M. N. El Agizy

Dynamic Inventory Models and Stochastic Programming*

Abstract: A wide class of single-product, dynamic inventory problems with convex cost functions and a finite horizon is investigated as a stochastic programming problem. When demands have finite discrete distribution functions, we show that the problem can be substantially reduced in size to a linear program with upper-bounded variables. Moreover, we show that the reduced problem has a network representation; thus network flow theory can be used for solving this class of problems. A consequence of this result is that, if we are dealing with an indivisible commodity, an integer solution of the dynamic inventory problem exists. This approach can be computationally attractive if the demands in different periods are correlated or if ordering cost is a function of demand.

Introduction

The purpose of this paper is twofold, to formulate a wide class of dynamic inventory models as a stochastic programming problem that can be reduced to a linear program with upper-bounded variables and to show that the special structure of the problem allows the use of network flow solution techniques in preference to the simplex method of linear programming. To keep the exposition in terms of one particular model we restrict our initial discussion to the single-commodity, multiperiod inventory model with no back orders and with a finite horizon. Later, attention is turned to other models to which our approach applies.

The model to be considered in detail is described as follows: A decision maker can procure a single item in any of a finite number of time periods. We assume that procurements are accomplished immediately and that the distribution of the demand for the item is known for each period. The decision maker balances the discrepancy between stock on hand and the actual demand either by holding the item in inventory or by emergency purchasing. His economic problem arises because he can expect to accrue savings by buying in one period, while facing the demand uncertainty, and then holding the item in inventory for future periods.

The problem is formulated in the next section as a stochastic program and assumptions are made about the different cost elements. Then we focus attention on the discrete demand-distribution case and formulate the problem as a linear program. Using optimality conditions and additional assumptions, we reduce the linear program to an equivalent one with upper-bounded variables and with significantly fewer equations and variables. We also show that the reduced problem can be solved as a network flow problem. In the last section we show that our approach also applies to the inventory problem in which back orders are allowed.

No attempt has been made to compare the network flow approach with dynamic programming for computational efficiency.

Dynamic inventory model

- *Stochastic program* (*without back orders*)

We now formalize the discussion in the preceding section. Let x_t denote the amount of stock procured at the beginning of period t and let c_t be the associated unit cost. Let b_t be a random variable denoting the total demand for the item during period t, $t = 1, \cdots, k$. To simplify the description we assume that the demand in period t is stochastically independent of the demand in the preceding periods. At the end of each period let the excess of demand over supply be u_t and let the associated unit-shortage cost or emergency-purchase cost be α_t. The excess of supply over demand is denoted by v_t and the associated unit cost by β_t. The variables u_t, v_t and x_{t+1} are functions of $b_1, \cdots, b_t$ and all costs are appropriately discounted.

The inventory problem can be stated mathematically as follows:

$$\min_{x_1, \cdots, x_k} \mathrm{E} \sum_{t=1}^{k} (c_t x_t + \alpha_t u_t + \beta_t v_t), \tag{1}$$

The author is located at IBM Corporate Headquarters, Armonk, New York 10504.

* This work was described at the Institute of Management Science international meeting in Mexico City, Mexico, August 1967.

where E denotes expectation with respect to $b_1, \cdots, b_k$, subject to

$$b_t = v_{t-1} + x_t + u_t - v_t,$$

$$x_t \geq 0, \quad u_t \geq 0, \quad v_t \geq 0 \quad \text{and} \quad u_t v_t = 0,$$

where $t = 1, \cdots, k$ and v_0 is prespecified (e.g., $v_0 = 0$).

The problem is characterized by a sequence of ordering decisions over a horizon of k periods. At the beginning of each period t, we start with an initial stock v_{t-1} and we are allowed to order additional stock x_t. Orders are assumed to be filled immediately. When a random event occurs that specifies a particular value of b_t, the total demand during period t, immediate action is taken to compensate for imbalance between supply and demand. The excess $v_t = [(v_{t-1} + x_t) - b_t] \geq 0$ of supply over demand is transferred to the next period; however, the shortage $u_t = [b_t - (v_{t-1} + x_t)] \geq 0$ cannot be backlogged. The problem for the decision maker is to choose sequentially for each period the optimal order quantity that will minimize his expected cost.

