PARALLEL PROCESSORS FOR PLANNING UNDER UNCERTAINTY

George B. DANTZIG and Peter W. GLYNN
Department of Operations Research, Stanford University, Stanford, CA 94305, USA

Abstract

Our goal is to demonstrate for an important class of multistage stochastic models that three techniques — namely nested decomposition, Monte Carlo importance sampling, and parallel computing — can be effectively combined to solve this fundamental problem of large-scale linear programming.

1. Hedging against uncertainty

An unresolved problem of great importance is that of finding an "optimal" solution to a linear program whose parameters (the coefficients and constant terms) are uncertain but whose distributions are known. The problem may be viewed as one of finding an optimal allocation of scarce resources and hence is fundamental to Operations Research, Control Theory, and Economics. If it could be solved in general, it would significantly advance man’s ability to plan, schedule and control complex situations.

Industry and government routinely solve deterministic mathematical programs for planning and scheduling purposes, some involving thousands of variables with a linear or nonlinear objective and many thousands of inequality constraints. The solutions obtained are often ignored, however, because they do not properly hedge against future contingencies. It is relatively easy to reformulate models to include uncertainty. Because the resulting size of system in practical applications can be enormous, the bottleneck has been (and still is) our capability to solve them.

Over the years, since the time it was first proposed independently by Dantzig and Beale in 1955 [14,2], there has been progress in stochastic programming but on a disappointingly limited scale. However, with the advances in the past ten years in

© J.C. Baltzer AG, Scientific Publishing Company
computer technology, in techniques for solving large-scale models in general, and in simulation methods (particularly advanced sampling techniques), the research of specialists in the field of stochastic programming has appreciably accelerated. Workers in the field include such names in the United States as Birge, King, Nazereth, Rockafellar, and Wets; in Canada as Dempster, Gassmann, and Ziemba; in Brazil as Pereira; in Hungary as Prekopa; in Switzerland as Frauendorfer and Kall; at IIASA, Russians such as Ermoliev and Gaivoronski. Specific references to their work will be given later.

Many important optimization problems involve complex systems that repeat a matrix pattern over time with variations in coefficients due to random events and the introduction of new technologies. For example, models of dynamic activities are often formulated as linear programs: the time-dependent nature of the system implies that the constraint matrices display a nearly block-triangular (staircase) structure. Staircase linear programs occur in many large-scale applications, in particular the modeling of multi-period economic or energy-related processes. Engineering applications such as the design of an exhaust nozzle of a jet engine can also be posed in this form. Thus, staircase linear programs have a matrix structure that arises in many problem contexts and therefore efficient methods for finding their solution is of great importance.

Although the simplex method is an extremely efficient algorithm for general linear programs, the number of iterations required to solve problems with staircase structure has been observed to be inordinately high. In addition, each iteration is more expensive than average because the basis factorizations tend to be more dense. On the other hand, interior methods have been observed to be more efficient than a general purpose simplex code when applied to staircase models. Even so, as the size of the system to be solved grows because the formulator increases the number of time steps or makes his model stochastic (or both), it becomes necessary to use solution methods based on decomposition and then employ one of several techniques that have been proposed for solving stochastic programs.

We favor coupling a nested dual-decomposition approach with Monte Carlo importance sampling and the assignment of the sampling tasks to parallel processors. In this paper, we outline the role that parallel processors and importance sampling can play when combined with earlier results on decomposition.

2. Approaches to solving deterministic dynamic systems

Lower block-triangular matrix structures are typical for planning problems over time because activities initiated in period \( t \) have input and output coefficients in periods \( t, t+1, \ldots \). For example, for \( T = 3 \) such a matrix has the form:
By the introduction of in-process inventories and other devices, linear programs of lower block-triangular type are mathematically equivalent to staircase or multi-stage problems of the form:

\[
\begin{bmatrix}
  A_{11} \\
  A_{21} & A_{22} \\
  A_{31} & A_{32} & A_{33}
\end{bmatrix}
\]  

Find Min \( Z \) and vectors \( X_t \geq 0 \), such that

\[
\begin{align*}
  b_1 &= A_1 X_1 \\
  b_2 &= -B_1 X_1 + A_2 X_2 \\
  &\vdots \\
  b_t &= -B_{t-1} X_{t-1} + A_t X_t \\
  &\vdots \\
  b_T &= -B_{T-1} X_{T-1} + A_T X_T
\end{align*}
\]

(Min) \( Z = c_1 X_1 + c_2 X_2 + \ldots + c_{t-1} X_{t-1} + c_t X_t + \ldots + c_{T-1} X_{T-1} + c_T X_T \)

where the matrices \( A_t, B_t \) and vectors \( b_t, c_t \) are given.

