As a lot-sizing technique for minimizing the sum of ordering and inventory costs, the algorithm described is based on some simple dimensions. By dividing the ordering and setup costs by the inventory holding costs per part per time period, the ordering costs are expressed in part-periods. This value is used to determine lot size.

First a simplified version is shown for demand sets that do not vary widely between periods. For large variations in demand, significantly greater overall accuracy is achieved with simple look-ahead and look-back tests which are also discussed.

Two of the more important economic lot-sizing algorithms are compared with the Part Period Algorithm.

An economic lot-sizing technique

I The part-period algorithm

by J. J. DeMatteis

The objective of inventory management is to maintain optimum levels of inventory consistent with customer demands and plant capacity. Stated simply, management must determine what to order, when to order, and how much to order. This is not an easy task, for there are many conflicting goals. Nevertheless, management must inevitably make a decision to order what it considers to be an "economic" quantity.

The determination of an economic ordering quantity is commonly referred to as "lot-sizing." Calculations of the size of an order result in minimizing the sum of the ordering costs (including setup, if any) and the inventory holding costs.

The purpose of this part of the paper is to describe a simple economic lot-sizing algorithm and to compare it with two other techniques. In Part II, A. G. Mendoza presents a mathematical analysis of this algorithm.

The algorithm

The part-period concept is based on the following simple consideration: if one part (i.e., one unit) is held in inventory for one period, it incurs a particular holding cost; if it is held two periods, it incurs twice the holding cost. Two parts held one period incur the same cost as one part held two periods, and three parts held two periods incur the same cost as two parts held three periods, etc. If the number of parts held in inventory are multiplied by the number of

periods over which they are held, the dimension "part-period" is derived. Expressing the ordering costs in this new dimension, a simple and effective lot-sizing technique emerges.

By dividing the ordering costs (including setup, if any) by the inventory holding costs per part per period, ordering costs are expressed in part-periods. If, for example, the ordering cost for any number of pieces of a certain part is \$50 and the inventory holding cost is \$0.50 per unit per period, we have a part-period value of 50/0.50, or 100.

The term part-periods, in this instance, is analogous to "man-days." If a particular job costing \$200 for labor may be performed by one laborer working alone, or any number working in various combinations, and if labor costs are \$20/man/day, the value of the job may be expressed in man-days: \$200/20 = 10 men for one day or 10 man-days.

Once ordering costs for each part number are converted to part-periods, this simple calculation need be done again only if there is a change in material and labor costs, setup, inventory holding, etc. A slight modification of the Part Period Algorithm (PPA) permits use of a different holding cost for each period. However, this feature is not considered of sufficient value to incorporate it in the algorithm.

simplified version

A simplified version of part-period will be described first. This is also a necessary part of a more accurate version described later. The simplified version is very effective for all except the most demanding circumstances, and also outperforms the Wagner-Whitin algorithm[1] in the short-horizon environment.

Assume the part-period value of a particular part number to be 100, and the demand by periods to be $d_1, d_2, d_3, \ldots, d_n$. Assume also that no holding costs are incurred for items consumed in the period in which they are ordered. To determine the reorder point and the reorder quantity, we proceed as follows:

$$(0)d_1 + (1)d_2 + (2)d_3 + (3)d_4 + \ldots$$

until the value of this expression exceeds 100. The setup should be made in the period that causes the value to exceed 100. The reorder quantity is then the sum of the demands of the periods covered by the order. A specific example for computing economic lot sizes is shown in Table 1 and now discussed.

Assume the setup costs to be \$50 and the inventory holding cost \$0.50 per part per period. The part-period value of this item will be 50/0.50 or 100. This value is maintained in the part-number record.