Frequently, in practice, there will be no penalty cost associated with the storage activity, so that $\beta_t = 0$ for $t = 1, \cdots, k$. In fact, it is mathematically convenient to reduce the general problem to one with $\beta_t = 0$. To do this we substitute

$$v_t = v_{t-1} + x_t + u_t - b_t$$

(in the order $t = k, k-1, \cdots, 1$) in the functional (1). The coefficients c_t, α_t and β_t in (1) are replaced, respectively, by $c_t + \sum_{h=t}^{k} \beta_h$, $\alpha_t + \sum_{h=t}^{k} \beta_h$ and 0. Now we can impose the assumption that $\beta_t = 0$ with no loss of generality in the remainder of the paper.

In most real problems it is reasonable to expect that the following assumptions will be satisfied:

1. $\alpha_t > c_t > 0$.

2. $\alpha_t > c_{t+1}$.

The first assumption states that for any period t the shortage cost is greater than the procurement cost. If this assumption were not satisfied, the optimal policy would be to make no orders and to pay the cost associated with the shortage. We require that the sum of the procurement cost in period t and the discounted holding costs in periods $t, t+1, \cdots, k$ be positive. The problem would become trivial if this condition were violated. Assumption 2 stipulates that the emergency purchase cost plus the holding cost in one period is greater than the ordering cost in the next period. These mild assumptions guarantee that an optimal solution exists and that u_t and v_t are not both positive at optimality.

Dynamic programming has been used very elegantly to solve the multiperiod inventory problem[1–4] by the application of the principle of optimality. It is computationally attractive when the current demand is independent of the demand during the preceding periods and the cost elements are independent of the realized demand. However, if the values of demand in different periods are correlated or if the costs are dependent on the demand, the computational requirement of dynamic programming becomes enormous.

• *Deterministic-equivalent linear program*

Concerning ourselves now with finite discrete values of demand, we show how the problem can be solved using network flow techniques. We assume that the demand in period t is statistically independent of the demand in preceding periods, that the number of possible b_t is the same for all t and that the procurement cost is independent of demand. It will be clear later that our analysis would still be valid if these assumptions were waived.

Our objectives are (a) to consider the deterministic linear program equivalent to the program (1) and to state (without proof) necessary and sufficient conditions for optimality of the latter program; (b) to use optimality conditions and assumptions 1 and 2 to reduce the deterministic program to an equivalent linear program with upper-bounded variables and with considerably fewer equations and variables; and (c) to show that the reduced problem has a network representation.

When the random variable b_t has a finite discrete distribution we can denote the possible outcomes by b_{tl_t} (where $l_t = 1, \cdots, L$) and the corresponding probabilities by p_{tl_t} (where $\sum p_{tl_t} = 1$). (To simplify the notation, from this point we suppress the subscript on l when l itself is a subscript.) In program (1) the variable x_t is a function of the sequence of random variables $b_1, \cdots, b_{t-1}$ or is, equivalently, a function of the sequence of indices $l_1, \cdots, l_{t-1}$. Also, the variables u_t and v_t are functions of the sequence of indices $l_1, \cdots, l_t$. For each possible sequence of indices we define the corresponding variables $x_t(\mathbf{1}_{t-1})$, $u_t(\mathbf{1}_t)$ and $v_t(\mathbf{1}_t)$, where $\mathbf{1}_t$ denotes the vector $(l_1, \cdots, l_t)$. For any variable, e.g., u_t, we use the notation $u_t(\mathbf{1}_{t-1}, l_t)$ or $u_t(\mathbf{1}_{t-1}, l)$ interchangeably with $u_t(\mathbf{1}_t)$ for convenience.

Following Dantzig's approach[5–7] we write the program (1) as a linear program with explicit constraints corresponding to each possible value of the random vector $(b_1, \cdots, b_k)$ as follows:

$$\min \left\{ \sum_{t=1}^{k} \sum_{l=1}^{L} p_{1l} \cdots \sum_{l=1}^{L} p_{t-1,l} \times \left[c_t x_t(\mathbf{1}_{t-1}) + \sum_{l=1}^{L} p_{tl} \alpha_t u_t(\mathbf{1}_t) \right] \right\} \tag{2}$$

subject to

$$b_{tl} = v_{t-1}(\mathbf{1}_{t-1}) + x_t(\mathbf{1}_{t-1}) + u_t(\mathbf{1}_t) - v_t(\mathbf{1}_t),$$

$x_t(1_{t-1}) \geq 0, \quad u_t(1_t) \geq 0, \quad v_t(1_t) \geq 0 \quad$ and

$u_t(1_t)v_t(1_t) = 0.$

Proposition 1
Necessary and sufficient conditions for solvability of program (2) are that $c_t \geq 0$ and $\alpha_t \geq 0$ for $t = 1, 2, \cdots, k$.

