164

WP: 164

Working Paper

IIM
WP-164

विद्याविनियोगाद्विकासः
IIM
AHMEDABAD

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD

ON WAIT-AND-SEE STOCHASTIC LINEAR PROGRAMS:
AN APPLICATION AND AN ALGORITHM

by

NITIN R. PATEL

W P No. 164
JUNE 1977

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD

ABSTRACT

This paper develops a computational algorithm for estimating the mean objective function value of a stochastic linear programming problem of the passive or wait-and-see type. The algorithm is applied to a problem connected with design of a 'milk-grid' in India and is found to be computationally effective in that case. It is most likely to be useful in the case of fairly large LP problems with a few (<10) stochastic right hand side variables.

ON WAIT-AND-SEE STOCHASTIC LINEAR PROGRAMS: AN APPLICATION AND AN ALGORITHM

by
Nitin R. Patel

Introduction

This paper is concerned with a particular type of stochastic linear programming problem that arises in practice. The type of situation where this problem occurs is best illustrated by a capacity expansion decision. Suppose that there is a system whose operational characteristics in a single period can be modelled as a linear program. If we are interested in augmenting the capacity of this system and that the augmented capacity will be available over a number of periods, and if the demands and supplies in the system are stochastic with known distributions, it is useful to know what the average performance of the system will be under various levels of additional capacity. A concrete situation in which this type of problem arose was in connection with a study of the potential for establishing a 'milk-grid' in India.
A major programme termed 'Operation Flood' has been set up to step up milk production in milk-shed areas adjacent to the four major cities of Bombay, Calcutta, Delhi and Madras. Milk production in all four

areas has been increasing. However, due to vagaries of the monsoon there are substantial year-to-year fluctuations in milk production and collection in each area. The main idea behind the 'milk-grid' was to provide transport capacity to move milk between the areas in order to improve the average price to the milk supplier. In the absence of such capacity, prices tended to plunge drastically in times of high supply. In fact, there were even occasions when the dairies did not lift offered milk due to their processing capacity limitations. It was felt that maintaining attractive prices was critical in sustaining long-term trends of increasing milk production and that the 'milk-grid' could contribute to this objective.

The specific stochastic linear programming problem considered here is of the "wait-and-see"[1,2] or "passive"[3,4] type as opposed to the "here-and-now" or "active" type. In this problem some of the elements of the right hand side are assumed to be random variables with known distributions. The decision-maker makes decisions regarding the activity levels of the linear program <u>after</u> the realized values of the random variables are known. It is assumed that for any possible realization he always has a feasible solution. Also, these decisions are made for several periods independently, each period having a particular realization of the random variables. The focus of the algorithm is to provide estimates of the average or expected value of the objective function. In the 'milk-grid' situation, each period is a year. The decision-maker is assumed to have good estimates of the milk production obtaining in a period and will make optimal decisions in allocation of transport capacity (in the shape of rail tank-cars) for that period. What is of interest is to know how much on the average the objective function value will be for different levels of transport capacity.

In attempting to answer this question it is essential to develop an efficient computational algorithm for estimating the expected value of the objective function. In our review of the literature we found that the "wait-and-see" type of stochastic program has received much less attention than the "here-and-now" type, probably because the latter type of problem tends to crop up much more frequently. In fact, often the analysis of the "wait-and-see" program is carried out to develop insight into the behaviour of the "here-and-now" type[5]. The only article we found which attacked the computational problem of interest here was that of Berneau[6]. His procedure involves enumerating all basic feasible solutions and determining the probabilities of each being optimal. While this is workable for small problems, it is impracticable for the milk-grid problem which has about 20 rows and 40 columns. The algorithm we have developed will be of use for large problems in which the number of stochastic elements is small (less than 10). The algorithm is not practical for problems having a large number of stochastic right hand side elements.

Some results required in developing the algorithm

Let us establish our notation before going on to treating some results required in formulating our algorithm.

