The Part Period Algorithm discussed in Part I is compared with optimal solutions, determining the maximum deviations possible. Optimality of the algorithm is established for the case of constant demand.

Performance characteristics are compared with those of the Least Unit Cost algorithm.

An economic lot-sizing technique

II Mathematical analysis of the part-period algorithm

by A. G. Mendoza

The purpose of this paper is to analyze the solution of the Part Period Algorithm (PPA) to the economic lot-size problem with known future demands as presented by J. J. DeMatteis in Part I of this paper. Our model assumes that the manufacturing (or purchasing) cost function is a straight line with positive setup (or ordering) cost and non-negative slope; this cost function and the unit inventory holding cost are time invariant.

The analysis is directed primarily at comparing the PPA solution versus optimal solutions as, for instance, the Wagner-Whitin solution.[1] Optimal algorithms yield results that are better up to a certain point. However, the fact that such algorithms lean heavily on the size of the horizon in a given problem makes them vulnerable to any extension or reduction of that horizon, even by one time period. Thus, such algorithms are not readily adaptable to "open-ended" usage, whereas PPA is naturally geared to such usage.

For the case of constant demand, this paper establishes the optimality of PPA. For the case of non-constant demand, the maximum deviations of PPA from the optimal solution are determined. Also included is an analytical comparison of PPA with the Least Unit Cost algorithm[2] which, although not necessarily optimal, is quite elegant and economical.

The Least Unit Cost algorithm leans heavily on the demand of the already established setup period in order to determine the next setup, and assigns decreasing importance to the demands as they occur successively in future time periods. In addition, no weight is assigned to the demand of the next setup period in reaching the decision whether or not to set up in that period. PPA does not consider the demand of the already established setup period, but assigns increasing importance to successive demands, ending with the demand of the next setup period.

The term *solution* will refer to a collection of setups which, accompanied by the corresponding production (or order) quantities, satisfies a given demand vector: $\langle d_1, \cdots, d_n \rangle$. An *optimal solution* is a collection of setups in which the corresponding production amounts satisfy the demands, and the total cost of production or purchasing (including inventory holding costs) is a minimum.

We consider a time horizon that covers time period 1 through the time period immediately preceding the last setup of the PPA solution, given that the last setup is not the first or second one; i.e., we only analyze problems where the PPA solution has three or more setups, and we only consider the time horizon covered by all but the last setup. It is trivial to show that the PPA solution for the time horizon covered by the first setup is an optimal one.

Optimality for constant demand

We now compare PPA with optimal solution algorithms, assuming that all demands are equal. It is shown that no algorithm performs better than PPA, thus proving that PPA is an optimal solution algorithm for the specific case of constant demand.

Let the adjusted time horizon be N, and let P be the number of setups of the PPA solution.

First we prove that for any given solution with Q setups (hereafter called the Q solution) it is optimal to have each setup satisfy at least the demand of the time period where it occurs, plus the $L - 1$ immediately successive demands, and at most satisfy the demand of the time period after the $L - 1$ immediate successors; where

$$L \triangleq \left[\frac{N}{Q}\right]$$

i.e., the greatest integer in N/Q, and consequently

$$N = LQ + \alpha, \qquad \alpha = 0, 1, \cdots, Q - 1$$

We want to prove that, given Q setups, it is optimal to have $(Q - \alpha)$ setups, each of which satisfies L demands (from the time period in which the setup occurs through the period $L - 1$ units later), and each of the remaining α setups satisfies $L + 1$ demands (through the Lth period after the setup).

For any Q solution, the number of periods that contribute to the inventory holding costs is: $N - Q = (L - 1)Q + \alpha$. Thus, if we let $x_i (i = 1, \cdots, Q)$ be the number of demands that are satisfied by the ith setup and which contribute to the inventory holding costs (i.e., x_i does not count the time period during which the ith setup occurs), then we want to minimize the objective function

$$\sum_{i=1}^{Q} \frac{x_i(x_i+1)}{2}$$

subject to three constraints:

$$\sum_{i=1}^{Q} x_i = (L-1)Q + \alpha, \qquad x_i \geq 0 \text{ and} \qquad x_i = \text{integer}.$$

The answer to the above problem without the third constraint is readily seen to be $x_1 = x_2 = \cdots = x_Q$. Now, as we do not allow non-integer x_i's, and as the constraints are symmetric with respect to the x_i's, and the objective function is symmetric, continuous, and monotone (in each variable) in the region of interest, the minimum is attained by letting $(Q - \alpha)$ of the x_i's be equal to $L - 1$, and the remaining α be equal to L. Thus, in what follows, it is sufficient to consider those Q solutions which satisfy the above requirements.