Proposition 2
A sufficient condition that some optimal solution of program (2) satisfies $u_t(1_t)v_t(1_t) = 0$ for all 1_t is that $c_t \geq 0, \alpha_t \geq 0$ and $\alpha_t \geq \min(\alpha_{t+1}, c_{t+1})$ for $t = 1, \cdots, k$. [This is a relation among cost elements which ensures that program (2) satisfies $u_t(1_t)v_t(1_t) = 0$ at optimality.]

Assume for each t in (2) that the b_{tl} are ordered as increasing in l. In an optimal solution of program (2), if $u_t(1_{t-1}, r)$ is positive for $1 \leq r \leq L$, then

$$u_t(1_{t-1}, L) > u_t(1_{t-1}, L-1) > \cdots > u_t(1_{t-1}, r) > 0$$

and

$$v_t(1_{t-1}, L) = v_t(1_{t-1}, L-1) = \cdots = v_t(1_{t-1}, r) = 0.$$

Similarly, if $v_t(1_{t-1}, r)$ is positive, then

$$v_t(1_{t-1}, 1) > v_t(1_{t-1}, 2) > \cdots > v_t(1_{t-1}, r) > 0 \quad \text{and}$$

$$u_t(1_{t-1}, 1) = u_t(1_{t-1}, 2) = \cdots = u_t(1_{t-1}, r) = 0.$$

In other words, an optimal solution of program (2) must satisfy

$$\begin{aligned} &u_t(1_{t-1}, l) \geq u_t(1_{t-1}, l-1), \\ &v_t(1_{t-1}, l-1) \geq v_t(1_{t-1}, l), \\ &u_t(1_t)v_t(1_t) = 0, \\ &u_t(1_t) \geq 0 \quad \text{and} \quad v_t(1_t) \geq 0. \end{aligned} \tag{3}$$

Let b_0 be an integer satisfying

$$b_0 < \min_t (b_{t1}, 0). \tag{4}$$

From the tth period constraints on (2) we have

$$\begin{aligned} b_{tl} &= v_{t-1}(1_{t-1}) + x_t(1_{t-1}) + u_t(1_{t-1}, l) - v_t(1_{t-1}, l), \\ &\qquad l = 1, \cdots, L, \quad \text{and} \\ b_{tl} - b_{t,l-1} &= -u_t(1_{t-1}, l-1) + v_t(1_{t-1}, l-1) \\ &\qquad + u_t(1_{t-1}, l) - v_t(1_{t-1}, l), \\ &\qquad l = 2, \cdots, L. \end{aligned} \tag{5}$$

Observing that

$$v_{t-1}(1_{t-1}) + x_t(1_{t-1}) \geq 0, \tag{6}$$

from Eqs. (3), (4) and (5) we conclude that any optimal solution satisfies

$$\begin{aligned} &b_{tl} - b_0 > u_t(1_{t-1}) \geq 0, \quad l = 1, \cdots, L, \quad \text{and} \\ &b_{tl} - b_{t,l-1} \geq u_t(1_{t-1}, l) - u_t(1_{t-1}, l-1) \geq 0, \\ &\qquad l = 2, \cdots, L. \end{aligned} \tag{7}$$

For each t define a new set of variables,

$$\begin{aligned} y_t(1_{t-1}, 1) &= b'_{t1} - u_t(1_{t-1}, 1), \\ y_t(1_{t-1}, l) &= b'_{tl} + u_t(1_{t-1}, l-1) - u_t(1_{t-1}, l) \\ &= v_t(1_{t-1}, l-1) - v_t(1_{t-1}, l) \quad \text{and} \\ y_t(1_{t-1}, L+1) &= v_t(1_{t-1}, L), \; l = 2, \cdots, L, \end{aligned} \tag{8}$$

