A CASE HISTORY IN OPTIMUM INVENTORY SCHEDULING

George Schussel

Informatics Inc., River Edge, New Jersey

and

Steve Price

Northrop Corporation, Hawthorne, California

(Received August 20, 1969)

This paper describes four computer programs that perform what were formerly management-decision functions in inventory scheduling. The first program solves the classical EOQ problem (uniform inventory usage) with quantity discounts. The second program, Economic Requirement Batching (ERB), uses dynamic programming, a heuristic search algorithm, and the relation between shipment size and material cost to locate the optimum delivery schedule for any deterministic schedule of discrete (irregular) requirements. The third program, the Alternative Delivery Schedule Evaluator (ADSE), compares any alternative delivery schedules that meet requirements; it calculates all costs associated with inventory and displays both the optimum alternative and the opportunity losses incurred by inferior alternatives. The fourth program, the Alternative Delivery Schedule Generator (ADSG), solves the difficult problem of optimally scheduling inventory in those cases requiring vendor production to special specifications and, accordingly, where the exact price of the item is unknown and depends on the manner in which it is produced. This situation is differentiated from ERB because ERB analyzes standard vendor shelf items that can be delivered according to any schedule. With little input data, ADSG generates a few highly efficient alternative delivery schedules upon which the vendor quotes. The returned bids are then evaluated by ADSE, which determines the optimum delivery schedule. Since over three years have elapsed between when this work was done and the publication of this article, it concludes with a view of the program's successes and failures and some relevant empirically observed relations.

THE PURPOSE of this paper is to discuss how some recently developed techniques in operations research were combined with recently developed techniques in computer processing to provide a company's management with important tools for improving inventory control. The research described herein is not sophisticated from the mathematical point of view and, as a matter of fact, borrows very heavily from work already published in operations-research literature. The important point stressed by this paper is the interrelation of several different techniques that enabled the project to be successful.

An optimization technique using dynamic programming was essential in solving the discrete requirements case, because both a continuous approximation to the discrete case and an exhaustive enumeration of different policies were impossible. Only the technique of dynamic programming could have been used to solve this problem mathematically. However, dynamic programming was not sufficient. The assumptions that were made in order to allow a dynamic programming solution were still too restrictive and not representative enough of the actual situation for the solution to be useful. Heuristic modification to this optimal solution, however, enabled the development of an optimization technique that could do an effective job in solving the discrete requirements case.

In addition to the operations-research techniques, the availability of a time-shared console was extremely important in both development and production use of the computer programs necessary to solve the problems. It is quite possible that, without the time-shared approach to the problem, it would either have not been solved, or the implementation procedure would have been considered too unwieldy to be useful, since much of the information and the solutions proposed by the computer programs were needed on a real-time basis.

BACKGROUND

THE WORK that is described below was performed at Northrop Corporation, a medium-sized ($450 million sales) aerospace-defense firm, by the Corporate Management Sciences Staff on a consulting capacity for the Norair Division. The analyses and programs were developed in a very general vein, however, and are applicable to the general problem of ordering from outside vendors for any industry.

The Management Sciences Staff received a request from the Norair Division to assist divisional personnel in developing decision aids for inventory ordering problems. The divisional personnel were familiar with the use of automated techniques of data storage and retrieval in the inventory function, since they had been using several computer-based systems for storing and retrieving much of the basic data necessary for the inventory function. The available computerized systems assisted management by providing various levels of information, but offered no aid in decision making. With the receipt of a major new commercial contract, the division had come under a substantial amount of pressure to lower the levels of inventory that were maintained in raw material and work-in-process stocks. The general rule of thumb in the materiel operation had been to take quantity discounts whenever these were offered by a vendor. Generally, discounts were offered if Northrop was willing to take early deliveries on purchased items rather than wait to accept the delivery of these items at their need dates. Under a tight money situation created by corporate expansion, the materiel personnel were told they should not take many of these discounts, and should, accordingly, cut down the level of their inventories. Therefore, the materiel department was in a quandary as to when they should accept a material discount and early shipment, or when to insist upon shipment according to requirements and turn down discounts offered by vendors.

When the Management Sciences Staff was asked to consult on this problem, it made a survey of the relevant literature and computer systems in order to determine what prior work had been done in this area. The available literature on inventory theory is vast; a few examples are mentioned in the bibliography at the end of this article. Many of the theoretical investigations into inventory problems have not been implemented, however, and the number of computer programs available to help solve inventory ordering problems is not overwhelming. One of the best applications-oriented packages was the IMPACT System from IBM.[7] This system consisted of a group of IBM 1401 programs that were designed to perform data storage and retrieval and computational assistance in the inventory purchasing area. From the operations-research point of view, the IMPACT System is centered around the concept of EOQ. This reliance on EOQ is extremely limiting; because of this and the fact that the IBM 1401 was obsolete when the research was begun, a decision was made to undertake an independent effort.

After several conversations with materiel personnel, it was determined that tradeoff and operations-research analyses in the inventory area would require several different models in order to evaluate the substantially different problems that the materiel people were called upon to solve. Since the conditions in some of the problems were so different from the conditions and assumptions in others, it was felt that no one single model or computer program should be built to solve all the inventory-ordering problems. A more efficient approach of dividing the general inventory ordering problem into several submodels and solving each one of these was decided upon.