A setup is made in Period 1, and we wish to determine the quantity to be manufactured. The demand of 20 units for Period 1 is consumed in the same period in which it is produced and will therefore incur no part-period costs. The demand for Period 2, if ordered in Period 1, is held for one period, incurring a holding cost of 1×20 or 20 part-periods. Therefore, 20 is deducted from the 100 part-periods available for the item, leaving a balance of 80 part-

Table 1 Part-period lot-size calculation

Period	*1*	*2*	*3*	*4*	*(4)*	*5*	*6*	*7*	*8*
Demand	20	20	25	35	(35)	30	10	10	15
Part-Period Value	0	(1 × 20)	(2 × 25)	(3 × 35)	(0)	(1 × 30)	(2 × 10)	(3 × 10)	(4 × 15)
Cumulative Part-Periods	0	20	70	175	(0)	30	50	80	140
Setups	X			X	(X)				X
Cumulative Cost*	50	60	85		135	150	160	175	

Assumptions:
Re-order cost is $50.00
Inventory holding cost is $0.50/unit/period

* Computation of cumulative costs are not required by PPA, however they are shown for comparison with other algorithms described.

periods. The demand for Period 3 is held two periods, incurring a cost of 2×25 or 50 part-periods, leaving a new balance of 30 part-periods. The part-period requirements for Period 4 (which is 3 × 35) exceed the balance of 20 available, thus signaling the need for a setup in Period 4. The first ordering quantity (placed in Period 1) is 65, the sum of the demands for Periods 1, 2, and 3.

A second order is placed in Period 4, and the ordering quantity is to be determined. The demand for Period 5 is 1×30 or 30 part-periods, substantially less than the 100 available. To the 30 part-periods incurred in Period 5, we add 2×10 or 20, the part-periods for Period 6, making a total of 50. Period 7 contributes 3×10 or 30 part-periods for a total of 80, and we note that we have yet to exhaust the supply of part-periods. Finally, we try to satisfy the demand for Period 8 (which is 4 × 15) and find the supply inadequate. That is the signal for a new setup in Period 8. We found the second ordering quantity (placed in Period 4) to be 85, and the number of periods covered are four. In the first setup, 65 pieces were ordered to cover three periods, and in the second setup, 85 pieces were ordered to cover four periods. Thus, it may be seen that part-period is a variable-order-quantity, variable-order-point system which is providing the flexibility required for computing economic lot sizes.

refined version

For demand sets that do not vary widely between periods, the simplified PPA version performs very well. However, where the variation in demand is large, the refined PPA version provides significantly greater overall accuracy and reduces or eliminates a majority of the larger errors that creep into the simple version. This is generally accomplished at the very modest cost of three additional computer steps per planning cycle. In the refined version, PPA goes through exactly the same calculations indicated above for the simple version. However, once the decision to set up has been

Table 2 Look-ahead calculations

Period	*1*	*2*	*3*	*4*	*5*	*6*
Demand	10	90	10	90	10	90
Part-Periods	0	90	20			
Cumul. Part-Periods	0	90	110			
Tentative Setups	X		X		X	
Tentative Cumul. Cost	100	190	290	380	480	570
Final Setups	X			X		
Final Cumul. Cost	100	190	210	310	320	420

Table 3 Look-back calculations

Period	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Demand	10	20	40	10	20	40	10
Part-Periods	0	20	80	0	20	80	
Cumul. Part-Periods	0	20	100	0	20	100	
Tentative Setups	X			X			X
Tentative Cumul. Cost	100	120	200	300	320	400	500
Final Setups	X		X			X	
Final Cumul. Costs	100	120	220	230	270	370	380

made, it "looks ahead" at a minimum of two periods following the tentative setup period to determine if it is faced with a large demand in the immediate future. It also incorporates a "look-back" test, which checks to see if the demand for the previous period is also relatively very large. Thus, the look-ahead and look-back tests avoid the costly mistake of remaining on one level with a mountain of demand at either or both sides. Examples, summarized in Tables 2 and 3, will clarify this.