where

$$b'_{t1} = b_{t1} - b_0 \quad \text{and}$$

$$b'_{tl} = b_{tl} - b_{t,l-1}, \qquad l = 2, \cdots, L.$$

Then it follows that

$$\begin{aligned} u_t(1_{t-1}, l) &= b_{tl} - b_0 - \sum_{r=1}^{l} y_t(1_{t-1}, r), \\ v_t(1_{t-1}, l) &= \sum_{r=l+1}^{L+1} y_t(1_{t-1}, r) \quad \text{and} \\ u_t(1_{t-1}, l) - v_t(1_{t-1}, l) &= b_{tl} - b_0 - \sum_{l=1}^{L+1} y_t(1_{t-1}, l). \end{aligned} \tag{9}$$

From Eqs. (3), (7) and (8) we obtain

$$\begin{aligned} &b'_{tl} \geq y_t(1_{t-1}, l) \geq 0, \\ &y_t(1_{t-1}, L+1) \geq 0 \end{aligned} \tag{10}$$

and the conditions

$$y_t(1_{t-1}, 1) > 0, \tag{11a}$$

$$y_t(1_{t-1}, l-1) = b'_{t,l-1} \quad \text{if} \quad y_t(1_{t-1}, l) > 0 \quad \text{for} \quad l \neq 1 \quad \text{and} \tag{11b}$$

$$y_t(1_{t-1}, l+1) = 0 \quad \text{if} \quad b'_{tl} > y_t(1_{t-1}, l) \quad \text{for} \quad l \neq L+1. \tag{11c}$$

The derivation of Eqs. (11) can be shown as follows:

11a. This result is an immediate consequence of relations (7) and (8).

11b. If $y_t(1_{t-1}, l) > 0$, then $v_t(1_{t-1}, l-1) > 0$ from (8) and it follows from (3) that $u_t(1_{t-1}, l-1) = u_t(1_{t-1}, l-2) = 0$. Again using Eq. (8) we find $y_t(1_{t-1}, l-1) = b'_{t,l-1}$.

11c. If $b'_{tl} > y_t(1_{t-1}, l)$, then $u_t(1_{t-1}, l) > 0$ from (8) and it follows from (3) that $v_t(1_{t-1}, l) = v_t(1_{t-1}, l+1) = 0$. Using (8) we also obtain $y_t(1_{t-1}, l+1) = 0$.

Substituting $u_t(1_{t-1}, l)$ and $v_t(1_{t-1}, l)$ in terms of $y_t(1_{t-1}, l)$ in program (2), we reformulate the inventory problem as the linear program

$$\min \left\langle \left\{ c_1 x_1 - \alpha_1 \sum_{l=1}^{L+1} \delta_{1l} y_1(l) + \sum_{l=1}^{L} p_{1l} \left[c_2 x_2(1_1) - \alpha_2 \sum_{l=1}^{L+1} \delta_{2l} y_2(1_1, l) \right] + \cdots + \sum_{l=1}^{L} p_{1l} \cdots \sum_{l=1}^{L} p_{k-1,l} \left[c_k x_k(1_{k-1}) - \alpha_k \sum_{l=1}^{L+1} \delta_{kl} y_k(1_{k-1}, l) \right] \right\} \right\rangle - b_0 \sum_{t=1}^{k} \alpha_t, \tag{12}$$

subject to the conditions

$$b_0 = x_1 - \sum_{l=1}^{L+1} y_1(l)$$

$$b_0 = \sum_{r=l+1}^{L+1} y_1(r) + x_2(1_1) - \sum_{l=1}^{L+1} y_2(1_1, l)$$

$$\vdots \qquad \vdots$$

$$b_0 = \sum_{r=l+1}^{L+1} y_{k-1}(1_{k-2}, r) + x_k(1_{k-1}) - \sum_{l=1}^{L+1} y_k(1_{k-1}, l),$$

$$x_1 \geq 0, \cdots, x_k(1_{k-1}) \geq 0,$$

$$b'_{1l} \geq y_1(l) \geq 0, \cdots, b'_{kl} \geq y_k(1_{k-1}, l) \geq 0 \quad \text{and}$$

$$y_1(L+1) \geq 0, \cdots, y_k(1_{k-1}, L+1) \geq 0,$$

where, using P to denote probability, we have

$$\delta_{tl} = \sum_{h=l}^{L} p_{th} = \mathrm{P}[b_t \geq b_{tl}], \; l = 1, \cdots, L;$$

$$\delta_{t,L+1} = 0;$$

$$b'_{t1} = b_{t1} - b_0; \quad \text{and}$$

$$b'_{tl} = b_{tl} - b_{t,l-1}, \qquad l = 2, \cdots, L.$$

Proposition 3
An optimal solution of program (12) is also a solution of program (2) and conversely.