Let us suppose the decision maker solves problem (P) below in any period.

Problem (P)

$$z(e) = \text{Max } px$$

subject to

$$A_1 x \leq e$$

$$A_2 x \leq b$$

$$x \geq 0$$

where

p is a 1xn vector

b is an mx1 vector

e is a kx1 vector

x is a nx1 decision vector

A_1 is a kxn matrix

A_2 is an mxn matrix

Here e is the realization for that period of a random vector $\tilde{e}$. Thus, the objective function value $z(\tilde{e})$ is a random variable and we are interested in estimating its expected value from the known distribution of $\tilde{e}$.

Result 1

$z(e)$ is a concave function over the set F of all values of e such that (P) has a feasible solution.

Proof

Let $e_1, e_2 \in F$ and let x_1, x_2 be any feasible solutions to (P) with $e = e_1, e_2$ respectively. Let $e_3 = \lambda e_1 + (1 - \lambda) e_2$, $\lambda \in [0,1]$. Then $x_3 = \lambda x_1 + (1 - \lambda) x_2$ is a feasible solution to (P) with $e = e_3$. So that $z(e_3) \geq p\, x_3 = \lambda z(e_1) + (1 - \lambda) z(e_2)$

(Q.E.D.)

Note that $z(e)$ is also monotone and non-decreasing over the set F. (The proof is trivial).

Result 2

$z(e)$ is a continuous function of e over F. (For a proof see (7)).

Suppose that the random variable $\tilde{e}$ has a probability density function* $f(e)$ defined on E^k such that $f(e) > 0$ for points limited to a box** $S \subset F$.

Suppose the box S is partitioned into R boxes $S_1, S_2, \ldots, S_R$ so that $\bigcup_{r=1}^{R} S_r = S$ and $S_i \cap S_j = \emptyset$ for $i \neq j$.

Let $p_r = \int_{S_r} f(e)de$

We assume that $p_r > 0$ for $r = 1, 2, \ldots, R$

Define a random variable $\tilde{e}_r$ with the p.d.f. $f_r(e)$

Where $f_r(e) = \begin{cases} \dfrac{f(e)}{p_r} & \text{for } e \in S_r \\ 0 & \text{elsewhere} \end{cases}$

Let $\bar{e}_r = \int_{S_r} e\, f_r(e)\, de$ and $\tilde{e}_r = \begin{bmatrix} \tilde{e}^r_1 \\ \tilde{e}^r_2 \\ \vdots \\ \tilde{e}^r_k \end{bmatrix}$

---

*Could equally be a probability mass function or a mixture of a p.d.f. and a p.m.f.

**A box in E^k is a subset of E^k which is the Cartesian product of k closed and bounded intervals in E^1.

Lemma 1

$$E(z(\tilde{e})) = \sum_{r=1}^{R} p_r E(z(\tilde{e}_r))$$

Proof

$$E(z(e)) = \int_S z(e)f(e)de = \sum_{r=1}^{R} \int_{S_r} z(e)f(e)de$$

$$= \sum_{r=1}^{R} p_r \int_{S_r} z(e)f_r(e)de$$

$$= \sum_{r=1}^{R} p_r E(z(\tilde{e}_r))$$

(Q.E.D.)

Lemma 2

$$E(z(\tilde{e}_r)) \leq z(\bar{e}_r) \quad \text{for} \quad r = 1,2, \ldots R.$$

Proof

This follows directly from application of Jensen's inequality[8] to the random variable $\tilde{e}_r$ since z is a concave function over S_r.

(Q.E.D.)

