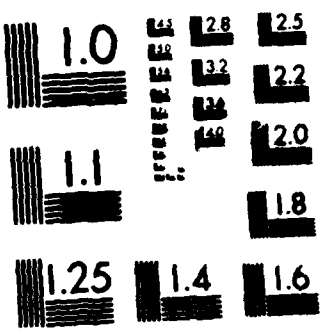
AD-A180 254 PLANNING UNDER UNCERTAINTY USING PARALLEL COMPUTING(U) 1/1
STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB G B DANTZIG
JAN 87 SOL-87-1 N00014-85-K-0343

UNCLASSIFIED F/G 5/1 NL

END
DATE FILMED
8



(4)

Systems
Optimization
Laboratory

PLANNING UNDER UNCERTAINTY USING PARALLEL COMPUTING

by
George B. Dantzig

TECHNICAL REPORT SOL 87-1
Revision of Technical Report SOL 86-11†

January 1987

Department of Operations Research
Stanford University
Stanford, CA 94305

4

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

PLANNING UNDER UNCERTAINTY USING PARALLEL COMPUTING

by
George B. Dantzig

TECHNICAL REPORT SOL 87-1
Revision of Technical Report SOL 86-11†

January 1987

† Technical Report SOL 86-11, "Need to do Planning Under Uncertainty and the Possibility of Using Parallel Processors for this Purpose", April 1986.

Research and reproduction of this report were partially supported by the National Science Foundation Grants DMS-8420623, ECS-8312142, and SES-8518662; U.S. Department of Energy Contract DE-AA03-76SF000326, PA# DE-AS-03-76ER72018; Office of Naval Research Contract N00014-85-K-0343; and Electric Power Research Institute Contract RP 5006-01.

PLANNING UNDER UNCERTAINTY USING PARALLEL COMPUTING [1]

by George B. Dantzig

Abstract

Industry and government routinely solve deterministic mathematical programs for planning and scheduling purposes, some involving thousands of variables with a linear or non-linear objective and inequality constraints. The solutions obtained are often ignored because they don't properly hedge against future contingencies. It is relatively easy to reformulate models to include uncertainty. The bottleneck has been (and is) our capability to solve them. The time is now ripe for finding a way to do so. To this end, we describe in this paper how large-scale system methods for solving multi-staged systems, such as Bender's Decomposition, high-speed sampling or Monte Carlo simulation, and parallel processors can be combined to solve some important planning problems involving uncertainty. For example, parallel processors may make it possible to come to better grips with the fundamental problems of planning, scheduling, design, and control of complex systems such as the economy, an industrial enterprise, an energy system, a water-resource system, military models for planning-and-control, decisions about investment, innovation, employment, and health-delivery systems.

Keywords:

linear programming; planning under uncertainty, large-scale systems, parallel processors, stochastic programming, decomposition principle

[1] Revision of an invited address before the XII[th] Mathematical Programming Symposium, August 1985, entitled "Need to do Planning Under Uncertainty and the Possibility of Using Parallel Processors for this Purpose"

Importance of Coming to Grips with Complexity

One approach to understanding complexity is to model large systems such as those of the economy or large-scale enterprises and engineering systems in the form of mathematical systems of linear and non-linear inequality constraints, and then seek a solution using computers that has desirable properties such as maximising some objective function. The solutions can then be used as guidelines for decision makers to follow. Parallel processors may make it possible to come to better grips with such problems.

Up to now, systems easiest to model and solve have been those whose resource availabilities, possible choices of technology, and future demands are assumed *known with certainty*. Industry and government routinely solve deterministic mathematical programs for planning and scheduling purposes, some involving thousands of variables and inequality constraints. As noted in the abstract, the solutions obtained are often ignored because they don't properly hedge against future contingencies. It is relatively easy to reformulate models to include uncertainty. The bottleneck has been (and is) our capability to solve them. The time is now ripe for finding a way to do so. To this end, we describe how large-scale system methods for solving multi-staged systems, such as Benders Decomposition, high-speed sampling, Monte Carlo simulation, and parallel procesors can be combined to solve some important planning problems involving uncertainty.

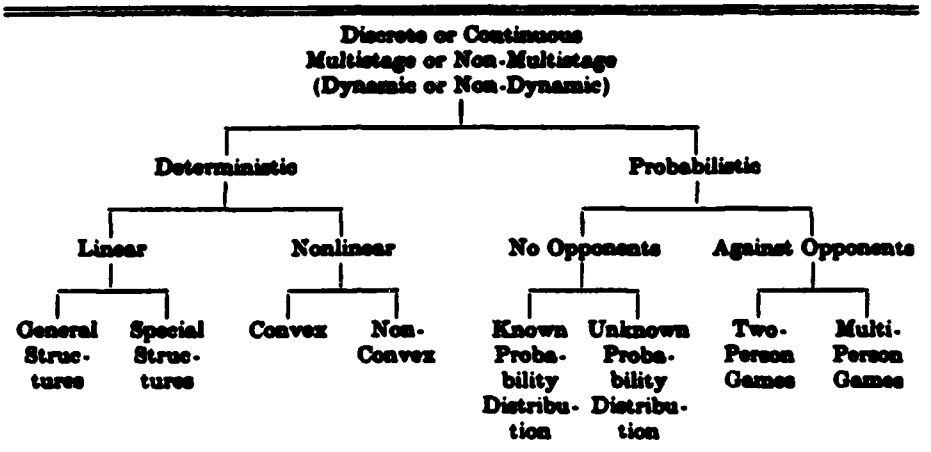
Section 1 describes the classical approach to solving large-scale dynamic systems using nested decomposition and how it can be applied to solving stochastic dynamic problems. Section 2 discusses the computer architecture requirements and configurations in a broad sense that are suitable for decomposition problems. Sections 3 and 4 detail the mathematical structure of the deterministic and stochastic case including, in particular, capacity planning problems under uncertainty. Section 5 discusses a possible application to an *intelligent control system with learning* in a multi-stage production system. Sections 6 and 7 describe on-going research and final comments.