Proof
There is a one-to-one correspondence between feasible solutions of programs (2) and (12) if we impose conditions (11) on the latter program. Moreover, the values of the two objective functions coincide under the correspondence. On the other hand, the hypotheses of Propositions 1 and 2 immediately imply that an optimal solution to (12) satisfies conditions (11).

Proposition 4
Program (12) is equivalent to a directed network.

Proof
Identify each equation in (12) by the label $(t : 1_{t-1})$. This labeling leads to a matrix representation of the linear program as shown in Table 1. For time period 2 subtract the equations labeled $(2 : l+1)$, $(3 : l+1, 1)$, $\cdots$, $(k : l+1, 1, \cdots, 1)$ from the equation labeled $(2 : l)$. In general, for period $t = 2, \cdots, k$ and for $l = 1, \cdots, L-1$ subtract the equations labeled $(t : 1_{t-2}, l+1)$, $(t+1 : 1_{t-2}, l+1, 1)$, $\cdots$, $(k: 1_{t-2}, l+1, 1, \cdots, 1)$ from the equation labeled $(t : 1_{t-2}, l)$. The resulting equivalent program has the property that each column contains at most two nonzero entries (+1 or −1 or both), which characterizes a network node-arc incidence matrix. The network matrix representation of the dynamic inventory problem obtained from the linear program matrix (Table 1) is given in Table 2; the steps accomplishing this transformation are listed in the Appendix.

Proposition 4 allows us to use network flow labeling techniques in solving inventory problems. As an immediate consequence of this proposition, if we are dealing with an indivisible commodity, an integral solution to the dynamic inventory problem exists. Veinott[8] has shown that a nonsingular linear transformation exists that reduces the constraint matrix of program (2) to a transportation type matrix. In his work, contrary to our approach, no reduction in the number of equations or variables is made.

• *Stochastic program (back orders allowed)*
When a stock shortage can be backlogged, the original problem description applies with the exception that the shortage $u_t = [b_t - (v_{t-1} + x_t)] \geq 0$ will be transferred to the next period and satisfied by the later order quantity. The problem is again stated as a stochastic program,

$$\min_{x_1, \cdots, x_k} \mathrm{E} \sum_{t=1}^{k} (c_t x_t + \alpha_t u_t + \beta_t v_t) \tag{13}$$

subject to

$$b_t = -u_{t-1} + v_{t-1} + x_t + u_t - v_t,$$

$$x_t \geq 0, \quad u_t \geq 0, \quad v_t \geq 0 \quad \text{and} \quad u_t v_t = 0,$$

where $t = 1, \cdots, k$ and $(v_0 - u_0)$ is prespecified [e.g., $(v_0 - u_0) = 0$]. Here the coefficients c_t, α_t and β_t can be replaced by $c_t + \sum_{h=t}^{k} \beta_h$, $\alpha_t + \beta_t$ and 0, respectively, without loss of generality. The modified assumptions appropriate to this case are

1. $c_t > 0$ and
2. $\alpha_t > 0$.

When the demand function has a finite discrete distribution, the deterministic-equivalent linear program with upper bounded variables for this problem is

Table 1 Linear program matrix representation of the dynamic inventory problem.

Row label \ Column label	x_1	$y_1(1)$	$y_1(2)$	$y_1(3)$	$y_1(4)$	$x_2(1)$	$y_2(1,1)$	$y_2(1,2)$	$y_2(1,3)$	$y_2(1,4)$	$x_2(2)$	$y_2(2,1)$	$y_2(2,2)$	$y_2(2,3)$	$y_2(2,4)$	$x_2(3)$	$y_2(3,1)$	$y_2(3,2)$	$y_2(3,3)$	$y_2(3,4)$
(1)	1	-1	-1	-1	-1															
(2:1)			1	1	1	1	-1	-1	-1	-1										
(2:2)				1	1						1	-1	-1	-1	-1					
(2:3)					1											1	-1	-1	-1	-1
(3:1,1)								1	1	1										
(3:1,2)									1	1										
(3:1,3)										1										
(3:2,1)													1	1	1					
(3:2,2)														1	1					
(3:2,3)															1					
(3:3,1)																		1	1	1
(3:3,2)																			1	1
(3:3,3)																				1