From Lemmas 1 and 2 we get an upper bound (UB) on $E(z(\tilde{e}))$ as

$$UB = \sum_{r=1}^{R} p_r \, UB_r \quad \text{where} \quad UB_r = z(\bar{e}_r). \qquad \ldots\ldots \quad (1)$$

Lemma 3

Let $S_r = (x_1, y_1) \times (x_2, y_2) \times \ldots\ldots\ldots\ldots (x_k, y_k)$

and let
$$e(j_1, j_2, \ldots\ldots, j_k) = \begin{bmatrix} x_1^{1-j_1} + y_1^{j_1 - 1} \\ x_2^{1-j_2} + y_2^{j_2 - 1} \\ \vdots \\ x_k^{1-j_k} + y_k^{j_k - 1} \end{bmatrix}$$

so that $e(j_1, j_2, \ldots\ldots, j_k)$ for $j_1, j_2, \ldots\ldots, j_k = 0$ or 1

are the 2^k extreme points of S_r.

Then
$$E(z(\tilde{e}_r)) \geq \sum_{j_1 \ldots j_k = 0}^{1} \bar{\theta}(j_1, j_2 \ldots j_k) \, z\left[e(j_1, j_2 \ldots j_k)\right]$$

where $\bar{\theta}\ (j_1, j_2, \ldots\ldots, j_k) = E \left(\prod_{i=1}^{k} \left(\frac{e^r_i - x_i}{y_i - x_i} \right)^{j_i} \left(1 - \frac{e^r_i - x_i}{y_i - x_i} \right)^{1-j_i} \right) \qquad j_1, j_2, \ldots\ldots, j_k = 0, 1$

Proof

For any vector $e_r \in S_r$ we assert that

$$e_r = \sum_{j_1, \ldots, j_k = 0}^{1} \prod_{i=1}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i}\, e(j_1, j_2, \ldots, j_k)$$

where $\lambda_i = \dfrac{e^r_i - x_i}{y_i - x_i}$

For consider the t^{th} component of the right hand side

$$= \sum_{j_1 \ldots j_k = 0}^{1} \prod_{i=1}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} (x_t^{1-j_t} + y_t^{j_t} - 1)$$

$$= \sum_{j_1 \dots j_k = 0}^{1} \left[\prod_{\substack{i=1 \\ i \neq t}}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{j_i} \right] \lambda_t^{j_t} (1 - \lambda_t)^{1-j_t} (x_t^{1-j_t} + y_t^{j_t} - 1)$$

$$= \sum_{j_1 \dots j_{t-1}\, j_{t+1} \dots j_k = 0}^{1} \left[\prod_{\substack{i=1 \\ i \neq t}}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} \right] \left[(1 - \lambda_t) x_t + \lambda_t y_t \right]$$

$$= (1 - \lambda_t) x_t + \lambda_t y_t \quad \text{(from Lemma 4)}$$

$$= x_t + \frac{(e^r_t - x_t)}{(y_t - x_t)} (y_t - x_t) = e^r_t$$

Let $\theta (j_1, j_2 \dots j_k) = \prod_{i=1}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} \quad j_1, j_2, \dots j_k = 0,1$

For any $e_r \in S_r$ it is clear that $1 \geq \lambda_i \geq 0$ so that

$\theta (j_1, j_2 \dots j_k) \geq 0$

and $\sum_{j_1 \dots j_k = 0}^{1} \theta(j_1, j_2 \dots j_k) = 1$ (from Lemma 4) so that any point $e_r \in S_r$ can be expressed as a convex combination of the extreme points of S_r using the weights

$$\theta(j_1, j_2 \dots j_k) \text{ for extreme points } e(j_1, j_2 \dots j_k)$$

Since z is a concave function of e it follows that for any $e_r \in S_r$

$$z(e_r) \geq \sum_{j_1 \dots j_k = 0}^{1} \theta(j_1, j_2 \dots j_k)\, z(e(j_1, j_2 \dots j_k))$$

For the random variable $\tilde{e}_r$ with p.d.f. $f_r(e_r)$, it follows that

$$E(z(\tilde{e}_r)) \geq \sum_{j_1 \dots j_k = 0}^{1} \bar{\theta}(j_1, j_2 \dots j_k)\, z(e(j_1, j_2, \dots j_k))$$

(Q.E.D.)