The decomposition approach to specially-structured linear programs was suggested for the primal by Dantzig and Wolfe [23] and for the dual by Benders [4]; the first application of a nested form of decomposition is given in Dantzig [15], pp. 466—469. In applying dual decomposition to staircase dynamic systems, the original linear program is decomposed into a set of smaller linear programs, one for each time period. Each period communicates with the adjacent periods by "sending" its optimal solution to the following period, and its optimal price vector to the preceding period, where it is used to generate necessary conditions called Benders "cuts".

Various explanations have been advanced as to why standard techniques fail to perform efficiently on staircase problems. Because the dynamics of the system are known, techniques that take into account its special structure offer promise of greatly increasing computational efficiency. Research exploiting this approach has been encouraging. Among the methods for solving such systems are those of Dantzig and Wolfe [23], Glassey [42], and Ho and Manne [48], who suggested a primal nested decomposition approach that has been applied to large-scale modeling problems in the European Common Market by Ho and Loute [47], Bisschop and Meeraus [10b], Dantzig and Perold [22], Fourer [34—36], and Nishiya [64]. For a collection of papers relating to this subject, see Dantzig, Dempster and Kallio [18].
The dual form of nested decomposition for staircase systems (which is one we use in our current work) has intuitive appeal because of its resemblance to certain "greedy" heuristics used by practical planners. The chief difference is the decomposition technique converges to an optimal solution. An important early work is that of Van Slyke and Wets [76]. Our more recent research along these lines began with Birge [5] and Dantzig [16], followed by major studies by Philip Abrahamson [1], Robert Wittrock [80], and Dan Scott [72] for the nonlinear case. These studies are concerned with enhancing the process of communication in the dual form of nested decomposition. Rather than sending the next period a single solution, each communicates a family of optimal solutions to the following period (and similarly for the information communicated to the preceding period).

Motivated by the prospect of greatly improved efficiency offered by the results of Wittrock and Abrahamson, Robert Entriken has been continuing their investigations of dual nested decomposition methods, particularly as they relate to vector processors and parallel processors. His results form part of his Ph.D. thesis. Earlier (experimental) computer implementations were only suitable for small test problems. More sophisticated software has been under development by Entriken, using modules of (Murtagh and Saunders [60]) the MINOS nonlinear programming system. Features include sparse LU factorization of the local bases, efficient ways to update the data structures that preserve the sparsity, techniques for increasing the numerical stability, and special staircase matrix partition generators. Experimentation on a number of large staircase test problems has been encouraging (Entriken [29]).

3. Stochastic case: Mathematical structure

For a wide class of stochastic planning problems, the values of $b_t$, $B_{t-1}$, $A_t$, $c_t$ for $t > 1$ of model (2), described above, are not known to the planner with certainty at time 1 but would become known to him at some later time $\tau \leq t$. The value $\tau$ itself could be a random variable and there could be a different $\tau$ for every element of the matrices and vectors. While the values of these matrices may not be known, their probability distributions may be given. In such problems, the planner wants to make a decision $X_1$; let random events happen; make a decision in period $t = 2$; let random events happen; make a decision in period $t = 3$, etc. He may wish to make the choice $X_1$ so that the expected value $E(Z)$ is minimum. For an extensive review of research on stochastic programming, see Birge and Wets [10a]. First, we give reasons why this very general class of stochastic problems is likely to remain intractable in the foreseeable future, with or without the availability of parallel processors. We then describe our plan to concentrate on an important class of practical stochastic problems, which are currently not practical to solve on serial-processor mainframes but which we believe could become so on multi-processor mainframes and parallel processors using our proposed techniques.
We begin with a review of the simplest two-stage case, first studied in Dantzig [14] and [15], and subsequently developed by R. Wets [75,77] and many others.

\[ b_1 = A_1 X_1, \quad (X_1, X_2) \geq 0 \]
\[ b_2 = -B_1 X_1 + A_2 X_2 \]

(Min) \[ Z = c_1 X_1 + c_2 X_2 \] (3)

where the first stage data \((b_1, A_1, c_1)\) are known with certainty and \((b_2, c_2, B_1, A_2)\) can take on possibly a continuum of values \(b_2(\omega), c_2(\omega), B_1(\omega), A_2(\omega)\) with probability (density) distribution \(p(\omega)\) for \(\omega\) in \(\Omega\), or a discrete probability distribution \(p(\omega)\) for \(\omega = 1, 2, \ldots, K\). The range of \(\omega\) in \(\Omega\) may therefore be continuous or it may be discrete, finite or infinite.