Row label \ Column label	$x_3(1,1)$	$y_3(1,1,1)$	$y_3(1,1,2)$	$y_3(1,1,3)$	$y_3(1,1,4)$	$y_3(1,1,5)$	$x_3(1,2)$	$y_3(1,2,1)$	$y_3(1,2,2)$	$y_3(1,2,3)$	$y_3(1,2,4)$	$y_3(1,2,5)$	$x_3(1,3)$	$y_3(1,3,1)$	$y_3(1,3,2)$	$y_3(1,3,3)$	$y_3(1,3,4)$	$y_3(1,3,5)$
(1)																		
(2:1)																		
(2:2)																		
(2:3)																		
(3:1,1)	1	-1	-1	-1	-1	-1												
(3:1,2)							1	-1	-1	-1	-1	-1						
(3:1,3)													1	-1	-1	-1	-1	-1
(3:2,1)																		
(3:2,2)																		
(3:2,3)																		
(3:3,1)																		
(3:3,2)																		
(3:3,3)																		

Row label \ Column label	$x_3(2,1)$	$y_3(2,1,1)$	$y_3(2,1,2)$	$y_3(2,1,3)$	$y_3(2,1,4)$	$y_3(2,1,5)$	$x_3(2,2)$	$y_3(2,2,1)$	$y_3(2,2,2)$	$y_3(2,2,3)$	$y_3(2,2,4)$	$y_3(2,2,5)$	$x_3(2,3)$	$y_3(2,3,1)$	$y_3(2,3,2)$	$y_3(2,3,3)$	$y_3(2,3,4)$	$y_3(2,3,5)$
(1)																		
(2:1)																		
(2:2)																		
(2:3)																		
(3:1,1)																		
(3:1,2)																		
(3:1,3)																		
(3:2,1)	1	-1	-1	-1	-1	-1												
(3:2,2)							1	-1	-1	-1	-1	-1						
(3:2,3)													1	-1	-1	-1	-1	-1
(3:3,1)																		
(3:3,2)																		
(3:3,3)																		

Row label \ Column label	$x_3(3,1)$	$y_3(3,1,1)$	$y_3(3,1,2)$	$y_3(3,1,3)$	$y_3(3,1,4)$	$y_3(3,1,5)$	$x_3(3,2)$	$y_3(3,2,1)$	$y_3(3,2,2)$	$y_3(3,2,3)$	$y_3(3,2,4)$	$y_3(3,2,5)$	$x_3(3,3)$	$y_3(3,3,1)$	$y_3(3,3,2)$	$y_3(3,3,3)$	$y_3(3,3,4)$	$y_3(3,3,5)$
(1)																		
(2:1)																		
(2:2)																		
(2:3)																		
(3:1,1)																		
(3:1,2)																		
(3:1,3)																		
(3:2,1)																		
(3:2,2)																		
(3:2,3)																		
(3:3,1)	1	-1	-1	-1	-1	-1												
(3:3,2)							1	-1	-1	-1	-1	-1						
(3:3,3)													1	-1	-1	-1	-1	-1

Table 2 Network matrix representation of the dynamic inventory problem.

Row label \ Column label	x_1	$y_1(1)$	$y_1(2)$	$y_1(3)$	$y_1(4)$	$x_2(1)$	$y_2(1,1)$	$y_2(1,2)$	$y_2(1,3)$	$y_2(1,4)$	$x_2(2)$	$y_2(2,1)$	$y_2(2,2)$	$y_2(2,3)$	$y_2(2,4)$	$x_2(3)$	$y_2(3,1)$	$y_2(3,2)$	$y_2(3,3)$	$y_2(3,4)$
(1)	1	-1	-1	-1	-1															
(2:1)			1			1	-1	-1	-1	-1	-1	1								
(2:2)				1							1	-1	-1	-1	-1	-1	1			
(2:3)					1											1	-1	-1	-1	-1
(3:1,1)								1												
(3:1,2)									1											
(3:1,3)										1										
(3:2,1)													1							
(3:2,2)														1						
(3:2,3)															1					
(3:3,1)																		1		
(3:3,2)																			1	
(3:3,3)																				1