Corollary 1*

If the components of $\tilde{e}$ are independently distributed

$$\bar{\theta}(j_1, j_2 \dots j_k) = \sum_{i=1}^{k} \bar{\lambda}_i^{\,j_i} (1 - \bar{\lambda}_i)^{1-j_i} \text{ for } j_1, \dots j_k = 0, 1$$

---

*Madansky[9] derives a similar result.

where $\bar{\lambda}_i = \dfrac{\bar{e}^r_i - x_i}{y_i - x_i}$

Proof

Since $\tilde{e}_i$ are independent it follows that $\tilde{e}^r_i$ are independent. Further λ_i are linear functions of e^r_i, so that λ_i are independent.

$$\therefore \quad \bar{\theta}(j_1, j_2 \ldots j_k) = E\left(\prod_{i=1}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} \right) = \prod_{i=1}^{k} E\left(\lambda_i^{j_i} (1-\lambda_i)^{1-j_i}\right)$$

$$\text{Now } E\left(\lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} \right) = \begin{cases} E(\lambda_i) & \text{when } j_i = 1 \\ 1 - E(\lambda_i) & \text{when } j_i = 0 \end{cases}$$

$$\text{so that } \quad \bar{\theta}(j_1, j_2 \ldots j_k) = \sum_{i=1}^{k} \bar{\lambda}_i^{j_i} (1 - \bar{\lambda}_i)^{1 - j_i}$$

(we note that $\bar{\lambda}_i$ are easy to compute in this case)

(Q.E.D.)

Lemma 4

$$\sum_{j_1 \ldots j_k = 0}^{1} \prod_{i=1}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} = 1 \quad \text{for any } \lambda_i.$$

Proof

We use induction on k.

Clearly for $k = 1$,

$$\lambda_1 + (1 - \lambda_1) = 1$$

Suppose it is true for $(k - 1)$ we shall show it is true for k:

$$\sum_{j_1 \ldots j_k = 0}^{1} \prod_{i=1}^{k} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} = \sum_{j_1 \ldots j_k = 0}^{1} \left[\prod_{i=1}^{k-1} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} \right] \lambda_k^{j_k} (1 - \lambda_k)^{1-j_k}$$

$$= \sum_{j_1 \ldots j_{k-1} = 0}^{1} \left\{ \left[\prod_{i-1}^{k-1} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} \right] (1 - \lambda_k) + \left[\prod_{i-1}^{k-1} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} \right] \lambda_k \right\}$$

$$= \sum_{j_1 \ldots j_k = 0}^{1} \prod_{i=1}^{k-1} \lambda_i^{j_i} (1 - \lambda_i)^{1-j_i} = 1 \text{ by the induction hypothesis.}$$

The proposition is true for all k.

(Q.E.D.)

Let $$LB_r = \sum_{j_1, j_r \ldots j_k = 0}^{1} \bar{\theta}(j_1, j_2 \ldots j_k)\, z\,(e\,(j_1, j_2 \ldots j_k))$$