1. PROGRESS TO DATE SOLVING LARGE-SCALE SYSTEMS

As early as 1970, it became evident that although much progress had been made in proposing methods for solving very large-scale systems, especially those involving uncertainty, these were for the most part ideas on paper – little or no testing had been done on practical problems. Because there was no systematic testing, papers in the literature were in reality little more than academic exercises in pure mathematics. In the 1970's, many places like Stanford's newly formed Systems Optimization Laboratory began to place greater emphasis on systematically testing algorithms on

real applications. Research has now reached the stage in which basic software tools are being used for extensive testing and development on deterministic dynamic linear programs.[2] Techniques under study include variants of the simplex method, interior solution methods, and nested decomposition.

Classification of Programming Problems[3]



SPECIAL CASES OF THE ABOVE

Linear inequality theory	Dynamic systems, Leontief models, Networks	Decreasing payoff, Chemical equilibrium, Convex programs	Increasing returns to scale, Many local maxima	Inventory control, Markov chains	Sequential decisions	Zero-sum games	Coalition theory

While there has been much progress on the deterministic class outlined on the left branch in the above diagram, the many problems involving uncertainty shown on the right branch are by far the most important and require for their solution computers more powerful or of a different "architecture" than those presently available.

1.1 Solving Stochastic Dynamic Programs

Stochastic dynamic programs or multi-stage programs can always be transformed into a deterministic one by a "certainty equivalent", i.e., by adding additional constraints. The resulting

[2] Dantzig, Dempster, and Kallio, eds. (1981) can be used as a general reference source of research work on uncertainty; this source also contains 25 pages of references.

[3] This is a photo copy of a diagram on page 8 of Dantzig (1963).

matrix structure consists of many submatrices strung together in a tree-like structure. While forming a certainty equivalent is theoretically possible, it is rarely a practical approach because of the exponential growth in the size of the system to be solved. Instead crude indirect methods are used to account for some of the uncertainty. For example, after an analyst solves a large dynamic linear program of an industrial complex, or a dynamic equilibrium-type model for studying the economy, he typically tests how sensitive his solution is to various uncertainties by calculating the effects of small changes in parameters and analyzes the effects of larger changes by solving a number of scenarios.

Other devices are also used. For example, the original problem is often treated as one of deciding "here-and-now" what *single* path of actions to take into the future rather than deciding here and now what myriad of alternative paths to take in the future, each path depending on which of a myriad of future events occur. The "system capacities expansion" problem, which we will discuss later, belongs to this single-path class.

In spite of inherent difficulties solving problems involving uncertainty, there has been active research extending large-scale deterministic methodology. Early examples are Dantzig (1955) and Charnes and Cooper (1955); a more recent one is Beale, Dantzig, and Watson (1986). The case of two-staged planning under uncertainty has been extensively studied by Wets and reported in Wets (1984); also Edwards (1985).

1.2 Nested Decomposition

The recent research of Birge (1980) for the multi-stage case under uncertainty, the dual nested-decomposition method of R. Wittrock (1983) for linear programs, and that of D. Scott (1985) for non-linear programs provides a promising starting methodology for attacking dynamic stochastic problems. Each time period is linked to the next time period by many subprograms in parallel each representing some random set of circumstances. Each of these in turn have subprograms in the next period in parallel and these in turn have many subprograms in parallel in the following period, etc. These subprograms fan out forming the tree-like structure referred to earlier. All arcs of the directed tree having a node in common correspond to the subprograms having the same starting conditions.

The general idea of dual nested decomposition is to provide each period with a set of additional constraints that it must obey if it is to generate outputs that allow future periods to be feasible and if feasible to take on values that include those that optimize the problem over all time periods. The iterative process sharpens these additional constraints, called "Benders Cuts", to the point

that they become both necessary and sufficient. These "cuts" are substitutes or surrogates for the future. The method moves "forward in time" by supplying starting conditions for the next period and "backwards in time" by generating cuts for the proceeding period. Eventually the method generates at most m additional inequalities that period t must satisfy (not counting those that are not tight in the final solution) where m is the number of equations in the period $t+1$ subproblem.

Parallel processors can be effectively employed in the dual nested decomposition approach to estimate the cuts by means of high-speed sampling of random events.

The general case of dual dynamic stochastic planning appears to be intractable by current methods even if parallel processors were inexpensive and available in quantity because of the proliferation in the number of branches in the tree of possible future paths. These grow exponentially as uncertain events follow upon uncertain events, on and on into the future. However, there are important subclasses, such as the one mentioned earlier of the determination of a plan for the expansion of the vector of capacities of a production system, which show much promise. For the latter, proliferation does not occur if the random events in a future period do not affect the amount of installed capacities, or the resources carried forward to the next period, or the random events of future periods. If so, only expected values (in a certain sense) associated with the random events of a period need be taken into account.

2. COMPUTER ARCHITECTURE

In the past, researchers in the large-scale linear programming area have been content to make the best use of the equipment available rather than to ask what future computer architectures would be best suited to their needs. But this may no longer be the case. Parallel computers which can be configured in various ways are becoming available and this has spurred interest in organizing the computer architecture to make it more suitable for solving mathematical programs. Thus, in addition to ongoing research developing, comparing and perhaps merging promising approaches to solve stochastic dynamic linear programs, such as special adaptations of the simplex method, interior methods, and dual-nested decomposition, there is now the opportunity to explore ways in which new architectures for supercomputers — parallel and pipeline — can also be used to improve efficiency.

It might seem that one way to modify *LP* computations to fit new architectures would be to take advantage of their vector processing capabilities. However, present-day codes are efficient with commercially available hardware because the software algorithms have been cleverly designed to maintain during the course of the computations to the maximum extent possible the sparsity of

non-zero coefficients of the initial data. Any obvious, simple-minded adaptation of current software to parallel processors which does not maintain sparsity but stores zeros explicitly is likely to prohibitively increase storage requirements and not show much improvement, if any, over existing methods.