Row label \ Column label	$x_3(1,1)$	$y_3(1,1,1)$	$y_3(1,1,2)$	$y_3(1,1,3)$	$y_3(1,1,4)$	$y_3(1,1,5)$	$x_3(1,2)$	$y_3(1,2,1)$	$y_3(1,2,2)$	$y_3(1,2,3)$	$y_3(1,2,4)$	$y_3(1,2,5)$	$x_3(1,3)$	$y_3(1,3,1)$	$y_3(1,3,2)$	$y_3(1,3,3)$	$y_3(1,3,4)$	$y_3(1,3,5)$
(1)																		
(2:1)																		
(2:2)																		
(2:3)																		
(3:1,1)	1	-1	-1	-1	-1	-1	-1	1	1	1	1	1						
(3:1,2)							1	-1	-1	-1	-1	-1	-1	1	1	1	1	1
(3:1,3)													1	-1	-1	-1	-1	-1
(3:2,1)																		
(3:2,2)																		
(3:2,3)																		
(3:3,1)																		
(3:3,2)																		
(3:3,3)																		

Row label \ Column label	$x_3(2,1)$	$y_3(2,1,1)$	$y_3(2,1,2)$	$y_3(2,1,3)$	$y_3(2,1,4)$	$y_3(2,1,5)$	$x_3(2,2)$	$y_3(2,2,1)$	$y_3(2,2,2)$	$y_3(2,2,3)$	$y_3(2,2,4)$	$y_3(2,2,5)$	$x_3(2,3)$	$y_3(2,3,1)$	$y_3(2,3,2)$	$y_3(2,3,3)$	$y_3(2,3,4)$	$y_3(2,3,5)$
(1)																		
(2:1)	-1	1	1	1	1	1												
(2:2)																		
(2:3)																		
(3:1,1)																		
(3:1,2)																		
(3:1,3)																		
(3:2,1)	1	-1	-1	-1	-1	-1	-1	1	1	1	1	1						
(3:2,2)							1	-1	-1	-1	-1	-1	-1	1	1	1	1	1
(3:2,3)													1	-1	-1	-1	-1	-1
(3:3,1)																		
(3:3,2)																		
(3:3,3)																		

Row label \ Column label	$x_3(3,1)$	$y_3(3,1,1)$	$y_3(3,1,2)$	$y_3(3,1,3)$	$y_3(3,1,4)$	$y_3(3,1,5)$	$x_3(3,2)$	$y_3(3,2,1)$	$y_3(3,2,2)$	$y_3(3,2,3)$	$y_3(3,2,4)$	$y_3(3,2,5)$	$x_3(3,3)$	$y_3(3,3,1)$	$y_3(3,3,2)$	$y_3(3,3,3)$	$y_3(3,3,4)$	$y_3(3,3,5)$
(1)																		
(2:1)																		
(2:2)	-1	1	1	1	1	1												
(2:3)																		
(3:1,1)																		
(3:1,2)																		
(3:1,3)																		
(3:2,1)																		
(3:2,2)																		
(3:2,3)																		
(3:3,1)	1	-1	-1	-1	-1	-1	-1	1	1	1	1	1						
(3:3,2)							1	-1	-1	-1	-1	-1	-1	1	1	1	1	1
(3:3,3)													1	-1	-1	-1	-1	-1

$$\min \left\langle \left\{ c_1 x_1 - \alpha_1 \sum_{l=1}^{L} \delta_{1l} y_1(l) + \sum_{l=1}^{L} p_{1l} \left[c_2 x_2(\mathbf{1}_1) - \alpha_2 \sum_{l=1}^{L} \delta_{2l} y_2(\mathbf{1}_1, l) \right] + \cdots + \sum_{l=1}^{L} p_{1l} \cdots \sum_{l=1}^{L} p_{k-1,l} \left[c_k x_k(\mathbf{1}_{k-1}) - \alpha_k \sum_{l=1}^{L} \delta_{kl} y_k(\mathbf{1}_{k-1}, l) \right] \right\} \right\rangle - b_0 \sum_{t=1}^{k} t\alpha_t, \quad (14)$$