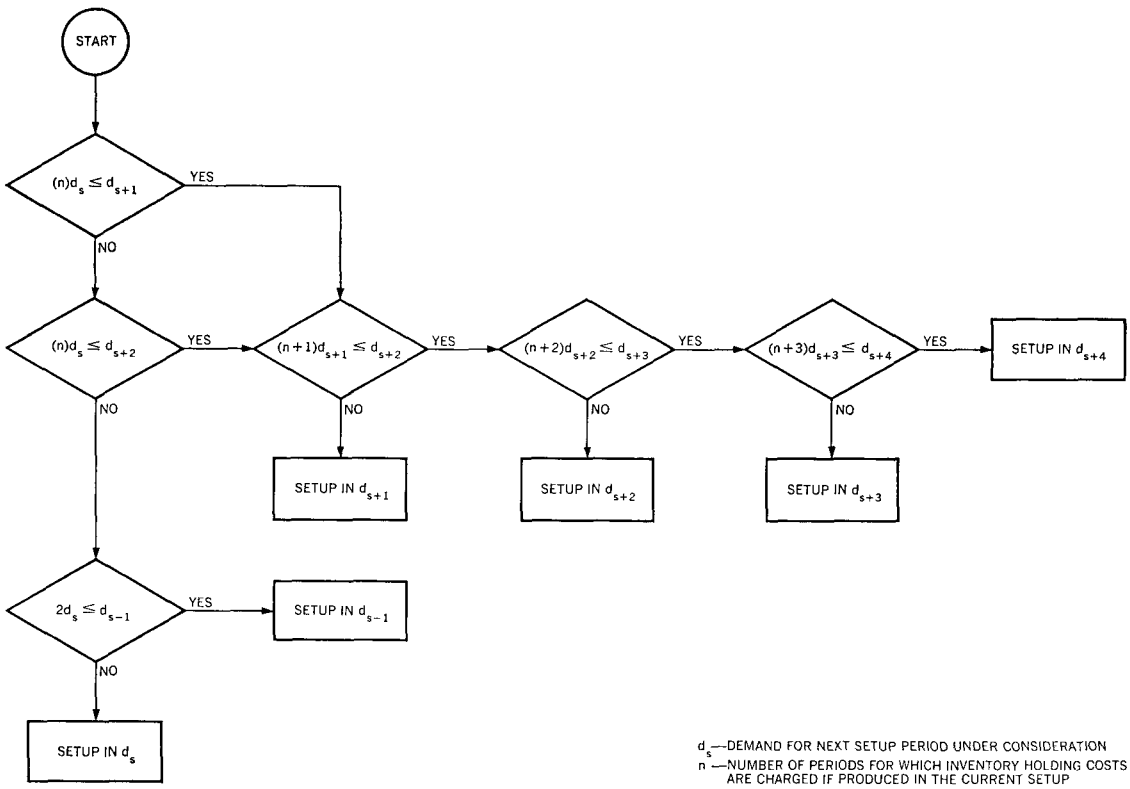
Assume \$100 to be the setup cost and \$1.00 to be the holding cost per item per period. Then, the item part-period value is 100. When the cumulative part-periods exceed 100, a setup is tentatively recommended. Before accepting the decision as final, the look-ahead feature is invoked as follows: The part-period value of Period 3 (namely 20) is compared with the demand for Period 4 (i.e., 90); if it is equal to or less than this demand, the setup period is moved ahead to Period 4. If it is not equal to or less than the demand for Period 4, the original decision stands. In our example, it is clear that the setup should be moved to Period 4.

The look-back test is illustrated in Table 3 (assuming the same setup and holding cost as for the previous example). The demand for tentative setup in Period 4 (d_s) is compared with the demand for the previous period (d_{s-1}) as follows: If

$$2\,d_s \leq d_{s-1}$$

the setup should be in the previous period. In our example, 2×10 is less than 40; therefore, the setup period becomes 3 rather than 4.

Figure 1 Look-ahead and look-back tests



The steps needed for a combination of the look-ahead and look-back tests are illustrated in Figure 1. The look-back test is not invoked if the look-ahead test succeeds in moving the setup to a future period. An optional, final refinement involves a check both ways and, if demand d_s is very low, swings to the higher of the two high values. This is accomplished by comparing d_{s-1} with d_{s+x} and selecting the one with the highest value.

In general, the combination of look-ahead and look-back tests adds only three steps to each setup cycle, with an average of approximately 3.2 steps. Although the tests are not infallible, they add greater precision in the real world of fluctuating demand. Certain combinations of demand also present problems to other algorithms tested, including the Wagner-Whitin algorithm in the short-horizon case.