In the dual nested-decomposition approach, it is possible to arrange the computation so that each stage gathers dual "price" information from later stages and "starting condition" information from early stages. We believe for the deterministic case that efficient algorithms using this approach can be devised which allow all computations to proceed in parallel while not stopping to wait for the most current information to be communicated.

In the analogous stochastic problem, parallel processors at the next-to-last time period (or stage) could be receiving as input the average of the dual vectors generated so far by the random events of the next stage. This process can be repeated back to the first time stage. All of the processors at all stages could be running in parallel, each feeding information to earlier and later stages. In this configuration, there could be one or many computers at each node of the branching tree busily doing high speed sampling of random events.

As noted earlier, for the most general case of planning under uncertainty there are severe limitations to this extension of the nested decomposition approach because of the exponential growth in the size of the problem. For the special case of the capacities expansion problem which we will discuss later, the prospects are much brighter since there is only a single path of capacities to be decided upon here and now for future periods. The capacities expansion problem, therefore, *is a natural one* to develop because there is no special need to warp existing algorithms into a parallel form. Indeed the parallel computer architecture required need be of nothing more sophisticated than a string of inexpensive personal computers tied together by a simple closed-loop bus; the PCs become high-speed sampling devices which simulate the uncertain events.

We will now discuss the mathematical structure of the multi-period and multi-staged planning problems beginning with the deterministic case. It will enable us to see why the problem grows exponentially in size when we extend its structure to include the general stochastic case. It will also become clear why, if we restrict the extension to the important class of investment or capacity expansion planning over time, the size does not grow exponentially. For the latter, we decide here and now what our expansion plan over time will be. There is no need for in-course corrections (branching) later on because the random events that occur within a period do not affect capacities

carried forward to subsequent periods. The formulation assumes that the random events of one period are independent of those of subsequent periods.

Not all problems involving uncertainty, however, fit this "here and now" mold. Sometimes a high demand for a product in one period implies a trend towards higher demand in the future. What one can do in this more general case is to limit the number of alternative future paths considered to a few main trends or contingencies that can happen in the future. In practice, this is done by boldly pruning the decision tree down to a few branches in order that the resulting model be of a size that it can be solved. [Wets 1966, Birge 1980, Dantzig 1963.]

3. MATHEMATICAL STRUCTURE OF THE DETERMINISTIC CASE

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 $T = 3$ periods:

$$\begin{bmatrix} A_{11} & & \\ A_{21} & A_{22} & \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{1}$$

By the introduction of in-process inventories and other devices linear programs of lower block-triangular type are mathematically equivalent to *staircase* or *multistage* problems of the form:

FIND $\min Z$ and vectors $X_t > 0$, such that

$$\begin{aligned} b_1 &= A_1 X_1 \\ b_2 &= -B_1 X_1 + A_2 X_2 \\ &\vdots \quad\quad \ddots \\ b_t &= \quad -B_{t-1} X_{t-1} + A_t X_t \\ &\vdots \quad\quad\quad\quad \ddots \\ b_T &= \quad\quad\quad\quad -B_{T-1} X_{T-1} + A_T X_T \\ (\min)\ Z &= c_1 X_1 \quad + \cdots + c_t X_t + \cdots \quad\quad + c_T X_T \end{aligned} \tag{2}$$

where matrices A_t, B_t and vectors b_t, c_t are given [Dantzig 1963]. A number of promising methods for solving such systems are known such as: Dantzig and Wolfe (1960), Glassey (1973), Ho and Loute (1980), Ho and Manne (1974), Bisschop and Meeraus (1977), Fourer (1982, 1983, 1984), Nishiya (1983), Jackson and Lynch (1986); for a general reference, see Dantzig, Dempster, and Kallio, eds. (1981).

4. MATHEMATICAL STRUCTURE OF STOCHASTIC CASE

For a very general class of stochastic planning problems, the values of b_t, B_{t-1}, A_t, c_t for $t > 1$ are not known to the planner with certainty at time 1 but could become known to him at some later time $\tau \leq t$. The value τ itself could be a random variable and there could be a different τ for every element of the matrices and vectors. While the values of these matrices may not be known, their probability distributions could 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 of Z is minimum. We now give reasons why this *general class* of stochastic problems is likely to remain intractable in the foreseeable future with or without the availability of parallel processors. We will then discuss less general, but important classes of stochastic problems which at present are not practical to solve on mainframes but could become so using parallel processors.

4.1. Two-stage Stochastic Case

We begin with the simplest *two-stage case* first studied in Dantzig (1955) and developed by R. Wets (1966, 1984); see also E. Edwards (1985):

$$\begin{aligned} b_1 &= A_1 X_1 && (X_1, X_2) \leq 0 \\ b_2 &= -B_1 X_1 + A_2 X_2 \\ (\min)\ Z &= c_1 X_1 \quad + c_2 X_2 \end{aligned} \tag{3}$$

where the first stage (b_1, A_1, c_1) are known with certainty while those of the second stage can take on possibly a continuum of values $(b_2(\omega), c_2(\omega), B_1(\omega), A_2(\omega))$ with probability (density) distribution $p_2(\omega)$ for ω in Ω, or a discrete probability distribution $p_2(\omega)$ where $\omega = 1, 2, \ldots, K$. The values of ω in Ω may therefore have a continuum of values, or it may take on a finite or an infinite set of discrete values.

