VARIATIONS IN OPTIMIZING SERIAL MULTI-STAGE PRODUCTION/INVENTORY SYSTEMS

By

ANDREW Z. SZENDROVITS, PH.D.
Professor of Production and Management Science

GEORGE O. WESOLOWSKY, PH.D.
Associate Professor of Management Science

APPEARED IN THE PROCEEDINGS OF THE CONFERENCE ON THE DISAGGREGATION PROBLEM, OHIO STATE UNIVERSITY, COLUMBUS, OHIO. MARCH, 1977.

FACULTY OF BUSINESS
McMASTER UNIVERSITY
HAMILTON, ONTARIO

Research and Working Paper Series No. 126

March, 1977

VARIATIONS IN OPTIMIZING SERIAL MULTI-STAGE PRODUCTION/INVENTORY SYSTEMS

Andrew Z. Szendrovits, McMaster University
George O. Wesolowsky, McMaster University

ABSTRACT

This paper presents two basic, deterministic, infinite horizon models for a serial multi-stage production/inventory system. One model assumes that the lot size is maintained through all production stages while transportation of equal sized sub-batches is allowed between stages. Consequently, only one set-up cost is incurred for each stage; however, the transportation cost of sub-batches must also be considered. An optimal solution is presented for this first model. The other model has varying lot sizes with lot-sized intershipments assuming certain integrality restrictions on the splitting of lots and monotonically increasing stage inventory holding costs. A good approximation to the optimal solution is given for this second model. Comparison of the two models provides an insight into the characteristics of multi-stage production systems. The efficiency of the computational procedures described is demonstrated with extensive computational experience.

INTRODUCTION

Deterministic, infinite horizon models for serial multi-stage production/inventory systems are well known in the literature; a general survey is given by Clark [1]. The continuous production case, but with continuous intershipment, has been treated previously by Jensen and Khan [3]. Lot size intershipment as well as back-logging were built into the model by Taha and Skeith [8] who used complete enumeration and some limiting assumptions to arrive at a solution. Schwarz and Schrage [4] presented a similar model, but with "echelon" inventory holding costs, the definition of which involves some difficulties in applying the proper holding costs, and may lead to certain anomalies as shown by Szendrovits [7]. Crowston, Wagner and Williams [2] offered a dynamic programming procedure to optimize a model for multi-stage assembly systems where production at each stage occurs instantaneously. Most of these models assume that the lot size at each facility is a positive integer multiple of the lot size at its successor facility. They also assume unconstrained capacity at all stages and do not allow the inventory unit holding cost to be less at a following stage. Transhipment costs are implied to be sunk cost or are built into the fixed cost per lot. Optimization procedures are illustrated by examples for cases with a small number of stages ($n < 10$).

An alternative to lot-splitting is to maintain the lot size through all production stages while allowing transhipment of equal-sized sub-batches between stages. A model for such a system, where the number of sub-batches is predetermined, is offered by Szendrovits [5]. It assumes a uniform (average) unit holding cost for the process inventory at all stages. It also assumes a single production facility at each stage and sunk transportation costs. When the transportation cost for the sub-batches is known, the possibility of optimizing sub-batch sizes is discussed in [6].

The two basic production/inventory models presented in this paper have n manufacturing stages in sequence. These stages are numbered in reverse; that is, the final stage, the one which meets the demand for the finished product is stage 1. Deterministic demand rates over an infinite horizon and unconstrained capacity at all stages are assumed in both models. It is further assumed that at a given stage, the inventory holding cost per unit (the holding cost of units with that stage completed) as well as the set-up cost (of single or multiple facilities), and the production rate are all constant.

Apart from these common characteristics the two models are distinctly different.

Model I incorporates the following assumptions:

a) the lot size is uniform at all stages and set-up cost is incurred only once at each stage;

b) equal sized sub-batches can be transported from a given stage to the next production stage before the production of the lot is finished;

c) the transportation cost of sub-batches between stages is known and constant;

d) the stage holding cost per unit at any stage may be greater than that at the next stage;

e) both the number of sub-batches per lot and the sub-batch size must be an integer.

Model II encompasses the following assumptions:

a) the lot size of a stage is an integer multiple of the lot size of the stage that follows it (this is done for the conventional reason of making the problem analytically tractable);

b) the product is transferred between stages in lots; therefore, when the lot size is split at a stage, multiple set-up costs are incurred at that stage and at subsequent stages (reflection on the model will show that the case where the following stage has a larger lot size cannot be optimal);

c) the transportation cost of lots is regarded as a sunk cost or, alternatively, it is included in the fixed cost per lot (this assumes that the transportation cost is independent from the lot size);

d) the stage holding cost per unit for a stage is never lower than that for a preceding stage (this assumption is to simplify the problem; however, it is also justified in practice when the inventory holding cost is proportional to the value of the product);

e) the lot size at a given stage is not restricted to be an integer number (however, the optimization methods proposed could be modified to accommodate this restriction).

The objective for both models is to minimize the sum of the fixed costs and the inventory holding costs of the system. An optimal solution is given for Model I and a close approximation to the optimal solution is found for Model II. Both optimization procedures are computationally efficient, (in contrast to other known methods) for a large number of stages. This is demonstrated by extensive computational experience.

Symbols common to both Model I and Model II are as follows:

D = demand rate of the final product (at stage 1);

P_i = production rate at stage i; note that $P_i > D$;

F_i = set-up cost per lot at stage i; note that $F_1 > 0$;

G_i = fixed transportation cost for a lot or for a sub-batch at stage i;

c_i = inventory holding cost per unit at stage i;

Q_i = lot size at stage i.

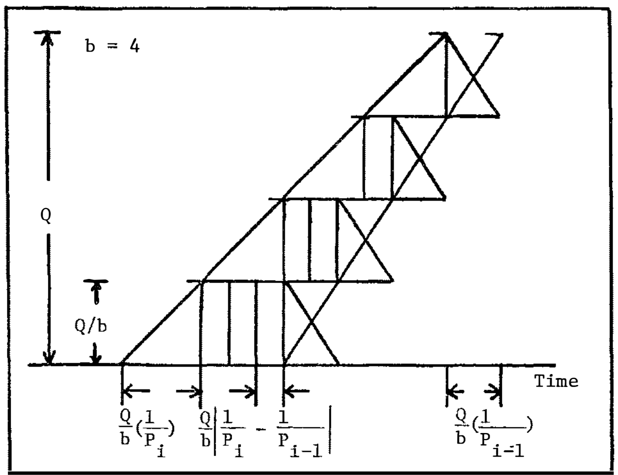
MODEL I - UNIFORM LOT SIZE WITH SUB-BATCHES

The Model and the Objective Function

In this model the lot size is the same at every stage and equal sized sub-batches can be transported from a given stage to the next production stage before the lot is finished. At all stages the entire lot is produced continuously, hence with a single set-up cost for each stage. Therefore, when the production rate of a given stage is greater than or equal to that of the following stage, (a shorter operation time followed by a longer), production at the following stage starts as soon as a sub-batch is available; subsequent sub-batches will be available to maintain the continuity of production at the following stage. On the other hand, when the production rate of a given stage is smaller than that of the following stage, production at the following stage must start with an appropriate delay. The inventory model (for the latter case) is illustrated in Figure 1, where the areas represent the time-weighted inventory for each stage.