For a horizon equal to one part-period planning cycle, part-period is invariably optimal. For longer horizons, PPA yields suboptimal results; its precision variance, however, is so low as to be of no consequence in a practical application. The advantages of this simple algorithm are many: it is open-ended; that is, it performs as well over short horizons as over long horizons. It is highly accurate

for demands with large variations between periods, as well as for those with very small variations; it is particularly outstanding in the typical industrial environment where the demand may be known, but only for a maximum of six or seven months. Finally, it is inexpensive and quick to implement and maintain.

Comparison with other algorithms

A large number of economic lot-sizing techniques are available to management, any one or combination of which may be incorporated in an inventory control system. In choosing a lot-sizing technique, the cost of applying the technique and its performance for certain demand characteristics are the major considerations. Low-cost items may be controlled by one technique, and those of higher cost by another. Some of the considerations in choosing a technique, or combination of techniques, are: the cost and time of applying the algorithms; the demand range, i.e., the variation in demand from period to period; the availability of a long-range forecast; and one's confidence in the accuracy of the forecast. Two of the more important economic lot-sizing algorithms are now briefly described and then compared with PPA. These two are generally considered to be more accurate than the classic square-root formulas commonly used.

Wagner-Whitin

The Wagner-Whitin algorithm[1] is an "exact solution" technique for known demands over a horizon equal to the life of the item. Rather than to test every combination of either ordering or not ordering in each period (which would require 2^{n-1} tests), this algorithm is able to achieve the same results in substantially fewer steps. The actual number of steps required varies with the nature of the demand (i.e., the particular manner in which it may fluctuate), the ratio of the setup and inventory holding costs, and the number of periods in the horizon. For short horizons, repeated applications of this algorithm result in solutions inferior to those of PPA.

The major drawback of the Wagner-Whitin algorithm in comparison to PAA is the relatively large number of calculations required in a typical environment. The computational time is extremely sensitive to the frequency of setups. Typically, from ten to thirty times as many calculations are required for Wagner-Whitin as for PPA.

least unit cost

The Least Unit Cost algorithm[2] attempts to compute for various order sizes the costs per unit chargeable to setup and to inventory holding and selects the minimum value. The following demonstrates the procedure followed:

Period:	1	2	3	4
Demand:	20	20	25	35

Setup cost: \$50; carrying costs: \$0.50 unit/period

Step 1

Unit cost for producing only the demand for Period 1 is the setup cost plus zero carrying charges, all divided by the demand for Period 1:

$$\frac{50 + 0}{20} = \$2.50$$

Step 2

Unit cost for producing the demand for Periods 1 and 2 in Period 1 is the setup cost plus the product of the demand for Period 2, the inventory holding cost per period, and the number of periods that inventory is carried—all divided by the sum of the demand for the periods covered:

$$\frac{50 + (20)\ (0.50)\ (1)}{20 + 20} = \$1.50$$

Since the unit cost at Step 2 is less than that of Step 1, an additional step is taken:

Step 3

$$\text{Unit cost} = \frac{50 + 20\ (0.50)\ (1) + 25\ (0.50)\ (2)}{20 + 20 + 25} = \$1.31$$

The cost is still decreasing, so another step is taken:

Step 4

$$\text{Unit cost} = \frac{50 + 20\ (0.50)\ (1) + 25\ (0.50)\ (2) + 35\ (0.50)\ (3)}{20 + 20 + 25 + 35}$$

$$= \$1.38$$

The cost at Step 4 is greater than that derived in Step 3. Therefore, it is assumed that \$1.31 represents the minimum cost, and a quantity of 65 is ordered.

test environment

The lot-sizing algorithms were tested in two different environments. In the first environment, called *known-demand, long-horizon environment,* the customer demand is known for a period of time sufficient to cover three or more production or purchase orders (12 or more months, in our tests), and the demand does not change. Also, no internal attrition (such as scrap, losses, etc.) is permitted. The interval of time covered by a production or purchase order is referred to as a *planning interval* or cycle. This is to be distinguished from the *planning horizon* which encompasses all periods for which a demand is available, that is, for several planning periods, or possibly, for less than one. It is also to be distinguished from the term *repetitive cycle* which involves the placement of an order at fixed time intervals. As the term *planning cycle* is used herein, successive orders may occur at different time intervals.