For (3), if the parameter ω takes on say $K = 3$ distinct values, the stochastic problem of minimizing expected costs under uncertainty has *as its certainty equivalent* the deterministic linear program:

Find $\min Z,\ X_1 \geq 0, X_2(\omega) \geq 0, \omega = 1,2,3$:

$$
\begin{array}{lllll}
b_1(1) = & A_1X_1 & & & \\
\cdots\cdots & & & & \\
b_2(1) = & -B_1(1)X_1 + & A_2(1)X_2(1) & & \\
b_2(2) = & -B_1(2)X_1 & & + \quad A_2(2)X_2(2) & \\
b_2(3) = & -B_1(3)X_1 & & & + \quad A_2(3)X_2(3) \\
\cdots\cdots & & & & \\
\min Z = & c_1X_1 & +p_2(1)\, c_2(1)X_2(1) & +p_2(2)c_2(2)X_2(2) & +\ p_2(3)c_2(3)X_2(3)
\end{array}
\tag{4}
$$

To simplify the discussion that follows, when optimal solutions exist we can assume without loss of generality $c_2(\omega) \geq 0$; moreover, except as noted otherwise, B_1 is assumed independent of ω.

Typically this problem is solved using "Benders" decomposition (see Benders (1962)). The key idea is to replace the objective function contribution of the second period variables by a scalar θ_1, and to replace the second period constraints — those shown in (4) between the dashed lines — by a set of inequalities expressed in terms of X_1 and θ_1 only, which are necessary conditions for feasible and optimal solutions to (4). These necessary conditions, (5.1), called "cuts", are added sequentially ($\ell = 1, 2, \ldots$) 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 method has the form:

MASTER PROBLEM. FIND $\min Z$, $X_1 \geq 0$, $\theta_1 \geq 0$,

$$
\begin{aligned}
& b_1 = A_1X_1, \\
\text{CUTS:} \quad & g_1^\ell \leq -G_1^\ell X_1 + \delta_1^\ell \theta_1, \qquad \ell = 1, \ldots, L \\
& \min Z = c_1X_1 + \theta_1
\end{aligned}
\tag{5.1}
$$

where $\delta_1^\ell = 0$ for feasibility cuts if the corresponding subproblem is infeasible, and $\delta_1^\ell = 1$ for optimality cuts if the corresponding subproblem is feasible. The solution $X_1 = X_1^*$ obtained is then "tested" by solving the corresponding subproblem (6) below to see (i) if the starting conditions it implies for the second period allows a feasible solution for the second period, and (ii) if it together with the optimal solution to the second period provides a global optimum to the original problem. The optimal $\pi_2(\omega)$ to (6) are used to generate the next cut $\ell+1$ which is then added to those already generated where

$$
g_1^{\ell+1} = \sum_\omega \pi_2(\omega) b_2(\omega); \quad G_1^{\ell+1} = \sum_\omega \pi_2(\omega) B_1(\omega).
\tag{5.2}
$$

Assuming B_1 is independent of ω, no more than $L \leq m_2$ of the cuts will be tight on any major iteration, where m_2 is the number of rows in B_1. This is so because G_1^ℓ is geneated by linear combinations of the rows of B_1 and hence has rank $\leq r$ where $r \leq m_2$ is the rank of B_1. The remainder may be dropped (possibly to be regenerated on some later iteration).

This second period problem is called the "*subproblem*" or "SUB" for short. The SUB *decomposes* into K independent problems:

For each ω in Ω, solve the *SUB PROBLEM*: Find $\min Z_2(\omega), X_2(\omega) \geq 0$:

$$\begin{array}{lll} & & \text{Dual Prices} \\ A_2(\omega)X_2(\omega) & = b_2(\omega) + B_1 X_1^* & : \pi_2(\omega) \\ p_2(\omega) \cdot c_2(\omega)X_2(\omega) & = Z_2(\omega)(\min) & \end{array} \tag{6}$$

where $\Omega = \{\omega | \omega = (1, \ldots, K)\}$. These problems are solved for $\omega = 1, \ldots, K$ and their optimal prices or feasibility prices $\pi(\omega)$ are computed to test feasibility and optimality. If the test fails the expected values of $g_1^{\ell+1} = \mathcal{E}\,\pi_2(\omega)b_2(\omega)$ and $G_1^{\ell+1} = \mathcal{E}\,\pi_2(\omega)B_1$ are used to generate new cut conditions to augment those of (5.1). Such an approach, however, is clearly only practical when K is small. When K is large or infinite or varies as a continuous parameter, parallel *computers* could be used as high-speed sampling devices to effectively solve such problems. One computer at the MASTER level serves as an *integrator*. It is sequentially receiving as input estimates of cuts (5.1) and solves (5), (5.1) to optimality with the estimates it has received so far; and generating, as output, revised $X_1 = X_1^*$. This process also provides a *lower bound* estimate for $\min Z$ which monotonically increases with each solution of the master problem.

Several parallel computers could be at the SUB level each having as input the latest value of X_1^* and solving (6) in dual form for many random choices of ω. When c_2, A_2 are the same for all ω, the dual of (6) is an L.P. with only the dual objective $b_2(\omega)$ changing. To provide cuts for the MASTER, the parallel processors are used to determine the expected values of $\pi_2(\omega)$ and $X_2(\omega)$ or to approximate them by means of a large enough sample.

If it is practical to solve (6) for all ω, the solutions to (6) would generate a valid cut and a correct *upper bound* estimate for $\min Z$. In that case, the difference between the lower bound and upper bound estimates can then be used to test optimality of X_1^* for the original problem.

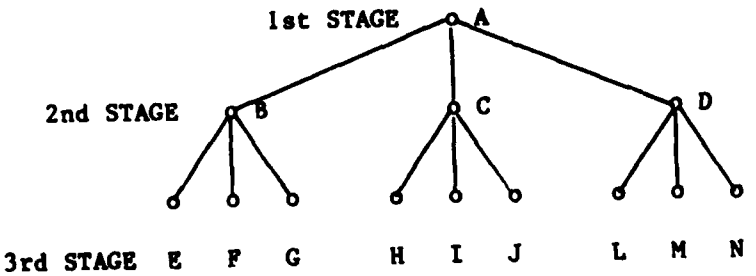
For the cases where K is large, infinite, or ω is continuous, it is no longer possible, in general, to solve (6) for all ω. Instead solutions to the SUB *based on a sample* of ω in Ω is used as the estimate. A

key aspect of the current research is concerned with the *sampling* error of these estimates, particularly the sampling error for the difference between the upper bound and lower bound estimates of Z which is used as a stopping rule for terminating the computation.