For (3), if the parameter \(\omega\) takes on \(K\) distinct values, the problem of minimizing expected costs under uncertainty is tractable for \(K\) small. For example, if \(K = 3\), the stochastic problem is equivalent to the deterministic linear program:

Find Min \(Z\) and vectors \(X_1 \geq 0, X_2(\omega)\) for \(\omega = 1, 2, 3\), such that

\[ b_1 = A_1 X_1 \] (4)
\[ b_2(1) = -B_1(1) X_1 + A_2(1) X_2(1) \]
\[ b_2(2) = -B_1(2) X_1 + A_2(2) X_2(2) \]
\[ b_2(3) = -B_1(3) X_1 + A_2(3) X_2(3) \] (4.1)

\[ Z = c_1 X_1 + p_2(1)c_2(1)X_2(1) + p_2(2)c_2(2)X_2(2) + p_2(3)c_2(3)X_2(3). \] (4.2)

To simplify the discussion, we may assume (without loss of generality in the bounded case) that \(c_2(\omega) \geq 0\). Typically, this problem is solved using Benders [4] decomposition. The key idea in Van Slyke and Wets [76] is to replace the second-period constraints (4.1) and their objectives by a set of inequalities expressed only in terms of \(X_1\) and a scalar variable \(\theta_2\), which are necessary conditions for feasible and optimal solutions to (4). \(\theta_2\) represents the second-period contribution to the objective \(Z\). These necessary conditions (5.1), called "cuts", are added sequentially \((l = 1, 2, \ldots)\) to the first-period problem until they become sufficient to solve (4). Cuts come in two "flavors": feasibility cuts and optimality cuts. The Master problem for Benders' decomposition has the form:
MASTER PROBLEM: \[ \text{Find Min} \, Z, \, X_1 \geq 0, \, \theta_2, \] \[ b_1 = A_1 X_1 \quad X_1 \geq 0 \] \[ g_l^I \leq G_l^I X_1 + \delta_l^I \theta_2 \quad l = 1, \ldots, L \] \[ (\text{Min}) \, Z = c_1 X_1 + \theta_2, \]

where \( \delta_l^I = 0 \) in the case of feasibility cuts and \( \delta_l^I = 1 \) for optimality cuts. Assuming \( B_1 \) is the same for all \( \omega \), on any major iteration no more than \( L \leq m_2 \) of the constraints (5.1) will be tight, the rest may be dropped (possibly to be regenerated at a later iteration).

The solution \( X_1 = X_1^* \) obtained is then "tested" for feasibility and optimality by solving the second period with \( X_1 = X_1^* \). This is called the subproblem or "SUB" for short. The SUB decomposes into \( K \) independent sub-subproblems:

SUBPROBLEM: For \( \omega \) in \( \Omega \), find \( \text{Min} \, Z_2(\omega), \) \[ X_2(\omega) \geq 0, \] \[ A_2(\omega) X_2(\omega) = b_2(\omega) + B_1(X_1^*), \] \[ c_2(\omega) X_2(\omega) = Z_2(\omega) (\text{Min}), \] \[ (6.1) \]

where \( \Omega = \{ \omega | \omega = (1, \ldots, K) \} \). These sub-subproblems are solved for \( \omega = 1, \ldots, K \).

If feasible, their expected value of \( \text{Min} \, Z_2(\omega) \) is computed to test global optimality. If the test fails, the expected values

\[ g_l^{I+1} = \sum_{\omega} p_2(\omega) \pi_2(\omega) b_2(\omega), \quad G_l^{I+1} = - \sum_{\omega} p_2(\omega) \pi_2(\omega) B_2(\omega) \] \[ (6.2) \]

are used to augment the cut conditions (5.1). If infeasible, then infeasibility prices are used in an analogous way. In the research of Wittrock, several extreme cuts are generated; in that of Abrahamson, in place of \( B_1(X_1^*) \) several columns \( B_1(X_1) \) for \( X_1 = X_1^*, X_1^{**}, \ldots \) are passed to the next period at once and convex combinations are allowed; these can greatly speed up convergence.

4. Research restricted to a relevant class of stochastic programs

The two-stage case (4) represents the simplest application of parallel computers to stochastic programming. The general multi-stage case, using the corresponding "reduction" to the equivalent deterministic linear program almost always, even for \( T = 3 \), becomes intractable due to the proliferation of possible outcomes. This is illustrated in diagram (7).

Each node in the tree represents a point in time where a decision can be made. One of several contingencies can then happen, which are represented by the
arcs branching out of the node. Corresponding to this, in the first stage we have only one set of constraints $A$, the same as in (4). In the second stage, we could have $K$ more sets of constraints, shown as $B, C, D$ above, but in general there could be infinitely many more. If each of these in turn has $K$ outcomes, then in the third stage we have $K \times K$ more sets of constraints, shown as $(E F G H I J L M N)$. It is now easy to see why for $K = \infty$ or $\omega$ continuous, that the proliferation of cases is exponential in the number of stages and is out of hand.

Even if the number of parallel processors available is large, it does not seem to be a practical way to solve the general class of such problems. Because these appear to be too general a class, we have been studying an important subclass of multi-stage uncertainty problems. Typical of such applications is the facilities expansion planning problem under uncertainty. It has a much more tractable structure.