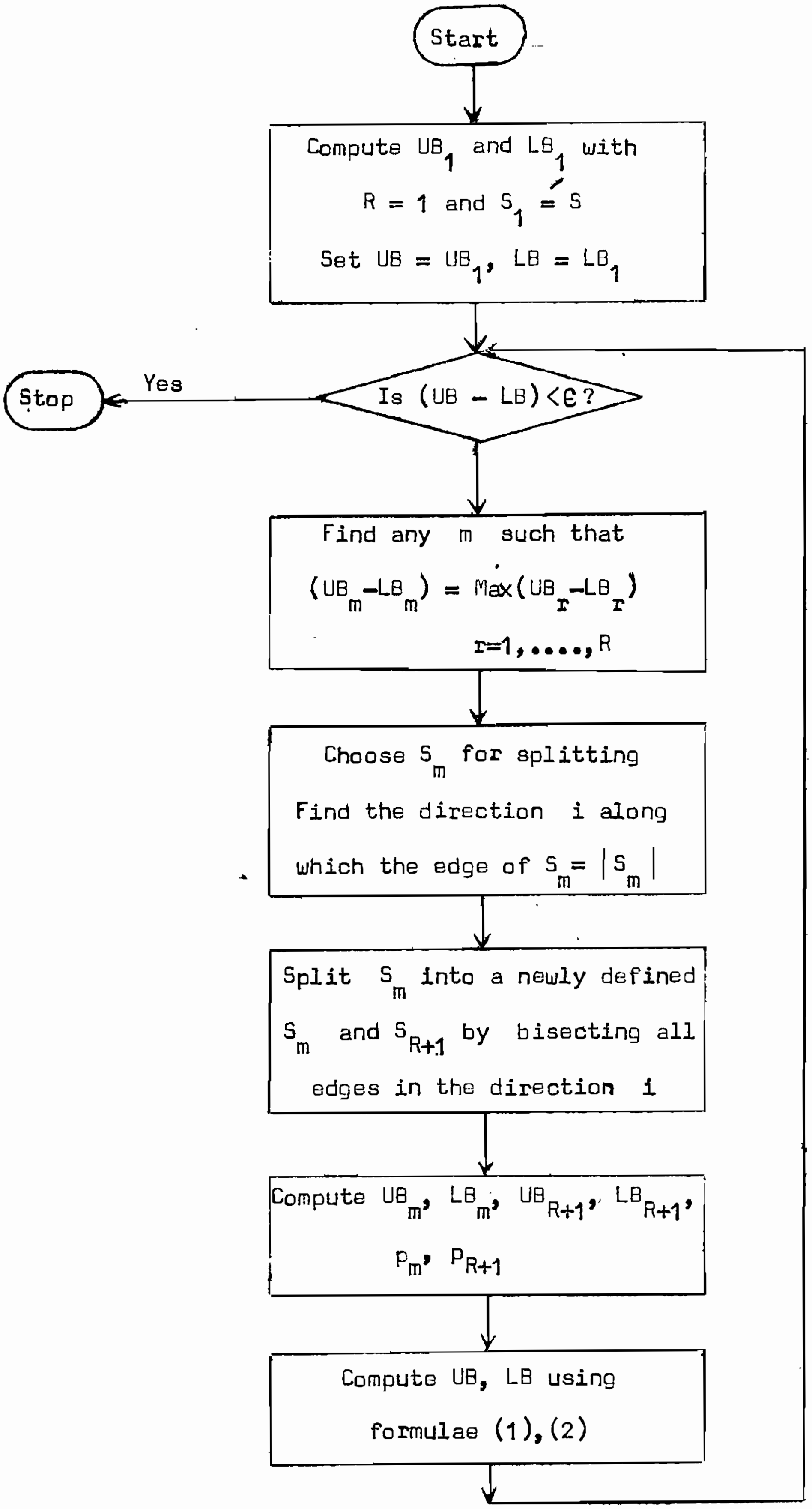
Then from Lemma 1 we obtain the following lower bound (LB)

$$LB = \sum_{r=1}^{R} p_r\, LB_r \leq E\,(z\,(\tilde{e})) \qquad \ldots\ldots (2)$$

We now have upper and lower bounds on $E\,(z\,(\tilde{e}))$

Algorithm

Based on the above bounds we construct an algorithm to estimate $E\,(z\,(\tilde{e}))$ within an interval of preset length $\varepsilon > 0$. This algorithm is flow charted on the next page. (Note $|S_r|$ is defined as the length of the longest edge of box S_r).

