Problem	(PA) max LP size (rows x cols)	CETLPsize (rows x cols)	SROtimc (elar/sed sec)
exile	430x460	6670x6940	34
exile	950x1000	33710x35020	179
exllf	1470x1540	107kx111k	474
cxllg	2354x2458	260kx270k	1632
exllh	2692x2809	539kx560k	2093
cxlli	5422x5644	1704kx 1770k	16286
exllj	7450x7750	4160k x 4320k	49756

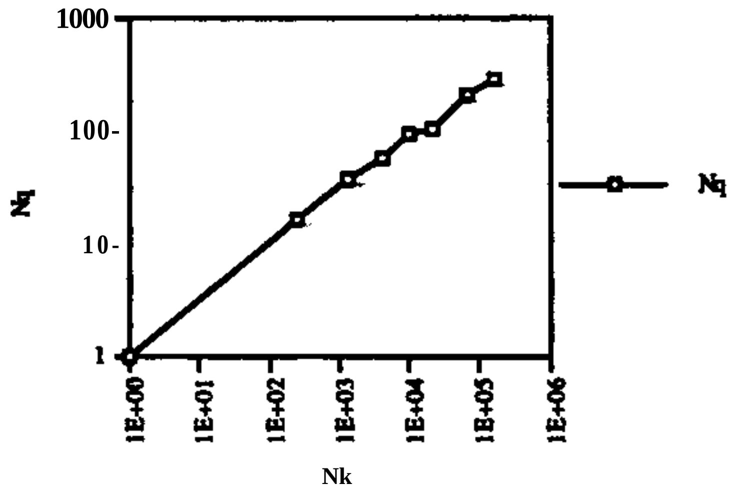

Figure 8. The number of partitions (required to get the exact[8] optimal solution) *NQ* versus the number of events IAII for problem EX11 with 4 stochastic demands. Increases in \K\ represent higher resolution of the discrete probability functions. Increasing resolution in the full space requires fractionally fewer partitionings to reach the exact solution.

The second example problem (EX30) is taken from the two-stage planning model of Ierapetritou and Pistikopoulos (1994). Two instances of the model are solved, each with stochastic RHS's (demands) and costs (sales costs) making (EX30) a case-3 problem. There are four events for each independent probability dimensions (i.e., stochastic demands or costs). In both instances, two stochastic costs were included. Problem instance 'a' has three stochastic demands, and 'b' has two. The fully aggregated problem has 13 stage-1 constraints, 33 stage-2 constraints, 26 stage-1 variables, and 39 stage-2

[8] Problems with \K\ > 3000 were not solved to with the GET formulation due to their size. The SR0 algorithm terminated for all cases when the predicted change in the objective was below the tolerance which wassetat10"*.

variables. The results for the problem solutions via the SRO algorithm are summarized in Table 7. Figures 9 and 10 show the progression of the solution through disaggregation for problem instances V and fb\ respectively.

In both instances, the SRO algorithm selected the partitioning sequence according to the maximal absolute predicted (positive or negative) change in the objective. The sensitivities $dx/d\bar{d}$ were computed numerically by perturbing the (PA) sub-problems. As might be expected, calculating die numerical derivatives dominated the computing resource requirements (ca. 98%). Both as a subset of the numerical derivative calculations and as a component of the SRO algorithm, the primary computing resource users were the solution to the LP sub-problems via Minos 5.1 (ca. 2/3 of the total) and the expansion and writing of the MPS files (ca. 1/3 of the total). Clearly, there is considerable room for improvements in the implementation, including parallelization. Our primary motivation was to test the method, as opposed to the computing implementation per se.

As shown in Table 7 - and as was observed for case-1 problem (EX11) - the ratio of expansion for algorithm SRO as compared to the GET solution is small, and decreases as IKII increases. As shown in Figures 9 and 10, the aggregate problems (PA) produced solution values below the upper bound values (generated by substituting the stage-1 solution to (PA) into the CET problem (P0)). While in general for case-3 problems the aggregate model solution does not provide a lower bound, in this instance the aggregate models behave similarly to that of a case-1 problem and terminate at the exact solution when the aggregate solution is equal to the upper bound.

Table 7a. Partition algorithm solution to Imperial planning example problem (EX30).

Problem	Case	101	10,1	\K\	N_l	K/\K\
ex30a	3	5	4	1024	22	0.02148
ex30b	3	4	4	256	13	0.05078

Table 7b. Expansion and timing for solution to example problem (EX30).