A case in point is the fleet planning done by airlines. The problem is to plan over time the number of aircraft of various types to have in the fleet, together with the expansion of repair and passenger facilities at various airports. Within each time period (say one year), the fleet composition can be changed by buying new aircraft and selling obsolete ones, etc. Another good example is the building of power stations and transmission of lines for electric utilities. For either example, the capacities in any period are used to meet an uncertain demand in period $t$ as shown as random events $E^1(1), \ldots, E^t(\omega_j)$ in diagram (8).

In many such practical applications, it is required to lay out a schedule here and now of the amounts of future capacities (rather than wait and see what the future will look like and then decide). The here and now model, due to Madansky [58], assumes that installed facilities (capacities) will not be affected by any particular event that happens within a period, i.e. by a particular value of the uncertain demand or by a particular repair status of some facility. Only the expected revenues and expected failures (in some sense) to meet demands in the period affect the decision to invest in new facilities or to get rid of older ones.
We illustrate the above concepts on an electric-power application that we use for test purposes. Corresponding to each scenario, \( \omega \) is a vector listing the amount of capacities of transmission lines and generators that are not down for repair at various locations in the system. Typically, we assume that the values of these components are independent random variables. This basically means that we are asserting that generation/transmission failures in one part of the system happen independently of what occurs elsewhere. Equipment down for repair is repaired within a period and therefore does not affect the status of equipment available in the next period. Although this independence assumption simplifies much of the following discussion, the basic ideas and methods that we will be describing are equally pertinent to power systems in which generation/transmission failure is modeled in a dependent fashion.

For this class of problems, the mathematical structure for the facility expansion part of the model remains the same as (2), namely:

\[
b_t = -B_{t-1} X_{t-1} + A_t X_t, \quad \text{for } t = 1, \ldots, T \quad \text{and} \quad B_0 X_0 = 0, \tag{9}\]

where the \( X_t \) are the planned facilities (to be determined) for period \( t \) at a cost \( Z = \Sigma c_t X_t \), where \( b_t, A_t, B_t, c_t \) are all known with certainty. These form the "deterministic part" of (8). Relations (9) plus generated cuts form the dual Master problem.
The Benders subproblem breaks down into sub-subproblems, one for each period $t$ and each $(\omega_t, \bar{\omega}_t)$ combination. The subproblem for period $t$ is defined for some $X_t = X_t^*$ by first finding the vector of facilities $F_t$ that the system owns at time $t$ and $Y_t$, the amount of $F_t X_t$ in repair and available for operations. Therefore,

$$Y_t = \text{Diag}(\omega_t) \cdot F_t X_t, \quad \omega_t \in \Omega_t,$$

where $\text{Diag}(\omega_t)$ is a diagonal matrix with diagonal corresponding to $\omega_t$, a random vector measuring the proportions of the components of $F_t X_t$ not down for repair. The probability distribution of $\omega_t$ is assumed known. Letting $f_t(\bar{\omega}_t)$ be an uncertain demand, we then find optimal dual multipliers and $\text{Min} \ Z_t$ that solve (11) below for each random choice of right-hand side, i.e. for $\omega_t$ in $\Omega_t$ and $\bar{\omega}_t$ in $\bar{\Omega}_t$:

$$DU_t = Y_t^*, \quad U_t \geq 0, \quad Y_t^* = \text{Diag}(\omega_t) \cdot F X_t^*$$

$$HU_t = f_t(\bar{\omega}_t), \quad \bar{\omega}_t \in \bar{\Omega}_t$$

$$\bar{c}_t U_t = Z_t (\text{Min}),$$

where, given the randomly generated capacity vector $Y_t^* = \text{Diag}(\omega_t) \cdot F X_t^*$, demand $f_t(\bar{\omega}_t)$, and fixed matrices $D, H$, the vector of activities $U_t$ that minimizes $Z_t$ is found. The event $(\omega_t, \bar{\omega}_t)$ occurring with probability $p_t(\omega_t) \cdot \bar{p}_t(\bar{\omega}_t)$, is assumed known. These samples are used to estimate cuts of the form

$$g_t^l \leq G_t^l X_t + \delta_{t+1}^l \theta_{t+1}^l ,$$

where $g_t^l$ and $G_t^l$ are generated by formulae corresponding to (6.2). The cost form

$$Z = \sum_t (c_t X_t + \theta_{t+1}^l),$$

together with (9) and (12), form the MASTER PROBLEM.

5. **Role of parallel processors**

Thus, the method of solution we are working on for solving (8) uses parallel processors at each stage. These receive as inputs $X_t^*$ from the Master Problem and give back to the main program cuts (12) that are used to augment (9). If the cardinalities of $\omega_t, \bar{\omega}_t$ are small, this can be done exactly for each $\omega_t, \bar{\omega}_t$ combination. If these are large, then we propose that their expected values be estimated by sampling (discussion of which is the subject of the next section).
Note that the main Master Problem has the form of a deterministic staircase system, which we solve using the dual nested decomposition approach described in detail in the Ph.D. theses of Abrahamson [1] and Wittrock [80], and as implemented by Entriken [29] for the deterministic case (and currently being modified to handle the stochastic case). The advantage of the dual decomposition approach is that the same parallel processors at each stage can be used effectively to provide information in the form of $X_t^*$ for stage $t + 1$ and to pass back cuts from time stage $t$ to stage $t - 1$; the latter cuts are in addition to those given by (12) for $t - 1$, which were generated by the random processes (10) and (11). The discussion which follows is for the $T = 2$ stage case.