FIGURE 1

Time-Weighted Inventory at Stage i with b=4



In this model the following additional symbols are used:

Q = uniform lot size at all stages; note that $Q = Q_1 = Q_2 = \ldots = Q_n$;

b = the number of equal sized sub-batches;

F = $\sum_{i=1}^{n} F_i$, the total set-up cost of the system;

G = $\sum_{i=1}^{n} G_i$, the total transportation cost of one sub-batch through all stages;

x = the number of units in one sub-batch; note that $x = Q/b$.

The inventory holding cost per unit time at stage i can be expressed as follows:

$$C_i = \frac{Q^2}{2b} c_i\left[\left(\frac{1}{P_i} + \frac{1}{P_{i-1}}\right) + \left|\frac{1}{P_i} - \frac{1}{P_{i-1}}\right|(b-1)\right]\frac{D}{Q} \qquad (1)$$

The inventory holding cost of the system is obtained by the summation of (1). Adding to this the total set up costs and the transportation costs (per unit time) of the sub-batches, we obtain the average total cost of the system:

$$C = \frac{D}{Q}(F + bG) + \frac{QD}{2b}\sum_{i=1}^{n} c_i\left[\left(\frac{1}{P_i} + \frac{1}{P_{i-1}}\right) + \left|\frac{1}{P_i} - \frac{1}{P_{i-1}}\right|(b-1)\right] \qquad (2)$$

where: $P_0 = D$

The objective is to minimize the total cost of the system. To simplify notation the objective function can be written:

$$\text{Minimize } C(Q,b) = \frac{D}{Q}(F + bG) + Q(M + N/b) \qquad (3)$$

subject to: $1 \leqslant b \leqslant Q$

where: $M = \sum_{i=1}^{n} \frac{c_i}{2}\left|\frac{D}{P_i} - \frac{D}{P_{i-1}}\right|$

$N = \sum_{i=1}^{n} \frac{c_i}{2}\left[\left(\frac{D}{P_i} + \frac{D}{P_{i-1}}\right) - \left|\frac{D}{P_i} - \frac{D}{P_{i-1}}\right|\right]$

$P_0 = D$

It is obvious that the larger b becomes, the smaller will be the sub-batch size, $x = Q/b$, and vice-versa. Thus, the rounding of small b or x values would make the result inaccurate. Therefore, the task is to minimize the cost function (3) subject to the integralities of b and x (of course, this also yields an integer optimal lot size).

We substitute $x = Q/b$ in (3) and the objective function becomes:

$$\text{Minimize } C(x,b) = D(F/b + G)/x + x(Mb + N) \qquad (4)$$

subject to: b and x integers; $1 \leqslant b \leqslant Q$ and $1 \leqslant x \leqslant Q$.

Optimization Without Integrality Constraints

We first ignore integrality restrictions. By partially differentiating (4) with respect to x and b, and by solving separately the extremal equations $\partial C(x,b)/\partial x = 0$ and $\partial C(x,b)/\partial b = 0$ we obtain:

$$x = \sqrt{\frac{D(F/b + G)}{Mb + N}} \qquad (5)$$

$$b = \sqrt{\frac{DF}{Mx^2}} \qquad (6)$$

Solving (5) and (6) we obtain the optimal values:

$$x^* = \sqrt{\frac{DG}{N}} \qquad (7)$$

$$b^* = \sqrt{\frac{FN}{GM}} \qquad (8)$$

It also will be useful to have:

$$C(x) = \min_b C(x,b) = DG/x + Nx + 2\sqrt{MDF} \qquad (9)$$

which is obtained by substituting (6) in (4), and

$$C(b) = \min_x C(x,b) = 2\sqrt{D(F/b + G)\ (Mb + N)} \qquad (10)$$

which is obtained by substituting (5) in (4). It could be shown that both C(x) and C(b) are unimodal. This also will be useful in the next section.

Algorithm for Finding Integer b and x Values

In subsequent expressions, square brackets around a variable indicate that only integer values of that variable are considered, e.g. the integer values of x are denoted [x]; $x_\downarrow$ denotes x rounded down and $x_\uparrow$ denotes x rounded up.

Before showing the algorithm for determining [x]* and [b]*, we describe the logic of the optimization. First, we determine the continuous optimal values for x* from (7) and for b* from (8). Then, we initiate the search on either of the axes. If $x^* \leq b^*$ we search integers on the x axis, otherwise we search integers on the b axis. Let us illustrate searching on the x axis. We select $x_* = [x^*_\downarrow]$ as the initial integer value of x and compute b in (6) at this x_*. We also compute the cost C(x,b) in (4) with two integer values of $[b_\downarrow]$ and $[b_\uparrow]$ and retain C_{min}, the smaller of the costs. Then we continue searching for a better cost on both sides of x_* simultaneously, i.e., we repeat calculating b and C(x,b) at x_*-1, x_*+1, x_*-2, x_*+2,...,etc, and retain the lowest of the C_{min} costs obtained at integer b's. Equation C(x) in (9) is the cost which is the lower bound for all b values at a given x value; therefore, we compute C(x) at each integer x value and end the search on the particular side of x_* whenever a C(x) value is higher than the lowest C_{min} value already obtained. Since C(x) is unimodal, this procedure guarantees an optimal integer solution.

The algorithm for finding $[x]^*$ and $[b]^*$ is illustrated for an initial search on the x axis:

1. Find x^* in (7) and b^* in (8). Since $x^* \leq b^*$, set $x_* = [x^*_\downarrow]$. If $x_* < 1$, set $x_* = 1$. Retain x_*. Set L = 1; set FLAG1 = 0, set FLAG2 = 0, $C_{min} = \infty$. Set $x' = x_*$ and call Subroutine CXB.