Now let $P > Q = P - x$, for $x \geq 1$. We shall show that the PPA solution is more economical than the Q solution.

By assumption, we have $N = KP$, where each of the P setups satisfies K demands. Also, $N = L(P - x) + \alpha$, where $0 \leq \alpha \leq P - x - 1$ and the $P - x$ setups satisfy the requirements of the first constraint above.
Thus

$$K \leq L = K + Y, \qquad Y \geq 0$$

Let S be the setup cost divided by the unit inventory holding cost. Let C_P and C_{P-x} be the costs (in terms of inventory holding cost) associated with each solution respectively, where

$$C_{P-x} = S(P-x) + d\left[(P-x)\frac{L(L-1)}{2} + \alpha L\right]$$

$$C_P = SP + d\left[P\,\frac{K(K-1)}{2}\right]$$

Consider the two cases

A) $Y = 0$

Then $\alpha = Kx$, and

$$C_{P-x} - C_P = x\left[dK^2 - \left(S + d\,\frac{K(K-1)}{2}\right)\right]$$

$$= x\left[d\,\frac{K(K+1)}{2} - S\right] > 0$$

since $d\,\dfrac{K(K+1)}{2} > S$ and $x \geq 1$.

B) $Y \geq 1$

Then

$$C_{P-x} - C_P = -Sx + d\Big[(P-x)Y\Big(K - 1 + \frac{Y+1}{2}\Big) - x\frac{K(K-1)}{2} + \alpha(K+Y)\Big]$$

But as $(P - x)\,Y = Kx - \alpha$, then

$$\begin{aligned} C_{P-x} - C_P &= -Sx + d(Kx-\alpha)(K-1) \\ &\quad + d(Kx-\alpha)\frac{Y+1}{2} \\ &\quad + d\alpha(K+Y) - dx\frac{K(K-1)}{2} \\ &= x\Big[d\frac{K(K+Y)}{2} - S\Big] + d\alpha\frac{Y+1}{2} > 0 \end{aligned}$$

since $x \geq 1$, $d\alpha \dfrac{Y+1}{2} \geq 0$

and as $Y \geq 1$, then

$$d\frac{K(K+Y)}{2} \geq d\frac{K(K+1)}{2} > S. \qquad \textit{Q.E.D.}$$

Now let $P < Q = P + x$, for $x \geq 1$. As before

$N = KP$, and $N = L(P + x) + \alpha \qquad 0 \leq \alpha \leq P + x - 1$

We note that $K = L$ implies $\alpha = -Kx < 0$, which is impossible; thus $K > L = K - Y$ for $Y \geq 1$.

Also

$$C_{P+x} = S(P+x) + d(P+x)\frac{L(L-1)}{2} + d\alpha L,$$

and

$$C_P = SP + dP\frac{K(K-1)}{2}$$

Thus,

$$\begin{aligned} C_{P+x} - C_P &= xS + d\Big[x\frac{L(L-1)}{2} - PY\frac{2L+Y-1}{2} + \alpha L\Big] \\ &\qquad \text{since} \quad K = L + Y \\ &= xS + d\Big[x\frac{L(L-1)}{2} \\ &\qquad - (\alpha + Lx)\frac{2L+Y-1}{2} + \alpha L\Big] \\ &\qquad \text{since} \quad PY = \alpha + Lx \end{aligned}$$