Variance reduction techniques are often utilized in sampling to limit and decrease the total computational work. The three most successful of these are for: 1) control variables, described in Lavenberg and Welch (1981), and Rubinstein and Marcus (1985); 2) importance sampling, and 3) stratified sampling. See Bratley, Fox, and Schrage (1983).

4.2. Multi-stage Stochastic Planning Problem.

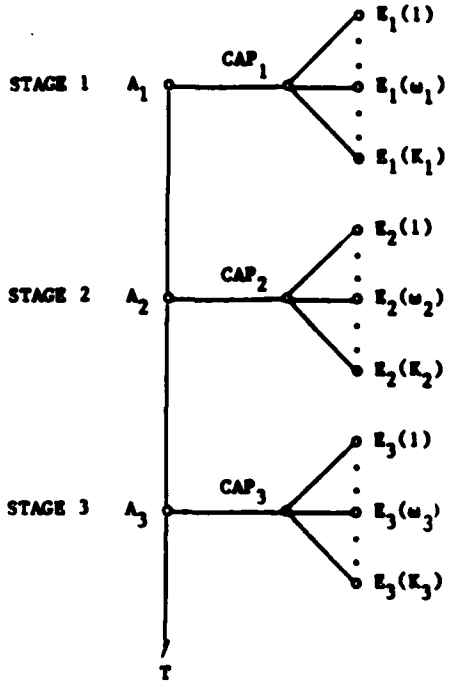
The two-staged case (4) represents the simplest application of parallel computers to stochastic programming. The general multi-staged case, using the corresponding "reduction" to the equivalent deterministic linear program, however, becomes intractable due to the exponential increase in the number of possible outcomes, see (7a) below for the case of $T = 3$.



(7a)

In the first stage we have only one set of constraints for A, the same as (5.1). In the second stage we could have K sets of constraints, shown as B, C, D in (7a), but there could be many more, see (4). If each of these in turn has K outcomes in the third state, then we have a situation of $K \times K$ sets of constraints in stage 3 shown as $(EFGHIJLMN)$. It is now easy to see why, for $K = \infty$ or Ω continuous, the proliferation of cases is out of hand. Even if large numbers of inexpensive parallel processors were available, it does not seem to be a practical way to solve stochastic problems in general.

We therefore restrict our discussion to an important subclass of multi-stage uncertainty problems whose tree-like matrix structure is more tractable, see (7b) below.



(7b)

The mathematical structure for T stages will be described later. For two stages, it has the form;

FIND $\min Z$, vectors $X_t \geq 0$, $U_t(\omega_t, \bar{\omega}_t) \geq 0$, $\omega_t \in \Omega_t$, $\bar{\omega}_t \in \bar{\Omega}_t$ such that

$$
\begin{array}{lllll}
b_1 & = A_1 X_1 & & & \\
0 & = -F_1(\omega_1) X_1 & & + D_1 U_1(\omega_1, \bar{\omega}_1) & \\
d_1(\bar{\omega}_1) = & & & E_1 U_1(\omega_1, \bar{\omega}_1) & \\
b_2 & = -B_1 X_1 & + A_2 X_2 & & \\
0 & = & - F_2(\omega_2) & & + D_2 U_2(\omega_2, \bar{\omega}_2) \\
d_2(\bar{\omega}_2) = & & & & E_2 U_2(\omega_2, \bar{\omega}_2) \\
Z & = c_1 X_1 & + c_2 X_2 & + \mathcal{E} f_1 U_1(\omega_1, \bar{\omega}_1) & + \mathcal{E} f_2 U_2(\omega_2, \bar{\omega}_2)
\end{array}
\tag{8}
$$

where $\mathcal{E}$ denotes expectation over $\omega_t \in \Omega_t, \bar{\omega}_t \in \bar{\Omega}_t$.

An important application in this class is *capacity planning problems under uncertainty*. A case in point is the fleet planning by airlines — the problem is to plan over time the number of aircraft of various types to have in the fleet. Within each time period t (say one year), the fleet composition CAP_t is changed by buying new aircraft and selling off obsolete ones. Another good example is the expansion of generating capacity and transmission line capacity of electric utilities. For either example, the vector of capacities CAP_t in operating condition and the demand are treated as uncertain events in period t — shown as $E_t(1), \ldots, E_t(\omega_t), \ldots$ in the diagram.

For this class of planning problems, the installed capacity is assumed not to be affected by particular events that happen during the course of a year, i.e., by a particular value of the uncertain demands or by a particular repair status of the capacity. Only the *expected revenues and expected failures* (in some sense) to meet demands in the period affect the decision to invest in new capacity or to get rid of old capacity. Note that our assumptions imply a "here and now" decision, namely the solution takes the form of a single path over time and not one with branches for different contingencies that arise in the future such as a high demand in year t implying a higher probability of a high demand in year $t+1$.

For this class of problems, the mathematical structure for capacity formation itself would be the same as in (2), namely:

$$b_t = -B_{t-1}X_{t-1} + A_tX_t\,, \quad \text{for } t = 1,\ldots,T \text{ and } B_0X_0 \equiv 0\,, \tag{9}$$

where $X_t \geq 0$ are the planned capacities (to be determined) for period t at a cost $\bar{Z} = \Sigma c_t X_t$ where b_t, A_t, B_t, c_t are all known with certainty. The subproblem for period t is defined for some $X_t = X_t^*$ by first finding the amount of capacity available for use in period t:

$$F_t(\omega_t)X_t^*, \qquad \omega_t \text{ in } \Omega_t\,, \tag{10}$$

which depends on X_t^* and a random variable ω_t measuring the proportion of X_t^* requiring repair. The probability distribution of ω_t is assumed known independent of X_t^*. Letting $d_t(\bar{\omega}_t)$ be uncertain demand, we then solve for the dual multipliers and Z_t that solves (11) below for each random choice of right-hand side, i.e., for ω_t in Ω_t and $\bar{\omega}_t$ in $\bar{\Omega}_t$, in order to determine their expected values. If the number of discrete values of $\omega_t, \bar{\omega}_t$ are small, this can be done exactly. If large, then their expected values can be estimated by sampling.