2. If FLAG1 = 1, go to 4.

3. Check optimality and/or search for a better C_{min} on the lower side of x_*. Set $x' = x_* - L$ and compute C(x') in (9). If $x' \leq 1$ or $C(x') \geq C_{min}$, set FLAG1 = 1 indicating that the search ended on the lower side of x_*. If $C(x') < C_{min}$, call Subroutine CXB.

4. If FLAG2 = 2, go to 6.

5. Check optimality and/or search for a better C_{min} on the upper side of x_*. Set $x' = x_* + L$ and compute C(x') in (9). If $C(x') \geq C_{min}$, set FLAG2 = 2 indicating that the search ended on the upper side of x_*. If $C(x') < C_{min}$, call Subroutine CXB.

6. If FLAG1 = 1 and FLAG2 = 2, stop. The optimum solution is: $[x]^* = x_{min}$, $[b]^* = b_{min}$, $C([x]^*, [b]^*) = C_{min}$ and $[Q]^* = [x]^*[b]^*$. Otherwise, set L = L + 1 and go to 2.

7. Subroutine CXB
   Compute b in (6) at x'; if b < 1, set b = 1. Compute by (4) $C(x'[b_\downarrow])$ and $C(x'[b_\uparrow])$, or only C(x',b) when b = 1. Select b' from $\{[b_\downarrow],[b_\uparrow]\}$ to minimize the cost, C_{min}. Retain the best C_{min} obtained thus far, and the minimizing x_{min} and b_{min} values. Return.

Had we found that $b^* < x^*$ in step 1 of the algorithm, the search would have been initiated on the b axis instead of the x axis. However, equations (5) and (10) would have been used in lieu of (6) and (9). The following example will illustrate the algorithm for such a case.

Example 1

TABLE 1

Problem Parameters for Example 1

i	F_i	G_i	c_i	P_i
1	5.0	5.0	2.0	1000
2	35.0	5.0	1.7	1600
3	395.0	5.0	1.3	400
4	220.0	5.0	0.8	2500
n = 4			D = 300	

From the problem parameters we calculate by (3):

$$M = 1.41325,\ N = 1.25850,\ F = \sum_{i=1}^{n} F_i = 655.0,$$

$$G = \sum_{i=1}^{n} G_i = 20.0$$

We now use the algorithm to find $[x]^*$ and $[b]^*$:

1. In (7), $x^* = 69.05$; in (8), $b^* = 5.40$. Since $b^* < x^*$, we initiate the search on the b axis: $b_* = 5$, $b' = b_* = 5$. We set L = 1, FLAG1 = 0, FLAG2 = 0, $C_{min} = \infty$. Call Subroutine CXB. In (5), x = 73.77. By (4), $C(b',[x_\downarrow]) = C(5,73) = 1228.26$ and $C(b'[x_\uparrow]) = C(5,74) = 1228.19$. $C_{min} = 1228.19$ at $b_{min} = 5$ and $x_{min} = 74$.

2. FLAG1 = 0, continue.

3. $b' = b_* - L = 5 - 1 = 4$. In (10), C(b') = C(4) = 1234.50. $C(b') > C_{min}$, hence FLAG1 = 1; search ended on the lower side of b_*.

4. FLAG2 = 0, continue.

5. $b' = b_* + L = 5 + 1 = 6$. In (10), C(b') = C(6) = 1228.57. $C(b') > C_{min}$, hence FLAG2 = 2; search ended on the upper side of b_*.

6. FLAG1 = 1 and FLAG2 = 2, stop. Optimal solution obtained:
   $[b]^* = b_{min} = 5$, $[x]^* = x_{min} = 74$, $C([x]^*,[b]^*) = C_{min} = 1228.19$, $[Q]^* = [x]^*\ [b]^* = 370$.

"Sunk" Transportation Costs

Transportation costs in a plant are often regarded as sunk costs because the transportation system must handle a whole spectrum of products and it is difficult to allocate the costs of the system to particular product lots. In such cases the sub-batch size is usually predetermined to suit the load capacity, or the best utilization, of the transportation equipment. Undoubtedly, such a choice of sub-batch size yields only a sub-optimal solution, but it still results in a better cost than transporting whole lots.

If the sub-batch size is a fixed integer, $[x^S]$, the sub-optimal number of sub-batches, $[b^S]$, can be found by setting G = 0 in the cost function (4) and applying only the subroutine of the algorithm.

Computational Experience and Conclusions

Numerous randomly selected examples were solved for different numbers of stages (n = 5, n = 10, n = 20, n = 30) on a CDC 6400 computer. No significant difference was found in the execution times for smaller and larger number of stages; the small differences appeared to be rather data dependent. Generally, the execution time was between 0.025 and 0.035 seconds per case for $n \leq 30$.

It is also interesting to note that optimum solutions for all cases were obtained with no more than two iterations. In some contrived cases there could be more than two iterations; nevertheless the computational procedure will always yield an optimum solution.

MODEL II - VARYING LOT SIZES

The Model and the Objective Function

In this model the lot size may be different at various stages; reflection on the model will show that the case where a following stage has a larger lot size cannot be optimal. The lot size is an integer multiple of the lot size at the stage that follows it, and only lot-sized intershipments are allowed between stages (including the final stage). This inventory model is illustrated by Figure 2 in which the triangular and rectangular areas (distinguished with different lines) represent the time-weighted inventory for each of four stages.

The following additional symbols are used:

$S_{i-1} = Q_i/Q_{i-1}$ for i = 2,3,...,n; note that S_{i-1} is an integer;

$\Pi_{i-1} = \Pi_{j=1}^{j=i-1} S_j$; for later notational convenience we set $\Pi_0 = \Pi_{-1} = 1$.

$c_i \leq c_{i-1}$ for i = 2,3,...n.

To simplify certain expressions, we define for stages i = 1,2,...,n:

$b_i = c_i(D/P_i + 1)/2$

$d_i = c_i(D/P_{i-1} - 1)/2$, where $P_0 = D$

$e_i = F_iD$

The inventory holding cost per unit time at stage i can be determined from the triangular and rectangular areas in Figure 2. For the triangular areas we obtain:

$$c_i\left(\frac{Q_i^2}{2P_i} + \frac{Q_{i-1}^2}{2P_{i-1}}\right)\frac{D}{Q_i} = Q_1\Pi_{i-2}c_i\left(S_{i-1}\frac{D}{P_i} + \frac{1}{P_{i-1}}\right)/2$$

for the rectangular areas it is:

$$c_iQ_{i-1}\left(\frac{Q_1}{D}\right)\frac{S_{i-1}(S_{i-1}-1)}{2}(S_1S_2\ldots S_{i-2})\frac{D}{Q_i}$$
$$= Q_1\,\Pi_{i-2}\,c_i(S_{i-1}-1)/2$$