$$= x\left[S - d\,\frac{L(L+Y)}{2}\right] - \alpha d\,\frac{Y-1}{2}$$

$$= x\left[S - d\,\frac{K(K-1)}{2}\right]$$

$$+ d\,\frac{Y-1}{2}\,(Kx - \alpha)$$

$$= x\left[S - d\,\frac{K(K-1)}{2}\right]$$

$$+ d\,\frac{Y-1}{2}\,[Y(P+x) - 2\alpha]$$

$$\text{since} \quad \alpha = Y(P+x) - Kx$$

$$\geq x\left[S - d\,\frac{K(K-1)}{2}\right]$$

$$+ d\,\frac{Y-1}{2}\,[Y(P+x) - 2\,(P+x-1)]$$

$$\text{since} \quad \alpha \leq P + x - 1$$

$$\geq x\left[S - d\,\frac{K(K-1)}{2}\right]$$

$$+ d\,\frac{Y-1}{2}\,[(P+x)(Y-2)] \geq 0$$

$$\text{since} \quad S \geq d\,\frac{K(K-1)}{2} \text{ and}$$

$$x \geq 1\,, \quad d > 0 \quad \text{and} \quad Y \geq 1\,.$$

Q.E.D.

Deviations for non-constant demand

We now compare PPA with optimal solution algorithms for a case of non-constant demand.

Demand patterns for which PPA produces strings of two or more consecutive setups will not be considered, since in those cases there exist optimal solutions with setups in the same time periods as the second, third, etc., of each PPA setup string; thus the demands to be considered are such that each PPA setup is followed by one or more periods of positive inventory holding costs. Also, as PPA does not recommend setups in periods of zero demand, we consider only strictly positive demands (excluding zero).

The worst such demand pattern for PPA is:

$$d_{2k-1} = 1\,, \qquad d_{2k} = \frac{S'}{h}\,(= \text{integer})$$

where S' is the setup cost, h is the unit inventory holding cost, and $k = 1, 2, \cdots, P$. Note that $h < S'$, because (1) there would be a setup in every period if $h > S'$, which we ruled out; and (2) if $h =$

S', the demand is constantly equal to 1, which contradicts our assumption of non-constant demand. The horizon is of length $2P$, with $P \geq 2$.

The above demand pattern is "worst for PPA" in the sense that: (1) the inventory holding costs have been accumulated in the smallest number of time periods (i.e., in P periods), and their totality is equal to the total cost incurred in setups; and (2) the sum total of demands in the PPA setup periods is minimal, i.e., P units.

The PPA setups occur in the time periods $t_k = 2k - 1$, for $k = 1, 2, \cdots, P$. The cost, C, of this solution is:

$$C = PS' + PS' = 2PS'$$

i.e., P setups and P periods where the inventory holding cost in each such period is $(S'/h)h = S'$.

There is an optimal solution, μ, with setups in time periods $\mu_1 = 1$, $\mu_{k+1} = 2k$, for $k = 1, 2, \cdots, P$. The cost, $\hat{C}$, of such a solution will be:

$$\hat{C} = (P + 1)S' + (P - 1)h$$

i.e., $P + 1$ setups and $P - 1$ time periods, where the inventory holding cost in each such period is $1 \cdot h = h$. Thus,

$$C - \hat{C} = 2PS' - [(P + 1)S' + (P - 1)h] = (S' - h)(P - 1)$$

and, as there are P setups in the PPA solution, we have

$$\frac{C - \hat{C}}{P} = (S' - h)\frac{P - 1}{P}$$

This equation gives the upper bound for the extra cost per PPA setup period over the optimal-solution cost for the worst possible case of non-constant demand. This upper bound indicates that PPA provides a comparatively good solution for non-constant demand patterns, especially if PPA's ease of computation is considered.

Comparison with Least Unit Cost

A comparison of PPA with the Least Unit Cost (LUC) algorithm provides some interesting insights into the meaning of the performance results shown in Part I of this paper.