		Dual Prices
$D_t\, U_t(\omega_t,\bar{\omega}_t) = F_t(\omega_t)X_t^*\,,$	$U_t(\omega_t,\bar{\omega}_t) \geq 0\,,$	$: \pi_t(\omega_t,\bar{\omega}_t)$
$E_t\, U_t(\omega_t,\bar{\omega}_t) = d_t(\bar{\omega}_t)\,,$	$\bar{\omega}_t \text{ in } \bar{\Omega}_t\,,$	$: \bar{\pi}_t(\omega_t,\bar{\omega}_t)$
$f_t U_t(\omega_t,\bar{\omega}_t) = Z_t(\min)$		

(11)

where corresponding dual prices are shown on the right. These samples can be used to estimate expected cuts $\ell = 1,2,\ldots$ of the form

$$g_t^\ell \leq G_t^\ell X_t + \delta_t^\ell \theta_t\,; \tag{12.1}$$

where $\delta_t^\ell = 0$ if (11) is infeasible, else $\delta_t^\ell = 1$, and where

$$g_t^\ell = \mathcal{E}\,\pi_t(\omega_t, \bar{\omega}_t) d_2(\bar{\omega}_t); \quad G_t^\ell = \mathcal{E}\,\pi_t(\omega_t, \bar{\omega}_t) F_t(\omega_t) \tag{12.2}$$

and a cost form

$$Z = \sum_t (c_t X_t + \theta_t)\,. \tag{13}$$

which, together with (9), form the MASTER PROBLEM (14) below.

Several parallel processors could be assigned to each stage. The processors receive X_t^* as inputs from the MASTER PROGRAM generate random $\omega_t, \bar{\omega}_t$ and solve the sub problems, and give back to the MASTER approximate cuts (12.1) obtained by sampling. These cuts augment those obtained earlier.

MASTER PROBLEM FOR A 3-STAGE PROBLEM.

FIND $\min Z,\ X_t \geq 0, \theta_t \geq 0$:

$$\begin{aligned}
b_1 &= A_1 X_1 \\
b_2 &= -B_1 X_1 + A_2 X_2 \\
b_3 &= -B_2 X_2 + A_3 X_3 \\
g_1^{\ell_1} &\leq G_1^{\ell_1} X_1 + \delta_1^{\ell_1} \theta_1 \\
g_2^{\ell_2} &\leq G_2^{\ell_2} X_2 + \delta_2^{\ell_2} \theta_2 \\
g_3^{\ell_3} &\leq + G_3^{\ell_3} X_3 + \delta_3^{\ell_3} \theta_3 \\
\min Z &= c_1 X_1 + \theta_1 + c_2 X_2 + \theta_2 + c_3 X_3 + \theta_3
\end{aligned} \tag{14}$$

where $\ell_1 = 1, \ldots, L_1,\ \ell_2 = 1, \ldots, L_2,\ \ell_3 = 1, \ldots, L_3$.

Note that the main MASTER PROBLEM has the form of a deterministic staircase system which can be solved directly; or, if too large, can be solved using the dual nested decomposition approach, see for example Abrahamson (1983) and Wittrock (1983). The advantage of the dual-decomposition approach is that the parallel processors at each stage can be used effectively to provide information in the form of X_{t+1}^* to the stage below and to pass back cuts generated in stage t to stage $t-1$.

5. INTELLIGENT CONTROL SYSTEMS WITH LEARNING

In addition to the two important applicational areas for using parallel processors for planning under uncertainty — two-stage planning and capacity planning over time — there is the possibility

of applying it to an automated production line. Envision a *multi-stage production system* with sensors feeding information to computers which control each stage. Initially the control system is provided with a set of general ground rules which is believed will keep the quality of the product under control under a variety of random conditions and keep its variability to a minimum.

Suppose during production the sensors at some "down-stream" stage detect a loss of quality. Under current procedures, feedback controls at that stage are activated to try to correct for the loss of quality. Suppose we apply the same Benders Decomposition approach except to a multi-stage production system instead of to a time-stage system. Then in addition to the immediate feedback control, the "cuts" passed back to the computer controlling the prior stage in the process can be interpreted as ground rules that must be followed. This in turn generates cuts for stages further upstream to obey. These cuts not only correct for the present "down-stream" loss of quality, but cleverly erect a "barrier" that future states of the system must not cross. These can be "remembered" by the computer in the form of a growing collection of necessary conditions (rules) which must not be violated in the future. As new contingencies arise down-stream, the *computers continuously learn from experience by adding to their set of control constraints and thus become a control system with memory and learning.* [4]

6. CURRENT RESEARCH AND TEST PROBLEMS

The Systems Optimization Laboratory at Stanford is currently doing research on a deterministic version of the PILOT Macro-Economic Model for the U.S. (see Dantzig, McAllister and Stone 1986) and a simplified version of the electric-power expansion problem in stochastic form under the general guidance of Mordecai Avriel and the author (see Bellovin 1986, and Yang 1985). The PILOT Model is used for studying the impact of innovation, modernization, foreign competition, technology assessment, and changes in economy due to shortages or high-cost energy supply (McAllister and Stone, 1984). The electric-power generation and distribution problem is based on the electric power generation and distribution network for California. We are currently using a version of the latter two models for testing the applicability of using parallel processors.