The set-up cost per unit time at stage i is:

$$F_iD/Q_i = F_iD/Q_1\Pi_{i-2}S_{i-1}$$

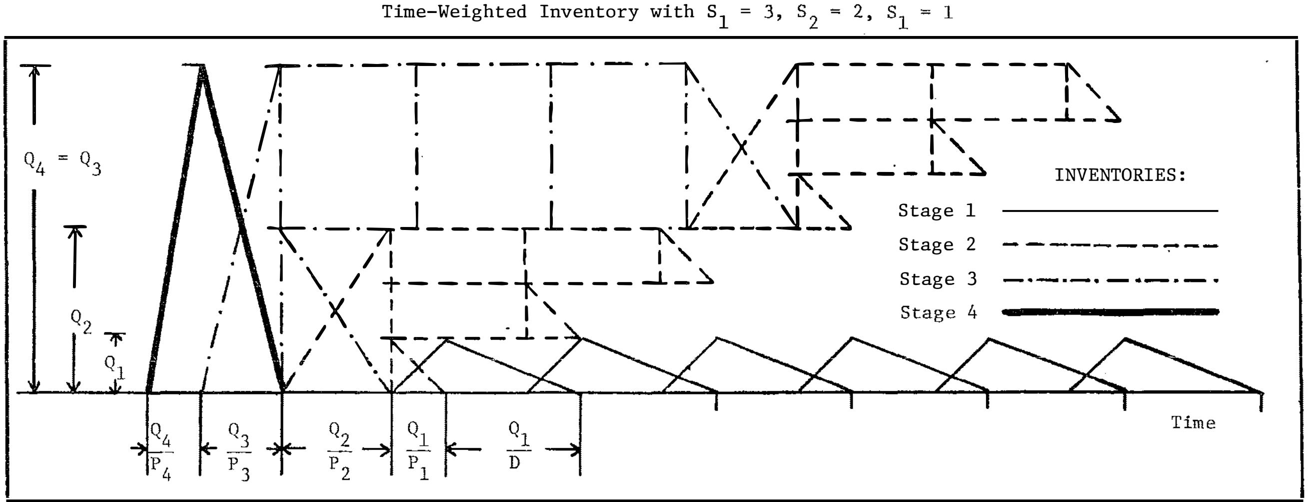
Consequently, the total of inventory holding costs and set-up costs per unit time for stage i is:

$$C_i = Q_1\Pi_{i-2}\left[S_{i-1}c_i\left(\frac{D}{P_i} + 1\right)/2 + c_i\left(\frac{D}{P_{i-1}} - 1\right)/2\right] + \frac{F_iD}{Q_1\Pi_{i-2}S_{i-1}} \qquad (11)$$

where: $P_0 = D$

FIGURE 2

Time-Weighted Inventory with $S_1 = 3$, $S_2 = 2$, $S_1 = 1$



Thus, C_i is a simultaneous function of Q_1, S_{i-1} and Π_{i-2}. The expression for C_i can be simplified to:

$$C_i = Q_1\Pi_{i-2}(S_{i-1}b_i + d_i) + \frac{e_i}{Q_1\Pi_{i-2}S_{i-1}} \tag{12}$$

The objective is to minimize the total cost of the system. The objective function is obtained by summation of (12) over all stages:

$$\text{Minimize}\quad TC(Q_1\zeta) = Q_1 \sum_{i=1}^{n} \Pi_{i-2}(S_{i-1}b_i + d_i) + (1/Q_1)\sum_{i=1}^{n} e_i/\Pi_{i-2}S_{i-1} \tag{13}$$

where ζ is the set $\{S_1, S_2, \ldots, S_{n-1}\}$

We partially differentiate (13) with respect to Q_1, set the derivative to zero and solve to obtain the minimizing Q_1, which we designate Q_{1min}:

$$Q_{1min} = \sqrt{\frac{\sum_{i=1}^{n} e_i/(\Pi_{i-2}S_{i-1})}{\sum_{i=1}^{n} \Pi_{i-2}(S_{i-1}b_i + d_i)}} \tag{14}$$

For a given Q_1 the cost at any i'th stage is defined by Π_{i-2} and S_{i-1}. A dynamic programming approach, using Π_{i-2} as a stage variable and S_{i-1} as a decision variable, will be used to minimize the total cost (13).

Optimization Without Integrality Restrictions

It will be handy, for several reasons, to have a solution to the minimization of (13) but with no integrality restrictions on the S_i's. There are a number of ways of solving a problem of this type. However, the convenient and efficient "collapsing" method used by Schwarz and Schrage [4] will be employed. A brief outline is given in this section.

To distinguish this case we use q_i's instead of Q_i's and s_i's instead of S_i's. We define $s_{i-1} = \frac{q_i}{q_{i-1}}$ for i = 2,...,n, and δ to be the set

$$\{s_1, s_2, \ldots, s_{n-1}\}. \tag{15}$$

Thus, the total cost in (13) can be rewritten:

$$TC_c(\Phi) = \sum_{i=1}^{n} q_i(b_i + d_{i+1}) + \sum_{i=1}^{n} \frac{e_i}{q_i} \tag{16}$$

subject to: $q_1 < q_2 < q_3 < \ldots < q_n$,

where: $\Phi = \{q_1, q_2, \ldots, q_n\}$.

This can be converted to the expression

$$TC_c(\Phi) = \sum_{i=1}^{n} (K_i q_i + M_i/q_i) \tag{17}$$

where: $K_i = b_i + d_{i+1}$ for i = 1,...,n, and we set $d_{n+1} = 0$;

$M_i = e_i$

Let us now define: q_i^N as the optimal lot size computed independently for stage i:

$$q_i^N = \sqrt{\frac{M_i}{K_i}} \tag{18}$$

It is evident that in the absence of constraints, (18) would give the optimum q values. Since (17) is a convex objective function and constraints are convex, the Kuhn-Tucker conditions are necessary and sufficient for optimality. The problem can be written:

$$\text{Minimize}\quad \sum_{i=1}^{n} (K_i q_i + M_i/q_i) \tag{19}$$

subject to: $q_i - q_{i+1} \leqslant 0$ for i=1,...,n-1

The Kuhn-Tucker conditions (somewhat simplified because we know that the q_i's are always positive for non-trivial cases) are:

$$K_1 - M_1/q_1^{*2} + u_1 = 0 \tag{20}$$

$$K_i - M_i/q_i^{*2} - u_{i-1} + u_i = 0 \quad \text{for } i = 2, \ldots, n-1$$

$$K_n - M_n/q_n^{*2} - u_{n-1} = 0$$

where: $q_1^*, \ldots, q_n^*$ are the optimal q values.