Envision one computer at the MASTER level sequentially receiving as inputs cuts (5.1) and solving (5), (5.1) generating as an output $X_1 = X_1^*$. This process provides a lower bound for $\text{Min } Z$ that keeps increasing.

Envision several parallel computers at the SUB level, each having as input the latest value of $X_1^*$ and solving (6) in dual form for all possible choices of $\omega$. When $c_2, A_2$ are the same for all $\omega$, the dual of (6) is an LP with only the dual objective $b_2(\omega)$ changing; to provide cuts for the Master, the processors are used to determine the expected values of dual prices $\pi_2(\omega)$ and $Z_2(\omega)$ to obtain very good approximations. If it is practical, solve (6) for all $\omega$, then via (6.2) and (5.1), solving (5) provides solutions to (6) which generate a valid cut and a correct upper bound estimate for $\text{Min } Z$. The difference between the lower bound and upper bound estimates is then used to test optimality of $X_1^*$ for the original problem. The iterative process is terminated when the difference is deemed small enough.

6. Review of methods for obtaining approximate upper and lower bounds

An extensive review of methods of approximation, together with a list of references, can be found in Birge and Wets [10a]. See also Chapter 1 of Ermoliev and Wets, eds. [33]. For the cases where $K$ is large, infinite, or $\omega$ is continuous, it is in general no longer possible to solve (6) for all $\omega$. Recent research has therefore concentrated on techniques for obtaining a solution which is within an acceptable upper and lower bound of the true optimum. The essential difficulty is a numerical one of approximating the cost term: $E(Z_2) = \sum_\omega p_2(\omega)c_2(\omega)X_2^*(\omega)$, where $X_2^*(\omega)$ for $\omega = 1, 2, \ldots$ are the optimal solutions of the subproblem (6).

Typically, the random event $\omega$ is represented by points in a $d$ dimensional sample space, with the components of $\omega$ representing the random amounts of various facility capacities available for the operating system. It is therefore convenient to view $\omega$ as a $d$ dimensional vector varying over some range of values rather than ranging over an index set:

$$\omega = (\omega_1, \omega_2, \ldots, \omega_d).$$ (14)
The \( j \)th sample point selected from the sample space will be denoted by \( \omega = \omega^j \) and its \( i \)th component by \( \omega_i^j \). It is also convenient to let \( p(\omega) = p_2(\omega) \), \( c(\omega) = c_2(\omega) X_2^* \), and \( \alpha = \mathbb{E}(Z_2) \). Accordingly, we are interested in approximating the sum:

\[
\alpha = \sum_{\omega} c(\omega) p(\omega). \tag{15}
\]

If the \( d \) components of the vector \( \omega \) vary continuously over their range of possible values in \( d \) dimensional sample space, these sums take the form of a multiple integral \( \int \int \ldots \), or if they vary over their range of discrete values, they take the form of multiple sums \( \sum \sum \ldots \).

One technique proposed by others for doing the approximations provides upper and lower bounds for the two-stage continuous case by discretizing the sample space \( \Omega \) into cells and summing the function values at representative points within the cells over all cells. Some references are Huang et al. [49], Kall and Stoyan [52], Birge [6], Birge and Wets [9], and Frauendorfer and Kall [37]. These authors obtain lower bounds by applying Jensen's inequality. Upper bounds require an exponential number of function evaluations with respect to dimension \( p \) of the sample space. Birge and Wets [10a] recently proposed an alternative scheme for obtaining upper bounds that requires solving "only" \( O(m_2) \) linear programs. It replaces partitioning into cells by a method that seeks out an approximation using a small number of "positive" basis representations that span the space of columns associated with the second stage.

Another technique for doing the approximations which is closer in spirit to the one we follow is to sample from \( \Omega \) randomly and to use sample information to guide the optimization algorithm. These include the stochastic quasi-gradient methods of Ermoliev [32] and Gaivoronski [38]. They provide asymptotic convergence in the continuous case as the size of the sample \( \rightarrow +\infty \). They do not provide any practical way to compute bounds.