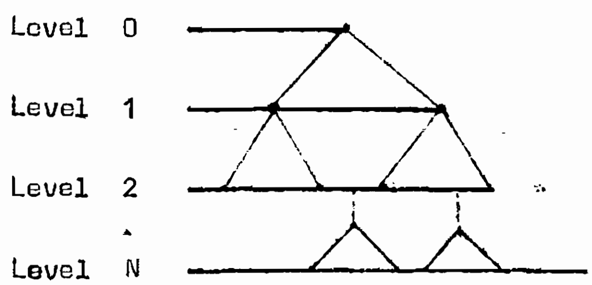


FLOW CHART FOR ALGORITHM

Proof of Finite Convergence of Algorithm

We shall now prove finite convergence for the algorithm.

A convenient way to look at the algorithm is to view it as the growing of a tree. We start with a node corresponding to box S. We split this box into two boxes which can be viewed as two new nodes with branches emanating from S. (See diagram). The current decomposition at any stage is completely specified by those nodes that are of degree 1. These will be called 'growing nodes'.



We shall call a node of degree 1 which has $UB_r - LB_r < \epsilon$ a 'dead' node. All nodes of degree 1 which are not dead will be called 'live'. At any stage of the algorithm we do one of two things:

1. Decide that we do not need to split any of the growing nodes.

2. Pick a growing node and split it into 2 nodes of degree 1.

Theorem 1

The algorithm never chooses a dead node for splitting.

Proof

For suppose a dead node was chosen for splitting. This implies that <u>every</u> node of degree 1 in the tree has $(UB_r - LB_r) < \epsilon$ (since we chose the dead node to split).

This implies $UB - LB = \sum_r p_r \; UB_r - \sum_r p_r \, LB_r < \epsilon$

which implies that we must have stopped the algorithm before we could choose a dead node for splitting.

(Q.E.D.)

Theorem 2

Beyond a certain level N all nodes are 'dead' nodes.

Proof

Since z (e) is a continuous function of e over S and since S is compact, z (e) is uniformly continuous on S.

This implies that there is $\delta > 0$ such that for any

$x, y \in S$ with $|x-y| < \delta$, $|z(x) - z(y)| < \epsilon/2$

Let $L_1, L_2 \ldots L_k$ be the lengths of the edges of box S.

Now find numbers β_i, $i = 1,2,\ldots,k$ such that

$$\frac{L_i}{2^{\beta_i}} < \delta \leq \frac{L_i}{2^{\beta_i - 1}}$$

Let $N = \sum_{i=1}^{k} \beta_i$. Then if we consider a node (corresponding to a box S_r) at level N, S_r must have edge lengths $L_i/2^{\beta_i}$, so that

$$|S_r| < \delta$$

We shall show that for box S_r, $UB_r - LB_r < \epsilon$ and hence S_r corresponds to a 'dead' node.

Since $\bar{e}_r \in S_r$, for any $e_r \in S_r$

$$|\bar{e}_r - e_r| < \delta \rightarrow |z(\bar{e}_r) - z(e_r)| < \epsilon/2 \rightarrow z(\bar{e}_r) - \epsilon/2 < z(e_r) < z(\bar{e}_r) + \epsilon/2$$

Taking expectations using the p.d.f. $f_r(e)$ we have

$$z(\bar{e}_r) - \epsilon/2 < E(z(e_r)) < z(\bar{e}_r) + \epsilon/2$$

$$E(z(e_r)) - \epsilon/2 < z(\bar{e}_r) < E(z(e_r)) + \epsilon/2$$

$$E(z(e_r)) - \epsilon/2 < UB_r < E(z(e_r)) + \epsilon/2 \qquad \ldots. \quad (3)$$

Let $v_1, v_2 \ldots\ldots v_T$ be the vertices of S_r

Since $v_t \in S_r$ for any $e_r \in S_r$

$$| v_t - e_r | < \delta \rightarrow |z(v_t) - z(e_r)| < \epsilon/2 \rightarrow z(v_t) - \epsilon/2 < z(e_r) < z(v_t)$$

$$+ \epsilon/2, \; t = 1,2 \ldots. T$$

Let w_t be arbitrary numbers satisfying $w_t \geq 0, \sum_{t=1}^{T} w_t = 1$

$$\sum_{t=1}^{T} w_t z(v_t) - \epsilon/2 < z(e_r) < \sum_{t=1}^{T} w_t z(v_t) + \epsilon/2$$