[4] Pereira, Monticello, and Dantzig apply this approach to scheduling electric power systems with security constraints (in progress).

7. FINAL COMMENTS

Planning under uncertainty in the author's opinion is the most important outstanding problem of Operations Research. In this paper we discussed the status of solving large-scale dynamic systems and what types of large-scale stochastic time-stage systems are likely to remain intractable. What we believe is tractable is the solution of an important subclass which we referred to as "capacity planning" over time. Examples include fleet planning problems for airlines or planning the expansion of generators and transmission lines for multiple-area electric-power systems. We described an approach to solving such problems using decomposition methods and parallel processors as high-speed sampling devices.

Research has only just begun in this important field. The classical approach has been to try to reduce stochastic models to a "certainty equivalent" and to solve them by special large-scale deterministic methods. In the author's opinion this approach remains important but limited in scope. On the other hand, using Monte Carlo simulation in combination with large-scale methods, while having a broader applicability, also has its limits, even if parallel processors were inexpensive and available in quantity. A key problem that future research must solve is finding ways to keep sample size small.

BIBLIOGRAPHY

Abrahamson, P.G. (1983). A nested decomposition approach for solving staircase linear programs, Report SOL 83-4, Department of Operations Research, Stanford University, Stanford, California.

Beale, E.M.L., Dantzig, G.B. and Watson, R.D. (1986). A First Order Approach to a Class of Multi-Time-Period Stochastic Programming Problems, **Mathematical Programming Study 27**, pp. 103-117.

Bellovin, M.S. (1986). Reliability and shortage distribution computations in general stochastic transportation networks, Report SOL 86-4, Department of Operations Research, Stanford University, Stanford, California.

Benders, J.F. (1962). Partitioning procedures for solving mixed-variable programming problems, **Numerische Mathematik 4**, pp. 238-252.

Birge, J.R. (1980). Solution methods for stochastic dynamic linear programs, Report SOL 80-29, Department of Operations Research, Stanford University, Stanford, California.

Birge, J.R. (1984). Aggregation in Stochastic Linear Programming, **Mathematical Programming 31**, pp. 25-41.

Birge, J.R. and J.B. Wets (1986). Designing Approximation Schemes for Stochastic Optimization Problems, in Particular for Stochastic Programs with Recourse, **Mathematical Programming Study 27**, pp. 54-102.

Bisschop, J. and Meeraus, A. (1977). Matrix augmentation and partitioning in the updating of the basis inverse, **Mathematical Programming 13**, pp. 241-254.

Bratley, P., Fox, B., and Schrage, L. (1983). A Guide to Simulation. Springer-Verlag, New York.

Charnes, A., Cooper, W.W. (1955). Constrained programming, **Management Science 6 (1)**, pp. 131-172.

Dantzig, G.B. (1955). Linear programming under uncertainty, **Management Science, 1**, pp. 197-206.

Dantzig, G.B. (1963), **Linear Programming and Extensions**, Princeton University Press, Princeton.

Dantzig, G.B. (1980). Time-staged linear programs, Report SOL 80-28, Department of Operations Research, Stanford University, Stanford, California.

Dantzig, G.B., Dempster, M.A.H. and Kallio, M.J. (eds.) (1981). **Large-Scale Linear Programming (Volumes 1, 2)**, IIASA Collaborative Proceedings Series, CP-81-51, IIASA, Laxenburg, Austria.

Dantzig G.B., and Perold, A.F. (1979). A basic factorization method for block triangular linear programs, Report SOL 78-7, Department of Operations Research, Stanford University, Stanford, California; Sparse Matrix Proceedings, 1978 (Iain S. Duff and G.W. Stewart, eds.), SIAM, 1979, pp. 283-312.

Dantzig, G.B., P.H. McAllister and J.C. Stone (1987), Using the PILOT Model to Study the Effects of Technological Change, **Strategic Planning in Energy and Natural Resources**, Proceedings of the 2nd Symposium on Analytic Techniques for Energy, Natural Resources and Environmental Planning (B. Lev, J. Bloom, A. Gleit, F. Murphy and C. Schoemarker, eds.), April 1986.

Dantzig, G.B. and Wolfe, P. (1960). The decomposition principle for linear programs, **Operations Research 8**, pp. 110-111.

Edwards, E. (1985). Documentation for the ADO/SDS collection of stochastic programming codes," IIASA WP-85-02.

Emoliev, Y. (1983), Stochastic Quasigradient Methods and Their Applications to Systems Optimization, **Stochastics 9**, pp. 1-36.

Fourer, R.H. (1982). Solving staircase linear programs by the simplex method, 1: Inversion, **Mathematical Programming 23**, pp. 274-313.

Fourer, R.H. (1983). Solving staircase linear programs by the simplex method, 2: Pricing, **Mathematical Programming 25**, pp. 251-292.

Fourer, R.H. (1984). Staircase matrices and systems, **SIAM Review 26**, pp. 1-70.

Glassey, R. (1973). Nested decomposition and multi-stage linear programs, **Management Science 20**, pp. 282-292.

Ho, J.K. and Loute, E. (1980). An advanced implementation of the Dantzig-Wolfe decomposition algorithm for linear programming, Discussion Paper 8014, Center for Operations Research and Econometrics (CORE), Belgium.

Ho, J.K. and Manne, A.S. (1974). Nested decomposition for dynamic models, **Mathematical Programming 6**, pp. 121-140.

Jackson, P.L. and D.F. Lynch (1982). Revised Dantzig-Wolfe Decompositon for Staircase-Structed Linar Programs, Technical Report 558, School of Operations Research and Industrial Engineering, Cornell University (revised 1985). To appear in **Mathematical Programming**.

Kall, P. (1979). Computational Methods for Solving Two-Stage Stochastic Linear Programming Problems, **Z. Angew. Math. Phys. 30**, pp. 261-271.