7. Our sampling procedure and termination rules

Historical experience of numerical analysis, nevertheless, suggests that Monte Carlo (random) sampling is the most efficient way to calculate the multiple integrals (or summations). In other words, random sampling is the method of choice for integrating functions in higher dimensional space (Davis and Rabinowitz [24], Deák [25]). Indeed, for higher dimensional space it may be the only way because, generally speaking, the amount of computational effort (function evaluations) is relatively insensitive to dimension size; this is to be contrasted with the schemes based on partitioning the space into cells and summing over values associated with representative points in the cells which grows exponentially with \( d \), the dimension of the coordinate space within which the sample points \( \omega \) in \( \Omega \) are imbedded.
The sum we need to approximate in the two-stage case is (15). To do this, an independent random sample $S$ of size $s$ is drawn and the appropriately weighted mean is used for the upper bound estimate. By the law of large numbers, the distribution of such means, even for moderate size $s$, is approximately normally distributed about the true mean with $\sigma^2 = (1/s)$ (population variance). If this estimate of error turns out to be too high, the sample size is increased until the error of the estimate becomes tolerable.

When an acceptable sample size has been attained, an approximate cut based on the sample is generated. This cut would be a true cut if the variable $\theta$ is replaced by $\theta(S)$ representing the subproblem objective when the universe of possible $\omega$ values and their relative probabilities are restricted to those of the sample $S$. Thus, the cut can be viewed as having all its coefficients correct except for the constant term $g_1^*$. We believe the error committed by the substitution of $\theta$ for $\theta(S)$, even for small size samples, is distributed about the same as standard error of the approximation of $E(Z_2)$ based on a sample of size $s$.

Based on an idea of Gerd Infanger, who has been collaborating with us, a distribution of lower bound estimates can be generated by randomly varying the $g_i^*$ terms according to their approximately normal distributions and solving the master problem for each such selection. The mean of the thus generated lower bounds can then be used as the lower bound estimate. Such a mean is also approximately normally distributed, and its standard error can be estimated. Using standard statistical tests (appropriate variants of Student's $t$ test), the upper and lower bounds can be tested to see if they differ significantly from a specified amount. If they test significantly different, another round of Benders' decomposition is initiated. If not, the iterative process is terminated with $X_1^*$ declared optimal. Should it turn out that the substitution of $\theta$ for $\theta(S)$ above can significantly bias the lower bound estimates, it will be necessary to find a way to estimate and adjust for the bias.

8. Variance reduction: Importance sampling

As the central limit theorem described above suggests, the convergence rate of Monte Carlo sampling (although dimension independent) can be quite slow, namely of the order $s^{-1/2}$ in the number of function evaluations, where $s$ is the sample size. This puts a premium on the development and application of variance reduction techniques for computation of $E(Z_2)$. The idea here is to try to reduce the variance. This then reduces the error of the estimates in proportion to the reduction in standard deviation achieved. A particularly promising approach for obtaining significant variance reduction is to apply importance sampling.

In our test application to electric power, we expect the solution to hedge to some degree against contingencies in which significant facilities are disabled because of (random) failure. Loss of capacity due to failure reflects itself in the linear program-
ming formulation primarily through constraints on expected unsupplied demand. Our test model has unsupplied demand constraints (called reliability constraints), which we treat as corresponding to a higher-order D-W or primal master so that the entire problem presented earlier is actually a sub to this primal master. This master’s dual variables are penalties (Lagrange Multipliers) which are used to weight the reliability constraints, which are then subtracted from the objective to form a modified objective for the primal master’s sub. Because of safety margins built into the electric power systems application, significant capacity must be disabled for the unsupplied demand constraints to come into play. Clearly, this event typically has low probability because power systems are designed to be very reliable. Thus, the events that have a major impact on the solution’s ability to hedge are precisely those events which will only rarely be picked up by a “naive” sampling procedure based on the original sampling frequencies. This, in turn, leads to a high variance in the estimator associated with naive sampling.

The idea behind importance sampling is to change the sampling procedure so that those events that matter more are sampled more. In the power systems context, this typically means that we sample contingencies in which multiple failures occur at a higher frequency than they occur naturally. Multiple failures lead to loss of significant system capacity, which in turn forces the solution to hedge properly to avoid such failures. Thus, importance sampling appears intuitively to be a useful tool in the power systems setting.

We now briefly describe the theory that underlies importance sampling. Let \( q(\omega) > 0 \) be any arbitrarily chosen (positive) probability mass function. Later, we say how \( q(\omega) \) is chosen. We now re-express \( \alpha \) in (15):

\[
\alpha = \sum_{\omega} \frac{c(\omega)}{q(\omega)} p(\omega) q(\omega) \tag{16}
\]

\[
\alpha = E_q \frac{c(Y)}{q(Y)} p(Y) \tag{17}
\]

where \( Y = (Y_1, Y_2, \ldots, Y_d) \) is a random vector having a probability mass function \( q(\cdot) \) (that is, \( P(Y = y) = q(y) \)), and \( E_q \) denotes expectation with respect to \( q \). Let \( Y = Y^j \) be the \( j \)th random sample point drawn with replacement. The variance of the above estimator of \( \alpha \) based on a sample of size \( s \) is just:

\[
\text{var}_q \left( \frac{1}{s} \sum_{j=1}^{s} c(y^j) \frac{p(y^j)}{q(y^j)} \right) = \frac{1}{s} \left( E_q \left[ c(Y)^2 \frac{p(Y)^2}{q(Y)^2} \right] - \alpha^2 \right). \tag{18}
\]
There is an obvious optimal choice of the mass function $q$, namely $q(y) = q^*(y)$, where

$$q^*(y) = \frac{c(y)p(y)}{\sum_{\omega} c(\omega)p(\omega)}. \quad (19)$$

We assume $c(y) \geq 0$ to simplify the discussion. In fact, the variance of $c(y)p(y)/q^*(y)$ is easily seen to be zero when $c$ is non-negative. In other words, we obtain a perfect estimate for $E(Z_2)$ in just one observation!

Of course, this result, while true, is too good. The basic difficulty is that generating variates from $q^*(\cdot)$ is not practical because the denominator of $q$ is the actual sum that we are trying to calculate. Nevertheless, the result provides a useful heuristic guide on how to choose a good sampling population $q$. The heuristic that one follows is: Choose a sampling population $q$ having two properties: (a) that $q(\omega)$ be roughly proportional to $c(\omega)p(\omega)$, and (b) that $q$ be a kind of function with which it is easy to carry out calculations and easy to generate variates from $q$ that are computationally inexpensive.

Suppose, for the sake of argument, that the function $c(\omega)$ is roughly multiplicative in its arguments:

$$c(\omega) \approx c_1(\omega_1) \ldots c_d(\omega_d). \quad (20.1)$$

(The rougher this approximation is, the higher will be the variance of the estimator.) We then choose

$$q(\omega) = \left( \frac{c_1(\omega_1)p_1(\omega_1)}{\bar{c}_1} \right) \left( \frac{c_2(\omega_2)p_2(\omega_2)}{\bar{c}_2} \right) \ldots \left( \frac{c_d(\omega_d)p_d(\omega_d)}{\bar{c}_d} \right), \quad (20.2)$$

where $Ec_i(\omega_i) = \bar{c}_i$ is easily estimated by sampling the one-dimensional marginal distribution of $\omega_i$. Sample points $y^j$ for $j = 1, \ldots, s$ are selected by independently choosing components $y^j_i$ according to their marginal distributions $c_i(y_i)p_i(y_i)/\bar{c}_i$. The arithmetic mean of the observed values $c(y^j)p(y^j)/q(y^j)$ for $j = 1, 2, \ldots, s$ is the estimate for $\alpha$.

Although the multiplicative representation is easy to do calculations with, for our cost function application it turns out to be an unreasonable one, even as an approximation. A better heuristic is to assume that the cost function is better approximated by one that is roughly additive in its arguments, i.e.

$$c(\omega) \approx \sum_{i=1}^{d} c_i(\omega^i). \quad (21)$$
In this case, we can again reduce the computation of the joint mass function $q(\omega)$ to something involving the marginal distributions. The $d$-dimensional problem is then basically reduced to $d$ one-dimensional problems. More specifically, if $c$ takes on the additive form, then choose $q(\omega)$ as follows:

\[
q(\omega) = \frac{p(\omega) \sum_{i=1}^{d} c_i(\omega_i)}{\sum_{\omega} p(\omega) \sum_{k=1}^{d} c_k(\omega_k)}
\]

\[
= p(\omega) \sum_{i=1}^{d} \left( \frac{\bar{c}_i}{\sum \bar{c}_k} \right) \left( \frac{c_i(\omega_i)}{\bar{c}_i} \right),
\]

where again $\bar{c}_i = E c_i(\omega_i)$ is easily estimated by sampling the marginal distribution of $\omega_i$. Finally, we write

\[
q(\omega) = \sum_{i=1}^{d} \left( \frac{\bar{c}_i}{\sum \bar{c}_k} \right) \left( \frac{p_i(\omega_i)c_i(\omega_i)}{\bar{c}_i} \right) \cdot \prod_{k \neq i} p_k(\omega_k).
\]

The expression (17) for $\alpha$ becomes

\[
\alpha = \sum_{i=1}^{d} \frac{\bar{c}_i}{\sum \bar{c}_k} E_i \left( \frac{c(Y)p(Y)}{q(Y)} \right),
\]

where $E_i$ means that component $y^i_j$ of sample point $y^j$ is to be independently sampled according to the marginal distribution

\[
q_i(\omega_i) = \frac{p_i(\omega_i)c_i(\omega_i)}{\bar{c}_i}
\]

and all other components $k \neq i$ according to the marginal distribution $p_k(\omega_k)$. These expectations weighted by $\bar{c}_i/\sum \bar{c}_k$ yield $\alpha$ or the estimate of $\alpha$ when each $E_i$ is estimated by sampling.