Problem	(PA) max LP size (rows x cols}	CET LP size (rows x cols)	SRO time (elapsed sec)
ex30a	740x885	33806 x 39963	4417[9]
ex30b	443 x 534	8462x10011	990

[9] Time reported was divided by 2, since machine was loaded with another job using 1/2 the CPU during this run.

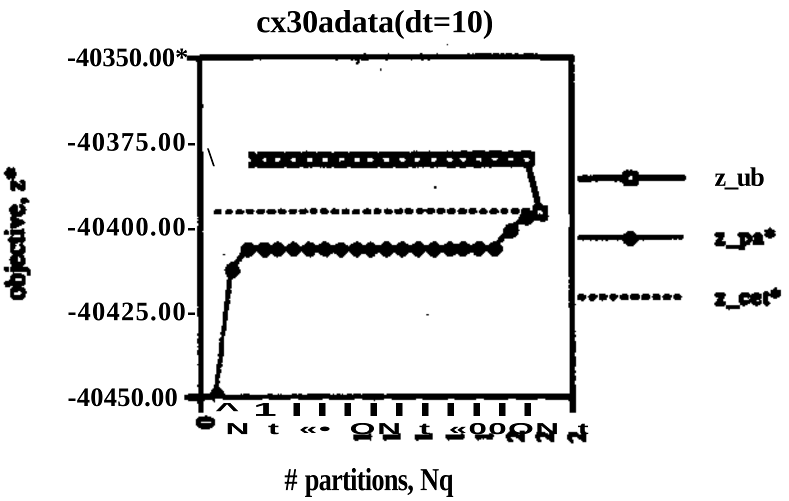

Figure 9. Imperial planning example (ex30a) with 3 stochastic demands and 2 stochastic costs, each with 4 independent events.

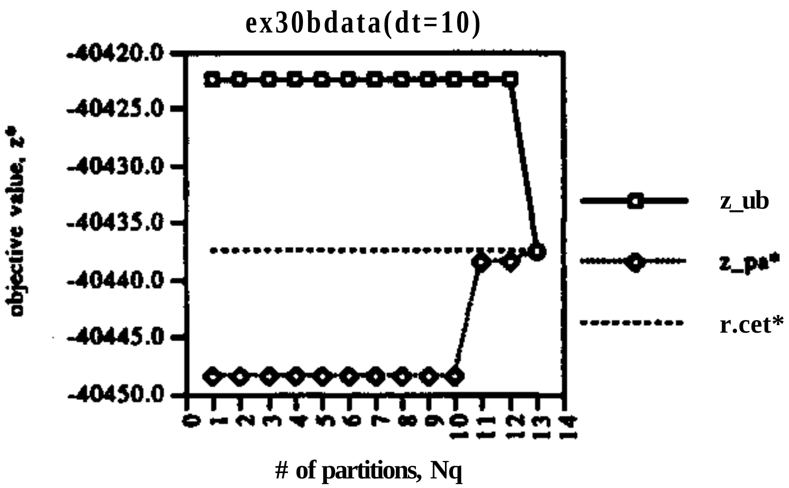

Figure 10. Imperial planning example (ex30b) with 2 stochastic demands and 2 stochastic costs, each with 4 independent events.

While Acre are numerous obvious improvements in the computing implementation, the qualitative results from the examples run by the authors indicates that the successive disaggrcgation algorithm may be a useful addition to the tool set for solving both two-stage and multi-stage stochastic linear programming problems.

8 Conclusions

Extending linear planning models to include probabilistic representation of parameter uncertainties is required for a complete solution to the problem. The resultant stochastic programming problems can be intractable due to the exponential growth in the problem size (variables and constraints). We have proposed a successive disaggregation algorithm which refines the solution to a desired tolerance level, reapplying the mean value approximations over partitions. Early results on two-stage examples indicate that the algorithm finds the optimal solution in a much reduced expansion, as compared to the CET formulation, and also requires considerably less work than Benders schemes. Work is underway to further investigate extensions to multi-stage stochastic linear programs.

Acknowledgments

This research was sponsored (in part) by the National Science Foundation and the Defense Advanced Research Projects Agency under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives. We are also grateful to PSC for their helpful assistance and for the computing resources which they provided. Ignacio Grossmann was partially supported by SERC in the UK for a Visiting Fellowship at Imperial College. Special thanks are also extended to Prof. Gregory J. McRae of M.IT. for his continued support and valued input throughout the effort

References

Benders, J.F. (1962), Partitioning Procedures for Solving Mixed-Variable Programming Problems, *Numerische Mathematic* 4, pp. 238-252.