The u_i's are positive or zero; also, $u_i > 0$ only when constraint i in (20) is tight and holds as an equality.

We can now show (this proof is similar to one in Schwarz and Schrage [4]) that:

$$\text{If } q_i^N < q_{i-1}^N \text{ then } q_i^* = q_{i-1}^*. \tag{21}$$

To prove this we assume the contrary conditions, i.e., $q_i^N < q_{i-1}^N$ and $q_i^* > q_{i-1}^*$ and show that they are not possible. Since, using (20),

$$q_i^* = \sqrt{\frac{M_i}{K_i - u_{i-1} + u_i}}$$

and because $q_i^* > q_{i-1}^*$ means that $u_{i-1} = 0$, the consequence is that $q_i^* \leqslant q_i^N$. Using the given conditions, we now have:

$$q_i^* < q_{i-1}^N \tag{22}$$

Further, since

$$q_{i-1}^* = \sqrt{\frac{M_{i-1}}{K_{i-1} - u_{i-2} + u_{i-1}}}$$

the consequence is that $q_{i-1}^* \geqslant q_{i-1}^N$. From the given conditions we now have:

$$q_i^* > q_{i-1}^N \tag{23}$$

The inequalities (22) and (23) are contradictory and the proof is complete.

To find the optimum solution to (19), we first find the quantities q_i^N from (18). Suppose that there is an i such that $q_i^N < q_{i-1}^N$ (if there isn't, then $q_i^* = q_i^N$ for all i). Since $q_i = q_{i-1}$ at optimality, we can combine the two stages so that $K_i' = K_i + K_{i-1}$ and $M_i' = M_i + M_{i-1}$. We now have a new problem of the type (17), but with one less stage. This collapsing continues until it can proceed no further and hence the set of optimal q_i values has been found:

$\Phi^* = \{q_1^*, q_2^*, \ldots, q_n^*\}$, and the set of optimal s_i's from (15) is: $\delta^* = \{s_1^*, s_2^*, \ldots, s_{n-1}^*\}$.

Example 2 illustrates the computational procedure.

Example 2

Here we use the same data as in Example 1. However, it is assumed that the set-up cost F_i includes G_i, the transportation cost of the lot (which is independent of the lot size) at stage i; i.e. $F_i = F_i + G_i$. Consequently the only change in the problem parameters in Table 1 is that:

$F_1 = 10.0$, $F_2 = 40$, $F_3 = 400$, $F_4 = 225$, and $G_i = 0$ at each stage.

From the problem parameters we can calculate q_i^N using (18) as it is illustrated in Table 2.

TABLE 2

Calculation of q_i^N Values

i	b_i	d_i	$e_i = M_i$	K_i	q_i^N
1	1.300	0.000	3000.	0.705	65.23
2	1.009	-0.595	12000.	0.481	157.91
3	1.138	-0.528	120000.	1.038	340.09
4	0.448	-0.100	67500.	0.448	388.16

Since $q_1^N < q_2^N < q_3^N < q_4^N$ no collapsing is necessary. We have obtained a solution to (19) and, consequently, to (16): $q_1^* = 65.23$, $q_2^* = 157.91$, $q_3^* = 340.09$, $q_4^* = 388.16$ or $\Phi^* = \{65.23, 157.91, 304.09, 388.16\}$ and $TC_c(\Phi^*) = TC^*(q_1^*) = 1297.45$. From (15), $s_1^* = 2.42$, $s_2^* = 2.15$, $s_3^* = 1.14$ or $\delta^* = \{2.42, 2.15, 1.14\}$.

The "Rounded Solution"

The solution to (16) is important for two reasons. First, since (16) is merely the minimization of (13), but with relaxed constraints, the solution to (16) gives a lower bound on the minimum of (13). Secondly, we can obtain an approximate minimum to (13) by "rounding".

Let $S_i^R = [s_i^*]$, where the brackets indicate conventional rounding to an integer. If we use S_i^R for S_i in (14) we obtain q_{1R}^*, and the minimum total cost of this "rounded solution" can be computed. The value of q_{1R}^* must yield a cost equal to or less than if q_1^* were used with S_i^R's.

From Example 2, we have $S_1^R = 2$, $S_2^R = 2$, $S_3^R = 1$ or or $\zeta_R = \{2,2,1\}$. Using (14) we obtain $q_{1R}^* = 85.69$. The cost of the rounded solution, by substitution in (13), $TC(q_{1R}^*, \zeta_R) = TC_*(\zeta_R) = 1304.12$.

The approximate rounded solution, as shown by computational experience, quite often could be the same as the optimal solution.

Dynamic Programming to Find Integer S_i's at a Given Q_1

In this section, we will discuss the problem of minimizing (13) for the case where a set of permissible values for S_i is given.

Let $SI = \{G_1, \ldots, G_{n-1}\}$, where G_i is the set of permissible values for S_i. Given SI, we can construct $PI = \{H_1, \ldots, H_{n-2}\}$, where H_i is the set of permissible values for Π_i. In practice, such limitations on S_i or Π_i could actually exist as a result of storage limitations or transportation convenience; however, they can be eliminated by extending the search region.

Let us assume for the moment that Q_1 is known in (13) and that the best S_i's must be selected from SI for that Q_1. The problem of minimizing (13) can be solved by a standard dynamic programming approach.

At any stage i, let Π_{i-2} be the state variable, and S_{i-1} be the decision variable. As seen from (12) the cost C_i of any stage, at a given Q_1, is defined by these two variables. We can now write the recursive relationship.

$$f_i^*(\Pi_{i-2}) = \min_{S_{i-1} \varepsilon G_{i-1}} (C_i(S_{i-1}, \Pi_{i-2}) + f_{i+1}^*(\Pi_{i-2} S_{i-1})) \quad (24)$$

where: $\Pi_{i-2} \varepsilon H_{i-2}$, $S_{i-1}\Pi_{i-2} = \Pi_{i-1}$, and we set $f_{n+1}^*(\Pi_{n-2} S_{n-1}) = 0$.

This recursive relationship can be used in the conventional way; moving backward from stage n to stage 1 and finding the optimal policy S_{i-1} for each state Π_{i-2} of stage i. After this is finished, at stage 1 the optimum S_1 will be found; tracing forward through the stages will give the rest of the S_i's. Thus, having found the set $\zeta = \{S_1, S_2, \ldots, S_{n-1}\}$, by using (4) we can find

$$TC^*(Q_1) = \min_{\zeta} TC(Q_1, \zeta) \quad (25)$$

The "Likely Optimum Solution"

We now describe the finding of a solution that we call (in high hopes) the "likely optimum solution." We first define

$$TC_*(\zeta) = \min_{Q_1} TC(Q_1 \zeta) \quad (26)$$

and we denote the Q_1 that minimizes $TC(Q_1, \zeta)$ with $Q_{1\zeta}^*$.