We now address the question of how to calculate the $c_i(\omega_i)$'s. This can be done by evaluating the cost function on a relatively small lattice of points, namely a set of lattice points along $d$ coordinate directions:

\[
c_i(\omega_i) \approx c(\tau_1, \ldots, \tau_{i-1}, \omega_i, \tau_{i+1}, \ldots, \tau_d) - c(\tau_1, \ldots, \tau_i, \ldots, \tau_d),
\]
where \((\tau_1, \ldots, \tau_I, \ldots, \tau_d)\) is arbitrarily chosen at some fixed set of values. This determines the \(c_i's\) up to the additive constant \(c(\tau_1, \ldots, \tau_d)\). This additive constant can be disposed of by writing the cost function \(c(\omega)\) as \(c(\omega) = c(\tau_1, \ldots, \tau_d) + \Delta C(\omega)\). The function \(\Delta C(\omega)\) is again of additive form, but has the advantage that we know a priori that we may take \(\Delta c_i(\tau_i) = 0\). This eliminates the additive constant from the picture.

The degree of variance reduction obtained by applying this "additive importance sampling method" depends on the extent to which the true cost surface is fit by an additive representation. If the fit is poor, the variance estimate of the mean of the sample will be high. Experimental results based on the additive approach are described by M. Nakayama [61], who applied this method to our large-scale electric power system facilities test problem. It turned out to be a highly effective means of variance reduction. Indeed, the size of sample required to obtain the same size interval with the same degree of confidence of covering the true minimum value was \(1/20,000\) smaller using "importance" sampling than would have been the case using "naive" sampling.

9. Research in progress and future research

We have presented in detail the approach we are taking to solve the facilities expansion problem. Future research concerns (a) the tedious steps of software implementation, (b) testing on large "real-world" models, and (c) ways to improve the efficiency of the algorithms. Some of these tasks are:

**TASK 1**

The present version of the software program, developed by Entriken [29], solves a deterministic staircase linear program by nested dual decomposition. MINOS is used as a subroutine. A version of this code was tested on the parallel computers at Oak Ridge (see Entriken [30]). Dr. Gerd Infanger (from the Technical University, Austria), Marvin Nakayama and Alamuru Krishna are setting up Entriken's code to run on the IBM 3090 multi-vector processor at Stanford. Without further modification, these versions will permit us to solve quite large problems that are staircase or staircase with small amount of uncertainty. Martin Rinhart of the Computer Science Department is currently adapting the 3090 version to run on the Encore parallel processor at Stanford.

**TASK 2**

Introduce into both the IBM 3090 version and into the Encore version subroutines for doing the importance sampling, the generation of cuts, the estimation of the confidence intervals associated with the approximate optimal solution, the
computation of standard errors, sample sizes, and stopping rules. As a preliminary step, Infanger has been running an APL version of our methodology on a Toshiba portable. The procedure has been tested on a "toy" model which he has developed, in which there is an electric power system consisting of three generator demand nodes, three transmission links, and two time periods. We have been able to solve the deterministic equivalent of this problem with 1280 discrete values of the random variables and to compare the exact results with those obtained by importance sampling using a sample size of 20 and the additive first-order approximation (25). The empirical result so far is that very accurate results can be obtained for the toy problem with a sample size as small as 20.

TASK 3

Test the software developed above on our large-scale Electric-Power System Model. This is a realistic multi-area prototype model comprising of six major utilities on the West Coast, covering an area from Canada to Mexico and some Rocky Mountain regions. We have a special version of the GAMS input generator, which inputs row and column indices in a form that can be used to conveniently partition the matrix into staircase and block-angular structures so that we can apply our decomposition code.

TASK 4

Theoretical problems: Future work will focus on ways to cut down the size of sample required to obtain a very close to optimal solution. Some of the topics of this research are: (1) For the higher-order Dantzig-Wolfe Master, can a small sample be used initially to obtain fairly good estimates of what penalty costs should be assigned to the reliability constraint? (2) Under what conditions can variances obtained from earlier Benders cycles be used to improve the standard error estimates of subsequent cycles. (3) We generate the approximate distribution of lower bounds from the Master by a supplementary Monte Carlo procedure which randomizes the RHS terms of the cuts. Can importance sampling be also used here effectively? (4) Is it possible to develop an approach that combines ours with those of Birge, Kall, King, Wets, Ziemba, and others? (5) Generalize the notion of the additive first-order approximation (which currently guides the importance sampling) to a second-order approximation scheme based on second-order marginal distributions. (6) Explore the possibility of combining random sampling with one which always includes certain disaster scenarios, with the view of protecting the system from some of the rare but disastrous contingencies along the lines of Pereira et al. [66,66a]. (8) What is the best way to assign the different sample points to the parallel processors so that only a few iterations will be required to obtain the optimal solution of one sampled scenario, using as a warm start the optimal solution from the prior one.
References

G.B. Dantzig, P.W. Glynn, Parallel processors for planning under uncertainty


[61] M. Nakayama, Section 6 in Dantzig, Glynn, Avriel et al. (1989).