subject to

$$b_0 = x_1 - \sum_{l=1}^{L+1} y_1(l)$$

$$b_0 = \sum_{l=1}^{L+1} y_1(l) + x_2(\mathbf{1}_1) - \sum_{l=1}^{L+1} y_2(\mathbf{1}_1, l)$$

$$\vdots \qquad \vdots$$

$$b_0 = \sum_{l=1}^{L+1} y_{k-1}(\mathbf{1}_{k-2}, l) + x_k(\mathbf{1}_{k-1}) - \sum_{l=1}^{L+1} y_k(\mathbf{1}_{k-1}, l);$$

$$x_1 \geq 0, \cdots, x_k(\mathbf{1}_{k-1}) \geq 0;$$

$$b'_{11} \geq y_1(1), \cdots, b'_{k1} + \sum_{t=1}^{k-1} b'_{tl} \geq y_k(\mathbf{1}_{k-1}, 1) \geq 0;$$

$$b'_{1l} \geq y_1(l) \geq 0, \cdots, b'_{kl} \geq y_k(\mathbf{1}_{k-1}, l) \geq 0,$$

$$l = 2, \cdots, L; \quad \text{and}$$

$$y_1(L+1) \geq 0, \cdots, y_k(\mathbf{1}_{k-1}, L+1) \geq 0;$$

where

$$\delta_{tl} = \sum_{h=l}^{L} p_{th}, \quad l = 1, \cdots, L;$$

$$\delta_{t,L+1} = 0;$$

$$b'_{t1} = b_{t1} - b_0; \quad \text{and}$$

$$b'_{tl} = b_{tl} - b_{t,l-1}, \quad l = 2, \cdots, L.$$

This program can also be reduced to a directed network.

At this point, it should be clear that the approach developed applies without assumptions about the statistical independence of demands, the equal number of possible outcomes for all time periods or the independence of the procurement cost of the demand.

Summary

We have shown that the stochastic, one-product, dynamic inventory problem has a structure that can be reformulated as a network flow problem and thus be made amenable to efficient computational procedures. (Our formulation is a special case of the general class of multiperiod stochastic programming problems investigated in Ref. 9.) In addition to the back-order problem discussed, the network formulation is applicable to other inventory situations, e.g., the case in which there is a time lag in delivery of orders. Here network techniques are preferable to dynamic programming because of the large number of state variables involved. In general, the approach is to identify and exploit the special structure of a stochastic programming problem to achieve a means of practical computation; it is not specifically limited to inventory problems.

Appendix

The transformation from the linear program matrix (Table 1) to the network matrix (Table 2) is accomplished by the following sequential subtraction of matrix rows:

1. (2:1) − (2:2) − (3:2, 1),
2. (2:2) − (2:3) − (3:3, 1),
3. (3:1, 1) − (3:1, 2),
4. (3:1, 2) − (3:1, 3),
5. (3:2, 1) − (3:2, 2),
6. (3:2, 2) − (3:2, 3),
7. (3:3, 1) − (3:3, 2) and
8. (3:3, 2) − (3:3, 3).

References

1. K. J. Arrow, S. Karlin and H. Scarf, *Studies in the Mathematical Theory of Inventory and Production,* Stanford University Press, Stanford, California 1958.
2. R. Bellman, I. Glicksberg and O. Gross, "On the Optimal Inventory Equation," *Management Sci.* **2,** 83 (1955).
3. R. Bellman, *Dynamic Programming,* Princeton University Press, Princeton, New Jersey 1957.
4. A. F. Veinott, Jr., "Optimal Policy in a Dynamic, Single-Product, Non-Stationary Inventory Model with Several Demand Classes," *Operations Res.* **13,** 761 (1965).
5. G. B. Dantzig, "Linear Programming Under Uncertainty," *Management Sci.* **1,** 197 (1955).
6. G. B. Dantzig, *Linear Programming and Extensions,* Princeton University Press, Princeton, New Jersey 1963.
7. M. N. El Agizy, "Two-Stage Programming Under Uncertainty with Discrete Distribution Function," *Operations Res.* **15,** 55 (1967).
8. A. F. Veinott, Jr., unpublished work.
9. M. N. El Agizy, "Multi-Stage Programming under Uncertainty," Ph.D. Thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley 1965.

Received January 24, 1969