We let S be the setup cost divided by the unit inventory holding cost, and d_m the demand in the mth time period ($m = 1, \cdots, M$). We assume that $d_m > 0$ for all m. Now assume there is an LUC setup in time period t. Then there will be an LUC setup in period $t + k$ (with no setups between t and $t + k$) for the smallest $k (k = 1, 2, \cdots)$ such that the following two conditions are satisfied:

$$\frac{S + \sum_{j=0}^{k-2} j \cdot d_{t+j}}{\sum_{j=0}^{k-2} d_{t+j}} > \frac{S + \sum_{j=0}^{k-1} j \cdot d_{t+j}}{\sum_{j=0}^{k-1} d_{t+j}} \tag{1}$$

and

$$\frac{S + \sum_{j=0}^{k-1} j \cdot d_{t+j}}{\sum_{j=0}^{k-1} d_{t+j}} \leq \frac{S + \sum_{j=0}^{k} j \cdot d_{t+j}}{\sum_{j=0}^{k} d_{t+j}} . \tag{2}$$

where for $k = 1$:

$$\sum_{j=0}^{k-2} j \cdot d_{t+j} = \sum_{j=0}^{k-2} d_{t+j} \triangleq 0 .$$

We now let

$$N_{k-2} \triangleq S + \sum_{j=0}^{k-2} j \cdot d_{t+j}$$

and

$$D_{k-2} \triangleq \sum_{j=0}^{k-2} d_{t+j}$$

From (1) we obtain

$$(k - 1)D_{k-2} < N_{k-2}$$

and from (2)

$$N_{k-2} \leq kD_{k-2} + d_{t+k-1}$$

Thus

$$(k - 1)D_{k-2} < N_{k-2} \leq kD_{k-2} + d_{t+k-1} \tag{3}$$

But as

$$N_{k-2} = S + (k - 2)D_{k-2} - \sum_{x=3}^{k} D_{k-x}$$

substituting in (3):

$$(k - 1)D_{k-2} - \left[(k - 2)D_{k-2} - \sum_{x=3}^{k} D_{k-x} \right] < S$$

$$\leq kD_{k-2} + d_{t+k-1} - \left[(k - 2)D_{k-2} - \sum_{x=3}^{k} D_{k-x} \right]$$

and simplifying terms:

$$\sum_{x=2}^{k} D_{k-x} < S \leq \sum_{x=1}^{k} D_{k-x}$$

that is

$$\sum_{j=0}^{k-2} (k - 1 - j)d_{t+j} < S \leq \sum_{j=0}^{k-2} (k - 1 - j)d_{t+j} + \sum_{j=0}^{k-1} d_{t+j} \tag{4}$$

Now assume there is a PPA setup in time period t. There will be a PPA setup in time period $t + k$ (with no setups between t and $t + k$) for the smallest k $(k = 1, 2, \cdots)$ such that:

$$\sum_{j=1}^{k-1} jd_{t+j} \leq S < \sum_{j=1}^{k-1} jd_{t+j} + kd_{t+k} \tag{5}$$

where, for $k = 1$,

$$\sum_{j=1}^{k-1} jd_{t+j} \triangleq 0 \, .$$

Thus, starting with $t = 1$ (assuming that both LUC and PPA have a setup in the first time period), we see that in determining the occurrence of the second setup, the algorithms behave as follows:

The LUC algorithm, in effect (see Equation 4), assigns to each demand an integer weight factor that is inversely proportional to the time span between the present setup period and the period in which the demand occurs; the first demand so weighted is that one occurring in the present setup period (and consequently carrying the largest weight factor), and the last one (with a factor equal to 1) is the one immediately preceding the next recommended setup. After each intermediate iteration, the weight factors are increased by 1.

The PPA algorithm assigns (see Equation 5) integer weight factors that are equal to the time span between the present setup period and the period in which the demand occurs; thus the demand occurring in the time period of the present setup has a factor of 0 assigned to it, and the last demand (with consequently the largest factor) to be so treated is the one occurring in the period of the next recommended setup. Once a demand has been assigned a weight factor, the assignment becomes permanent through the rest of the computations.

CITED REFERENCES

1. H. M. Wagner and T. M. Whitin, "Dynamic version of the economic lot-size model," *Management Science* **5**, 89–96 (October 1958).
2. *The Production Information and Control System*, E20-0280-1, International Business Machines Corporation, Data Processing Division, White Plains, New York (1967).