Kallberg, J.G., R.W. White and W.T. Ziemba (1982). Short Term Financial Planning Under Uncertainty, **Management Science 28**, pp. 670-682.

Lavenberg, S.S. and Welch P.D. (1981). A perspective on the use of control variables to increase the efficiency of Monte Carlo simulation. **Management Science, 27**, pp. 322-335.

Louveaux, F.V. (1986). Multistage stochastic programs with block-separable recourse, **Mathematical Programming Study 28**, pp. 48-62.

McAllister, P.H. and Stone, J.C. (1984). Will planned electric capacity limit economic growth?, Report SOL 84-1, Department of Operations Research, Stanford University, Stanford, California.

Nazareth, L. and R.J-B. Wets (1986). Algorithms for Stochastic Programs: The Case of Nonstochastic Tenders, **Mathematical Programming Study 28**, pp. 1-28.

Niederreiter, H. (1986). Multidimensional numerial integration using pseudo random numbers, **Mathematical Programming Study 27**, pp. 17-38.

Nishiya, T. (1983). A basis factorization method for multi-stage linear programming with an application to optimal operation of an energy plant (draft report).

Pereira, M.V.F., Monticelli, A., Dantzig, G.B. A mathematical framework for the economic scheduling of electrical power systems with security constraints, research paper in progress.

Prekopa, A., (1978). Dynamic Type Stochastic Programming Models, in **Studies in Applied Stochastic Programming**, (A. Prekopa, ed.), Hungarian Academy of Science, Budapest, pp. 179-209.

Rubinstein, R.Y. and Marcus, R. (1985). Efficiency of multivariate control variates, in Monte Carlo simulation. **Operations Research, 33**, pp. 661-677.

Scott, D.M. (1985). A dynamic programming approach to time-staged convex programs, Report SOL 85-3, Department of Operations Research, Stanford University, Stanford, California.

Strazicky, B. (1980). Computational Experience with an Algorithm for Discrete Recourse Problems, in **Stochastic Programming**, (M. Dempster, ed.), Academic Press, London, pp. 263-274.

Wets, R.J. (1966). Programming under uncertainty: the equivalent convex program, **J. SIAM Applied Math. 14 (1)**, pp. 89-105.

Wets, R.J. (1984). Large-scale linear programming techniques in stochastic programming, **Numerical Methods for Stochastic Optimization**, (Y. Ermoliev and R. Wets, eds.), Springer-Verlag. IIASA WP-84-90.

Wets, R.J. (1955). On Parallel Processor Design for Solving Stochastic Programs, in **Proceedings of the 6th Mathematical Programming Symposium**, Japanese Mathematical Programming Society, Tokyo, pp. 13-36.

Wittrock, R.J. (1983). Advances in a nested decomposition algorithm for solving staircase linear programs, Report SOL 83-2, Department of Operations Research, Stanford University, Stanford, California.

Yang, P. (1985). Computation of reliability and shortage distributions in stochastic transportation networks with cycles, Report SOL 85-9, Department of Operations Research, Stanford University, Stanford, California.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: SOL 87-1 (revision of 86-11)	2. GOVT ACCESSION NO. AD-A180 154 / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Planning Under Uncertainty Using Parallel Processors Computing	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): George B. Dantzig	8. CONTRACT OR GRANT NUMBER(s): N00014-85-K-0343
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Operations Research - SOL, Stanford University, Stanford, CA 94305	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR-047-064
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research - Dept. of the Navy, 800 N. Quincy Street, Arlington, VA 22217	12. REPORT DATE: January 1987
	13. NUMBER OF PAGES: 19 pp.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

linear programming
large-scale systems
stochastic programming
planning under uncertainty
parallel processors
decomposition principle

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

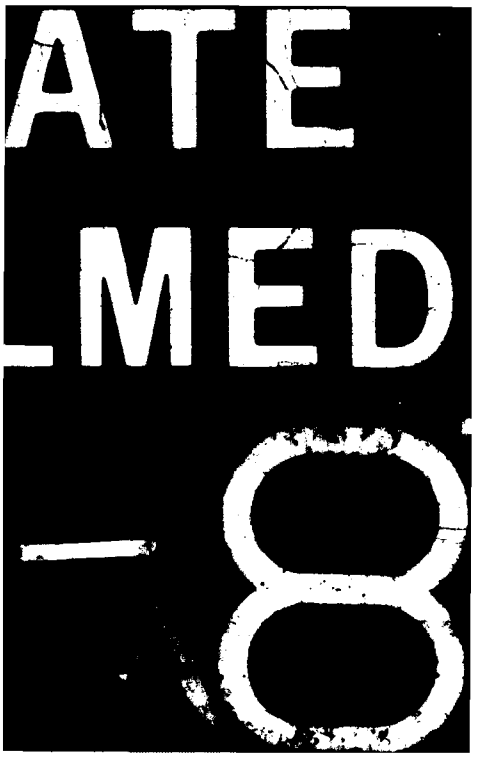
See next page.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOL 87-1: "Planning Under Uncertainty Using Parallel Computing,"
by George B. Dantzig

Industry and government routinely solve deterministic mathematical programs for planning and scheduling purposes, some involving thousands of variables with a linear or non-linear objective and inequality constraints. The solutions obtained are often ignored because they don't properly hedge against future contingencies. It is relatively easy to reformulate models to include uncertainty. The bottleneck has been (and is) our capability to solve them. The time is now ripe for finding a way to do so. To this end, we describe in this paper how large-scale system methods for solving multi-staged systems, such as Bender's Decomposition, high-speed sampling or Monte Carlo simulation, and parallel processors can be combined to solve some important planning problems involving uncertainty. For example, parallel processors may make it possible to come to better grips with the fundamental problems of planning, scheduling, design and control of complex systems such as the economy, an industrial enterprise, an energy system, a water-resource system, military models for planning-and-control, decisions about investment, innovation, employment, and health-delivery sytems.


ATE
LMED
8