First, the collapsing method is used to find q_1^* as described before. Then, we find the best ζ at q_1^* by dynamic programming described in (24). Once the best ζ is found, a $Q_{1\zeta}^*$ can be determined by (14) and $TC_*(\zeta)$ can be computed. Using the $Q_{1\zeta}^*$, the cycle of dynamic programming and application of (14) is continued iteratively until no further improvement in cost can be found. The result will be the likely optimum solution, $TC(Q_{1L},\zeta_L) = TC_*(\zeta_L)$. Example 3 illustrates the procedure and notation.

Example 3

We continue with the problem of Example 1. Assume that

$SI = \{\{1,2,3,4\}, \{1,2,3,4,5,6,7\}, \{1,2,3,4,5,6,7\}$, hence $PI = \{\{1,2,3,4,5\}, \{1,2,3,4,5,6\}\}$.

To find the likely optimum solution we first determined, by dynamic programming, ζ_1 at $q_1^* = 65.23$ (note that q_1^* was calculated in Example 2). The result was $S_1 = 2$, $S_2 = 3$, $S_3 = 1$ (or $\zeta_1 = \{2,3,1\}$). Using ζ_1 we calculated by (14) $Q^*_{1\zeta_1} = 61.8$, and the corresponding cost by (13) was $TC_*(\zeta_1) = 1305.18$. We now illustrate the dynamic programming procedure by finding ζ_2 at the new Q_1 (which is $Q^*_{1\zeta_1}$).

Table 3 illustrates the calculations according to (24).

It can be seen in Table 3 that $f_2^*(\Pi_0) = 1173.60$ occurs at $S_1 = 3$; consequently $\Pi_1 = 3$. At $\Pi_1 = 3$ the best cost is $f_3^*(\Pi_1) = 958.68$ which occurs at $S_2 = 2$, consequently $\Pi_2 = 6$. At $\Pi_2 = 6$ the best cost is $f_4^*(\Pi_2) = 311.18$ which occurs at $S_3 = 1$. Hence $\zeta_2 = \{3,2,1\}$. Now using ζ_2 in (14) we calculate $Q^*_{1\zeta_2} = 58.80$, and using (13) we find the cost to be $TC_*(\zeta_2) = 1300.94$.

Since the application of dynamic programming at $Q^*_{1\zeta_2} = 58.80$ does not change ζ_2 (i.e. ζ is stabilized), we have obtained the likely optimum solution:

$Q^*_{1\zeta_2} = Q_{1L} = 58.80$; $S_1^L = 3$, $S_2^L = 2$ and $S_3^L = 1$

or $\zeta_L = \{3,2,1\}$; $TC(Q_{1L},\zeta_L) = TC_*(\zeta_L) = 1300.94$.

The likely optimum solution, as shown by computational experience, very frequently could be the same as the optimal solution.

TABLE 3

Dynamic Programming

$f_4(\Pi_2) = C_4(S_3,\Pi_2)$

Π_2 \ S_3	1	2	3	4	5	6	7
1	1115.9	596.3	441.5	378.0	350.8	342.0	343.6
2	590.1	371.8	335.9	345.5			
3	429.2	329.7	351.8				
4	359.5	333.2	398.1				
5	326.2	354.9					
6	<u>311.2</u>	385.8					

$f_3(\Pi_1) = C_3(S_2,\Pi_1) + f_4^*(\Pi_2)$

Π_1 \ S_2	1	2	3	4	5	6	7
1	2325.2	1416.4	1156.1	1067.7	1033.5	1023.8	1043.1
2	1383.8	1035.1	991.3	1047.8			
3	1091.0	<u>958.7</u>	1064.7				
4	969.9	982.7					
5	903.2	1057.4					

$f_2(\Pi_0) = C_2(S_1,\Pi_0) + f_3^*(\Pi_1)$

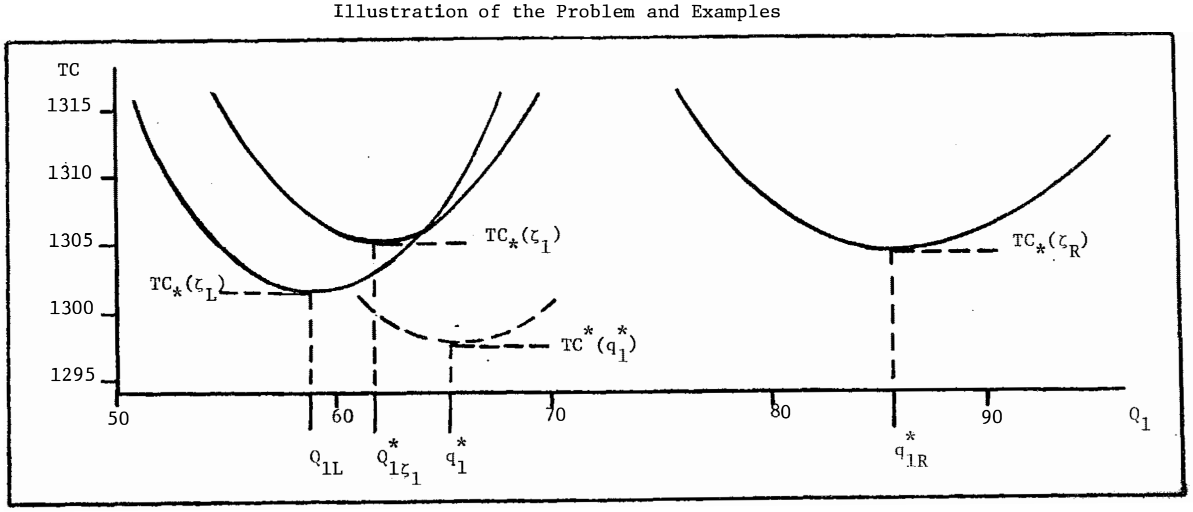
Π_0 \ S_1	1	2	3	4
1	1243.9	1176.4	<u>1173.6</u>	1230.9

The Approximate Optimum Solution

Figure 3 illustrates the cost curves obtained by the solution procedures applied to the example. The minimum cost with continuous s_i's, $TC^*(q_1^*)$, is the lower bound to the solution with integer S_i's. The minimum cost obtained by the likely optimum solution, $TC_*(\zeta_L)$, is smaller than that computed by the

FIGURE 3

Illustration of the Problem and Examples



rounded solution, $TC_*(\zeta_R)$. Hence, in our example the likely optimum solution is chosen for the approximate optimum solution (in fact this is the optimum solution). Of course, the smaller of the two costs will always indicate the approximate optimum solution. Computational experience has shown that the approximate solution is overwhelmingly close (or often identical) to the optimum.