In the second environment, more realistic than the first, the demand is known for an average of slightly less than two full planning cycles (six months in our tests). This is referred to as the *short-horizon environment.* In this case, it was assumed that lot-sizing would be necessary with each setup to accommodate possible changes in demand during the second planning cycle, although in our tests no changes in demand were actually made. The demand values used in the known-demand, long-horizon case were also used in these tests, but the planning horizon was restricted to six months. Short-horizon tests were made for the Wagner-Whitin algorithm and PPA only. The test results were as follows:

Part-Period algorithm	Lowest cost
Wagner-Whitin algorithm	1.6 percent higher (average of all samples)

A basic assumption is made for a system employing the algorithms described in this paper: a requirements planning subsystem is installed which converts demands for finished products by period to requirements for subassemblies and component parts at all levels, and which nets the component part requirements against inventory on hand and on order. When a requirement exists within the replenishment lead time for a particular item, an indication is made that an order is to be placed.

Table 4 Range groups

Range of demand	*Maximum variation from mean*
01–200	± 100%
01–99	± 100%
10–80	± 75%
25–75	± 50%
30–60	± 33%
40–60	± 20%
45–55	± 10%

performance results

One hundred and two sets of random demand data over twelve to fifteen periods each were grouped by range in demand as shown in Table 4. The distribution about the mean was uniform. Averaging the results obtained for each of the seven demand groups produced the results of Table 5 in which the sums of the setup and inventory holding costs are compared. The table gives an average of all the demand groups, including low as well as high-value items in each group.

The performance of the algorithms, using high-value items with wide variation in demand, is shown in Table 6. These results were obtained in the 01 to 99 group with a maximum variation from the mean of 100 percent, and an average variation of 50 percent.

The performance of the algorithms varied according to (1) range of demand, (2) maximum variation in demand between periods, and (3) the frequency of setup (value of the item). In the known-demand, long-horizon environment, Wagner Whitin always produces a minimum cost solution. PPA is remarkably low in cost and stable

Table 5 Cost comparison for known demand in long horizon environment

Algorithm	*Percentage cost above best performer*
Wagner-Whitin	Lowest cost
Part-Period	0.5%
Least Unit Cost	5.5%

Table 6 Cost comparison for wide variations in demand

Algorithms	*Performance*
Wagner-Whitin	Lowest cost
Part-Period	0.8% higher cost
Least Unit Cost	21.3% higher cost

Table 7 Cost comparison for extreme categories in percentage above minimum cost

	Group 01–200 (Average 5.9 setups/year)	*Group 45–55 (Average 2.8 setups/year)*
Wagner-Whitin	0.0	0.0
Part-Period	0.7	0.0
Least Unit Cost	16.4	0.2

in all categories; however, it, too, is influenced by the demand characteristics. The wide variation in performance is shown by the extreme categories of Table 7.

It is to be emphasized that the performance indicated assumes that the actual demand pattern for the twelve-month horizon remains precisely as originally forecast. If the demand remains the same for the first planning cycle only, the accuracy of the Wagner-Whitin algorithm is affected, whereas the Part Period and Least Unit Cost algorithms are not affected.

Conclusions

The Part Period Algorithm performs well in all environments, but is particularly well suited for industries whose demand forecast extends for a limited number of periods, and for those whose forecast is appreciably more accurate in the near future than for the more distant future. In the short-horizon environment, PPA outperforms the other algorithms tested. In addition, considerably few computations are required by the Part Period Algorithm than of the other tested algorithms with comparable performance. In the "known-demand, long-horizon" environment, setup and holding-cost performance with PPA is, on the average, approximately half of one percent higher than minimum cost.

CITED REFERENCE AND FOOTNOTE

1. For a complete description of the Wagner-Whitin algorithm see: H. M. Wagner and T. M. Whitin, "Dynamic version of the economic lot size model," *Management Science* **5**, No. 1 (October 1958).
2. *The Production Information and Control System*, E20-0280-1, International Business Machines Corporation, Data Processing Division, White Plains, New York.