Taking expectations

$$\sum_{t=1}^{T} w_t z(v_t) - \epsilon/2 < E(z(e_r)) < \sum_{t=1}^{T} w_t z(v_t) + \epsilon/2$$

$$E(z(e_r)) - \epsilon/2 < \sum_{t=1}^{T} w_t z(v_t) < E(z(e_r)) + \epsilon/2$$

But $LB_r = \sum_{t=1}^{T} w_t \, z(v_t)$ for some system of weights w_t

so that

$$E(z(e_r)) - \epsilon/2 < LB_r < E(z(e_r)) + \epsilon/2 \qquad \ldots\ldots (4)$$

From (3) and (4) it follows that $|UB_r - LB_r| < \epsilon$

From theorem 1, S_r corresponds to a dead node.

(Q.E.D.)

We have shown in Theorem 2 that any node that reaches level N must be a 'dead' node. If every node of degree 1 is at some stage at level N then every node is 'dead' and we have $UB_r - LB_r < \epsilon$ for r = 1,2,....R so that $UB - LB < \epsilon$. We will have therefore achieved our aim in the algorithm after at most 2^N+1 splits. (Since carrying out 2^N+1 splits will reduce all nodes of degree 1 to 'dead' nodes).

An upper bound on the number of iterations through the loop of the flow chart is therefore,

$$2^N+1 \quad \text{where} \quad N = \sum_{i=1}^{k} \beta_i \text{ with } \beta_i \text{ satisfying}$$

$$\frac{L_i}{2^{\beta_i}} < \delta \leq \frac{L_i}{2^{\beta_i - 1}}$$

Note the interesting fact that this upper bound is independent of f(e) which may be any continuous, discrete or mixed probability density/mass function.

Computational Experience with Milk Grid Problem

The algorithm outlined above was programmed in FORTRAN IV to solve the milk grid problem.

The specific model structure is outlined below:

$$z = \text{Max} \sum_{i=1}^{4} \sum_{k=1}^{3} p_i^k v_i^k - \sum_{i=1}^{4} \sum_{j=1}^{4} c_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^{4} x_{ij} \leq \tilde{a}_i \qquad i = 1,2,3,4$$

$$\sum_{i=1}^{4} x_{ij} - \sum_{k=1}^{3} v_j^k = 0 \quad j = 1,2,3,4$$

$$\left.\begin{aligned} v_i^1 &\leq t_i^1 \\ v_i^2 &\leq t_i^2 - t_i^1 \end{aligned}\right\} \quad i = 1,2,3,4$$

$$\sum_{i=1}^{4} \sum_{j=1}^{4} x_{ij} - \sum_{i=1}^{4} x_{ii} \leq k$$

$$x_{ij},\ v_j^k \geq 0 \qquad i = 1,2,3,4$$

$$j = 1,2,3,4$$

$$k = 1,2,3,4$$

Where $\tilde{a}_i$ are random variables reflecting availability in region i, these are truncated normally distributed, independent random variables. The truncation points were $\pm$ 3 standard deviations from the mean.

x_{ij} is the quantity shipped from region i to region j.

v_i^k is the quantity sold in region i in the k^{th} price range.

p_i^k is the unit price in region i in the k^{th} price range.

t_i^1 is the maximum of the first price range in region i.

t_i^2 is the maximum of the second price range in region i.

k is the total inter-regional transport capacity available.