The optimum solution can be computed with an elaborate search procedure somewhat similar to that applied for the likely optimum solution. The achievement of optimality depends on the width of search regions in the dynamic programming procedure. The widths adapted in our computations were such that, for practical purposes, we were assured of optimality.

Computational Results and Conclusions

The algorithm was tested on a large number of problems (with integer S_i's). Test problems were generated in the following manner. Ranges were chosen for the problem parameters: Range (D) = [5000,50000]; Range (P) = [60000, 625000]; Range (c) = [0.1,2.5]; Range (F_1) = [1,500]; Range ($F_{i>1}$) = [0,500].

For D, P and c, in each test problem, values were simply chosen by using a uniform probability distribution within each range. Since values of c_i must decrease with increasing stage number ($c_i < c_{i-1}$), the values generated were ordered accordingly. In practice, zero set-up costs can occur at certain stages, thus, for one half of the cases, F's were chosen so that one sixth of them took up at random a zero value. On the other hand, when $F_1 = 0$ was generated, it was adjusted to $F_1 = 1$; otherwise the model breaks down (if $F_1 = 0$, $Q_1 = 0$). The remaining values were sampled uniformly within the range. Of course, Q_1 is actually an integer and a continuous value is used only as an approximation; if Q_1 is not very small the resulting error is negligible. The parameters were thought to be "reasonable and representative" of what might be encountered in practice. Some contrived problems were also computed, but no discernible difference in performance was found. Tables 4 and 5 give comparisons of the performance and execution times (on a CDC 6400 computer) of rounded and likely optimum solutions.

TABLE 4

Accuracy of Solutions and CPU Times Per Case

No. of Stages	No. of Cases	Rounded Solution		Likely Solution	
		% Optimal	Seconds	% Optimal	Seconds
5	400	79.50	0.004	92.75	0.033
10	400	65.50	0.008	85.50	0.077
20	400	33.50	0.018	74.50	0.210
30	400	28.75	0.037	71.75	0.404

Some rather obvious generalizations can be made from the computational results in Table 4. The rounded solution is extremely fast even for large n, but it is optimal in a large percent of cases only for small n values (79.50% for n = 5). The likely optimum solution is somewhat slower, but is optimal in a large percent of cases even for large n values (71.75% for n = 30). Note that in computing the likely optimum solution, it takes a trivially small extra time to obtain the rounded solution. Both solutions can be obtained with virtually the same amount of time that is needed for the likely optimum solution.

TABLE 5

Cost Ratios of Approximate and Optimum Solutions

No. of Stages	Percent of Cases at Cost Ratios or Below					Highest Ratio
	1.000	1.005	1.010	1.020	1.030	
5	96.75	98.75	99.75	100.00	-	1.011
10	92.75	97.75	99.75	100.00	-	1.018
20	86.25	97.00	99.50	99.75	100.00	1.023
30	81.00	93.50	98.25	99.75	100.00	1.025

As shown in Table 5, the approximate optimum, the better of the two solutions for the same case is, in fact, optimal in a large percent of cases even for large n values (81.00% for n = 30). Further, the costs obtained by the approximate solution do not exceed the optimum costs by more than 1% in 99.50 per cent of the cases if $n \leqslant 20$, and the excess cost is hardly more than 2% in any case.

COMPARISON OF THE TWO MODELS AND CONCLUSIONS

Quantitative Considerations

Each of the two models implies and identifies a particular organization of the production process and is based on defined assumptions. There can be variations in process organization for which neither model would fit; nevertheless, our models can be regarded as two basic models for a multi-stage serial system. It can be shown that the cost function of each model reduces to the same expression, when the lot size is the same at each stage and only lot-sized intershipments are allowed.

When the stage inventory unit holding costs are not increasing monotonically, the cost function for Model II does not identify the best process organization. However, Model I still can be applied to such cases. When the conditions of both models are satisfied (the unit holding costs are monotonically increasing and sub-batching is feasible) the choice would be determined by least cost. We have used the same problem parameters for both models in our examples. Model I (Example 1) yielded a cost of 1228.19, and the cost computed for Model II (Example 3) was 1300.94. Thus, in our particular problem, Model II resulted in a 6% higher cost. Of course, this could be different for another problem; for example, if the transportation cost of a sub-batch had been greater.

Qualitative Considerations

The application of the models in the case of "sunk" transportation costs needs special attention. Sub-batch size (the optimal intershipment quantity) and, therefore, the optimal number of sub-batches is often determined by technical feasibility (for example, the sub-batch size may be chosen so that it fully utilizes the load capacity of the transportation equipment). Model I can be applied appropriately for such a case. One must remember, however, that in Model II the intershipment cost is incorporated in the fixed costs per lot at each stage (as in Example 3) and is assumed to be the same for any lot size. Consequently, if lot sizes at different stages are multiples of the optimal intershipment size, some additional and unaccounted transportation costs exist. Such considerations may warrant adjustments to the calculated cost.

In both models we distinguish between finished product inventory (inventory at the final stage) and process inventory (inventory at all other stages). It can be shown graphically that when the initial lot size, Q_n, as well as b in Model I and Π_{n-1} in Model II are about the same (as in our examples), the basic difference between the two models is that a relatively larger proportion of the average inventory is shifted to the final stage (Stage 1) in Model I than in Model II. Although, this occurrence is problem parameter dependent and is accounted for in the calculated minimum costs for the models, we must be careful because stage inventory unit holding costs derived from cost accounting data may be more unreliable for process inventories. Since they are difficult to measure, many intangible cost factors (e.g. relatively more expensive storage and handling costs in the plant than in stores, record keeping of process inventories, scheduling costs, etc.) are usually ignored in process inventory costs which, therefore, can easily be underestimated.