Birge, J.R. (1982), Decomposition and Partitioning Methods for Multi-Stage Stochastic Linear Programs, Technical Report No. 82-6, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor.

Birge, J.R. (1985), Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs, *Operations Research,* Vol.33, No.5, Sept-Oct. 1985.

Birge, J.R. (1992), Stochastic Programming: Optimizing the Uncertain, Technical Report No. 92-24, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor.

Birge, J.R. and S.W. Wallace (1986), Refining Bounds for Stochastic Linear Programs with Linearly Transformed Independent Random Variables, *Operations Research Utters,* Vol.5, No.2, July 1986.

Birge, J.R. and S.W. Wallace (1987), A Separable Piecewise Linear Upper Bound for Stochastic linear Programs, Navel Postgraduate School, Technical Report NLS55-87-001, February 1987.

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

Clay, RX. and IJL Grossmann (1994), Optimization of Stochastic Planning Models I. Concepts and Theory, to be submitted for publication.

Dantzig, G.B. (1955), Linear Programming Under Uncertainty, Management Science, VoL 1, No. 3 A 4, pp. 197-206.

______ (1963), *Linear Programming and Extensions** Princeton, N.J.: Princeton University Press.

______ (1987), Planning Under Uncertainty Using Parallel Computing, Technical Report SOL 87-1, Department of Operations Research, Stanford University, Stanford, California.

Dantzig, G.B. and P.W. Glynn (1989), Parallel Processors for Planning Under Uncertainty, Technical Report SOL 88-8R, Department of Operations Research, Stanford University, Stanford, California.

Dempster, M.A.H. (1965), *On Stochastic Programming,* Ph.D. Thesis, Dept. of Mathematics, Carnegie Mellon University.

______ (1980), Introduction to Stochastic Programming, in *Stochastic Programming,* M.A.H. Dempster (Ed.), New York: Academic Press, pp. 3-62.

Edgar, T.F. and D.M. Himmelblau (1988), *Optimization of Chemical Processes,* New York: McGraw Hill.

Eimolicv, Y. and RJ-B Wets (1988), *Numerical Techniques for Stochastic Optimization,* New York: Springer-Verlag.

Gassmann, H.I. (1990), MSLiP: A Computer Code for the Multistage Stochastic Linear Programming Problem, *Mathematical Programming,* 47, North-Holland, pp. 407-423.

Grossmann, I.E. and C.A. Floudas (1987), Active Constraint Strategy for Flexibility Analysis in Chemical Processes, *Computers and Chemical Engineering,* Vol. 11, No. 6, pp. 675-693.

Grossmann, I.E. and D.A. Straub (1991), Recent Developments in the Evaluation and Optimization of Flexible Chemical Processes, Proceedings Computer-Oriented Process Engineering (eds. L. Puigjaner and A. Espuna), pp. 49-59, Elsevier.

Ierapetritou, M.G. and E.N. Pistikopoulos (1994), Simultaneous Incorporation of Flexibility and Economic Risk in Operational Planning Under Uncertainty, *Computers and Chemical Engineering,* Vol. 18, No. 3, pp. 163-189.

Infanger, G. (1991), Monte Carlo (Importance) Sampling within a Benders Decomposition Algorithm for Stochastic Linear Programs, Extended Version: Including Results of Large-Scale Problems, Technical Report SOL 91-6, Dept of Operations Research, Stanford University, Stanford, California.

Louveaux, RV. (1986), Multistage Stochastic Programs With Block-Separable Recourse, Mathematical Programming Study 28, North-Holland, pp. 48-62.

Madansky, A. (1960), Inequalities for Stochastic Linear Programming Problems, *Management Science,* 6, pp. 197-204.

Swaney, R.E. and LE. Grossmann (198S), An Index of Operational Flexibility in Chemical Process Design: I. Fonnulation and Theory, *AIChE Journal,* Vol. 31, No. 4, pp. 621-630.

Wets, RJ.-B. (1966), Programming Under Uncertainty: The Equivatent Convex Program, *SIAMJ.Appl. Math.,* 14, pp. 89-105.

_______(1983), Stochastic Programming: Solution Techniques and Approximation Schemes, *Mathematical Programming: The State of the Art (Bonn 1982),* A. Bachem et al, Eds., Berlin: Springer-Verlag, pp. 566-603.

_______ (1989), Stochastic Programming, *Handbooks in Operations Research and Management Science,* Vol. 1, G.L. Nemhauser et al, Eds., North-Holland, pp. 573-629.