Data Estimation

The average daily availability of milk in the four regions was assumed to be:

Region 1	(Western Region)	...	650 tonnes
Region 2	(Northern Region)	...	500 tonnes
Region 3	(Eastern Region)	...	150 tonnes
Region 4	(Southern Region)	...	150 tonnes

This is roughly in line with the situation prevailing in 1975. The cost of transport was estimated to be approximately 22 p/tonne-mile. This translated into the following inter-regional transport costs (Rs 000's/tonne):

Region 1 – Region 2	(900 miles)	..	0.2
Region 1 – Region 3	(1400 miles)	..	0.3
Region 1 – Region 4	(900 miles)	..	0.2
Region 2 – Region 3	(900 miles)	..	0.2
Region 2 – Region 4	(1400 miles)	..	0.3
Region 3 – Region 4	(900 miles)	..	0.2

This cost covers the <u>variable</u> portion of the transport cost. The fixed portion was not estimated, but different runs were made assuming that there was a total inter-regional capacity = 0.0, 40.0, and 80.0 tonnes/day.

The price-volume relationship in each region was assumed to be the same and estimated from data available for the western region in the model. This relationship was approximated by the following piece-wise linear function.

Quantity (tonnes/day)	Price (Rs 000's/tonne)
0 – 400	2.5
400 – 800	1.5 (incremental price for each tonne above 400)
above 800	1.0 (incremental price for each tonne above 800)

Model Runs

Two sets of runs were made. In the first set of runs, the standard deviations of the available quantities in each region were set to 10 per cent of the average available quantity, whereas in the second set the standard deviations were set to 20 per cent. Each set consisted of three runs with inter-regional transport capacity set at 0, 40 and 80 tonnes/day.

The results are tabulated below:

AVERAGE REVENUE EARNED BY PRODUCERS PER DAY
(Rs 000's)

Inter-regional capacity	0 tonnes/day	40 tonnes/day	80 tonnes/day
Standard Deviation = 10%	3,275	3,347	3,419
Standard Deviation = 20%	3,260	3,342	3,414

The results clearly show that in each set, there is significant improvement in average revenue both in increasing capacity from 0 to 40 and from 40 to 80 tonnes/day. The increase is in excess of Rs 70,000/- for every 40 tonnes of capacity added. This improvement has to be balanced with the capital cost required to provide the capacity which has not (as mentioned earlier) explicitly entered the above analysis.

Conclusion

The algorithm has proved to be effective for the milk-grid problem. Computation times for the runs made on an IBM 360/44 ranged from 3 to 10 minutes. It would also be useful for problems of larger size if the number of stochastic elements is small. The algorithm can be readily extended to provide estimates of variance and higher moments of the objective function as well as the distribution of the objective function. It can also be extended to include risk averse utility functions defined on the objective function values[10].

!!!!!!!!!!!!!

REFERENCES

1 Madansky, A., "Methods of Solution of Linear Programs under Uncertainty", Operations Research, Vol.10 (1962) pp.463-470.

2 Vajda, S., "Probabilistic Programming", Academic Press, New York (1972).

3 Sengupta, JK, G. Tintner and C. Millham, "On Some Theorems of Stochastic Linear Programming with Applications", Management Science, Vol. 10, No.1 (1963).

4 Tintner, G. and M.V. Rama Sastry, "A Note on the Use of Non-parametric Statistics in Stochastic Linear Programming", Management Science, Vol. 19, No.2 (1972).

5 Kolbin, V.V., "Stochastic Programming", in Progress in Mathematics, Vol.11, R.V. Gamkrelidze (ed.) Plenum Press, New York - London (1971).

6 Berneau, "On Stochastic Linear Programming - The LaPlace Transform of the Distribution of the Optimum and Applications", Journal of Math. Analysis & Applications, Vol.15, 280-294 (1966).

7 Mangasarian, Olvi L., "Non-Linear Programming, McGraw-Hill, New York (1969) pp 62-63.

8 Ferguson, T.S., "Mathematical Statistics - A Decision Theoretic Approach", Academic Press, New York (1967) pp 76-78.

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

10 Patel, N.R., "Optimal Control of Sulfur Dioxide Emissions at Power Stations : Models and a Case Study", Ph.D. Thesis, MIT, June 1973, Appendices II and III.