It is realistic to assume that the more variety of stage inventory exists at a time, the more expensive it is to hold the process inventory. A measure of this variety can be expressed by comparing the manufacturing cycle time (the maximum time span during which any unit from an initial lot is in process inventory), and the demand cycle time (the maximum time span during which any unit from an initial lot is in finished product inventory). The determination of the manufacturing cycle time is discussed in [5]. Accordingly, for Model I the manufacturing cycle time is:

$$T_1^m = \frac{Q_n}{b} \left[\sum_{i=1}^{n} \frac{1}{P_i} + (b-1) \sum_{i=1}^{n} \left(\frac{1}{P_i} - \frac{1}{P_{i+1}}\right) I_i \right] \quad (27)$$

where:

$\frac{1}{P_{n+1}} = 0$; $I_i = 0$, if $\frac{1}{P_i} \leqslant P_{i+1}$; $I_i = 1$, if $\frac{1}{P_i} > \frac{1}{P_{i+1}}$

For Model II the manufacturing cycle time, derived from Figure 2, is as follows:

$$T_2^m = Q_1 \sum_{i=1}^{n} \frac{\Pi_{i-1}}{P_i} + \frac{Q_1}{D} (\Pi_{n-1} - 1) \quad (28)$$

The demand (usage) cycle time of the initial lot size in each model is:

$$T_j^d = Q_{nj}/D, \text{ for } j = 1,2 \quad (29)$$

where: Q_{nj} is the lot size at stage n in Model j.

The ratio between the holding time of process inventory and that of final product inventory of one initial lot indicates the average number of initial lots in process:

$$r_j = T_j^m/T_j^d \quad (30)$$

The meaning of this is that: if $r_j \leqslant 1$, there is only one initial lot in process at a time; if $r_j \geq 1$, then more than one initial lot could be found in process at some time.

Table 6 contains the cycle time data derived from the problem parameters and the results of our examples. We can see from Table 6 that the average number of initial lots in process is 0.97 for Model I, and 1.84 for Model II. This implies that in our particular problem one must keep track of a larger variety of stage inventories in Model II than in Model I.

Again, this result is problem parameter dependent. Nevertheless, the analysis shown is useful because the effect of a larger variety of stage inventories on holding costs is extremely difficult to evaluate and to incorporate in stage inventory unit holding costs. Therefore, in effect, the stage inventory unit holding costs may not be the same for both models. The cycle time analysis presented here could induce further adjustments to the calculated cost.

TABLE 6

Cycle Time Data for the Two Models

Symbols	Model I	Model II
Q_{nj}	370.00	352.00
D	300.00	300.00
T_j^m	1.19	2.17
T_j^d	1.23	1.18
r_j	0.97	1.84

Conclusions

Comparison of the two models has revealed the effects of various basic assumptions in multi-stage inventory systems. One must remember that each model fits a particular process organization. The results derived from the model are only valid if the process organization implied by the model is followed. Several models may be feasible under certain conditions. The selection of the "best" is problem parameter dependent; therefore, the result of different models must be compared. It was shown that inventory unit holding costs incorporated into the models may not account for subtle (but important) effects generated by differing process organizations. Consequently, beyond the quantitative comparisons of results derived from alternative models, a careful qualitative scrutiny of models and data is very much desirable.

REFERENCES

[1] Clark, A.J., "An Informal Survey of Multi-Echelon Inventory Theory." Naval Research Logistics Quarterly, Vol. 19 (December 1972), pp. 621-650.

[2] Crowston, W.B., Wagner, M. and Williams, J.F., "Economic Lot Size Determination in Multi-Stage Assembly System." Management Science, Vol. 19, No. 5 (January 1973), pp. 517-527.

[3] Jensen, P.A., and Kahn, H.A., "Scheduling in a Multi-Stage Production System with Set-Up and Inventory Costs." AIIE Transactions, Vol. 4, No. 2 (June 1972), pp. 126-133.

[4] Schwarz, L.B., and Schrage, L., "Optimal and System Myopic Policies for Multi-Echelon Production/Inventory Assembly Systems." Management Science, Vol. 21, No. 11 (July 1975), pp. 1285-1294.

[5] Szendrovits, Andrew Z., "Manufacturing Cycle Time Determination for a Multi-Stage Economic Production Quantity Model." Management Science, Vol. 22, No. 3 (November 1975), pp. 298-308.

[6] ________________,"On the Optimality of Sub-Batch Sizes for a Multi-Stage EPQ Model - A Rejoinder." Management Science, Vol. 23, No. 3 (November 1976), pp. 334-338.

[7] ________________, "A Comment on Optimal and Systems Myopic Policies for Multi-Echelon Production/Inventory Assembly Systems." Research and Working Paper Series, No. 118, Faculty of Business, McMaster University, May 1976, pp. 5.

[8] Taha, Hamdy A., and Skeith Ronald W., "The Economic Lot Sizes in Multi-Stage Production Systems." AIIE Transactions, Vol. 2, No. 2 (June 1970), pp. 157-162.

Faculty of Business
McMaster University

WORKING PAPER SERIES

101. Torrance, George W., "A Generalized Cost-effectiveness Model for the Evaluation of Health Programs," November, 1970.

102. Isbester, A. Fraser and Sandra C. Castle, "Teachers and Collective Bargaining in Ontario: A Means to What End?" November, 1971.

103. Thomas, Arthur L., "Transfer Prices of the Multinational Firm: When Will They be Arbitrary?" (Reprinted from: Abacus, Vol. 7, No. 1, June, 1971).

104. Szendrovits, Andrew Z., "An Economic Production Quantity Model with Holding Time and Costs of Work-in-process Inventory," March, 1974.

111. Basu, S., "Investment Performance of Common Stocks in Relation to their Price-earnings Ratios: A Text of the Efficient Market Hypothesis," March, 1975.

112. Truscott, William G., "Some Dynamic Extensions of a Discrete Location-Allocation Problem," March, 1976.

113. Basu, S. and J.R. Hanna, "Accounting for Changes in the General Purchasing Power of Money: The Impact on Financial Statements of Canadian Corporations for the Period 1967-74," April, 1976. (Reprinted from Cost and Management, January-February, 1976).

114. Deal, K.R., "Verification of the Theoretical Consistency of a Differential Game in Advertising," March, 1976.

114a. Deal, K.R. "Optimizing Advertising Expenditures in a Dynamic Duopoly," March, 1976.

115. Adams, Roy J., "The Canada-United States Labour Link Under Stress," [1976].

116. Thomas, Arthur L., "The Extended Approach to Joint-Cost Allocation: Relaxation of Simplifying Assumptions," June, 1976.

117. Adams, Roy J. and C.H. Rummel, "Worker's Participation in Management in West Germany: Impact on the Work, the Enterprise and the Trade Unions," September, 1976.

118. Szendrovits, Andrew Z., "A Comment on 'Optimal and System Myopic Policies for Multi-echelon Production/Inventory Assembly Systems'," [1976].

119. Meadows, Ian S.G., "Organic Structure and Innovation in Small Work Groups," October, 1976.