The models discussed in this article are only concerned with the development of an optimum delivery schedule. They do not determine the optimum size of a total commitment between the vendor and purchaser, a problem that requires a balance between the cost of frequently negotiating many small contracts against the cost of purchasing to meet a long but uncertain schedule of future requirements and that can be analyzed in decision-theoretic terms.

In many situations, the over-all commitment between the vendor and purchaser is not as important as the actual delivery schedule because of provisions in the contract or the common law for the relatively simple breaking of a contract when it does not result in substantial harm to either party. To illustrate, a long-term commitment to purchase 1,000 units may not be as binding contractually as smaller specific orders, resulting in production, that are placed against this commitment. Therefore, optimally solving the subproblem of determining a delivery schedule by balancing price and inventory-holding-cost considerations can be very important to a firm.

Because of the availability of a time-shared computer system (IBM's RAX), it was decided to develop the programs required to implement inventory-problem solutions on the time-shared system. The almost instantaneous turn-around that a time-shared computer system offered made it simpler for the analyst to do the programming himself, thus eliminating the need for coordination with a programmer; as a result, the use of the time-shared system greatly speeded the development effort.

Even though it was initially felt that the programs would be developed in a time-shared environment and later run in a batch-oriented mode, the quick turn-around time offered by the time-shared system turned out to be a distinct advantage in the actual production use of the finished programs.

This advantage resulted because all of the developed programs were meant to help materiel personnel in making decisions on how to purchase and schedule inventory optimally. The availability of the cost information provided by these programs was often needed on a real-time basis. Access to the processing capabilities of these programs on a time-shared basis and the associated instant turn-around enabled the operations-research techniques in the computer programs to be applied on a real-time basis in negotiations with vendors. This access was found to be of critical importance to management.

ALTERNATIVE DELIVERY-SCHEDULE EVALUATOR (ADSE)

One basic problem was comparing alternative delivery schedules that satisfied the requirements of the purchases. Typically, there might be an alternative that provided for the delivery of all requirements on an order by the date of the first requirement; this alternative usually had the lowest total material cost. The opposite alternative might consist of a separate delivery from the vendor for each individual requirement of the purchaser; this delivery alternative usually had the highest material cost.

It was determined that a very useful tool might be a computer program that would evaluate the costs associated with each alternative delivery schedule and rank the alternatives by a total-cost criterion. Thus, the alternative delivery-schedule evaluator program (ADSE) was written; it took any given number of different alternatives for meeting any schedule of requirements, uniform or not, and simply traced through the various costs such as material, receiving, holding, etc., for each alternative. The final result was a listing of the alternatives and the total associated cost for each one.

This program turned out to be a very useful management decision-making tool in itself, because the calculations usually associated with tracking the holding cost for a fluctuating inventory were such that no one was willing to undertake them manually. The simplicity of having a computer program available to track these various costs enabled management to evaluate different alternatives where previously only a guess had been made as to which was the best course of action.

In addition, however, it was felt that the development of such a tool would prove very useful in prompting further operations-research efforts in the inventory-ordering area. For example, if a technique could be designed to find an optimum delivery schedule under a given set of conditions, it would be very easy to test this optimality against criticisms offered by nonbelievers by suggesting that alternative delivery schedules be determined that would improve upon the one suggested by the operations-research study. The availability of ADSE would then make it a very simple task to compare the alternative delivery schedules and show very quickly which one was better.

For exactly this reason, it was felt that the materiel personnel should have substantial confidence in the calculations performed by the ADSE program. Accordingly, in its first stages the ADSE program printed out most of the calculations that it made; this consisted of tracking the holding, material, and other relevant costs through time. Various people in the materiel operation then traced through the calculations on a few individual problems and became convinced of the accuracy of the ADSE program. With this confidence in ADSE, the rest of the work went much more simply.

The approach taken by ADSE, as in the other three programs, was to minimize total inventory-associated costs such as holding, material, receiving, and any other applicable costs (such as penalties that may be assigned to a vendor who is always late).

The input for ADSE consisted of the following five basic items:

1. Receiving cost in dollars per shipment.
2. Holding cost as a percentage on an annual basis.
3. Flow time through the materiel facility. (This flow time was the maximum

required and was treated as a constant difference between the dock date and the date material would become available at the place where it would be required.)

4. Schedule of requirements, which consisted of a series of dates with the quantity required at each date.

5. A list of the various alternative delivery schedules—each delivery schedule defined by a list of dock dates, the quantity for each one of these dates, the unit cost for each shipment, and a list of other costs to handle miscellaneous items.

The program output consisted of a summary of results from the evaluation and was essentially a listing of the various alternatives and the costs associated with each. The best alternative was identified and the opportunity loss associated with taking any alternative other than the best was also computed.

ECONOMIC REQUIREMENT BATCHING (ERB)

SINCE IT was determined that most of the material ordered from vendors was required to support job-shop production on a repetitive, staggered basis, most vendor-procured items did not have a meaningful average annual usage rate. The requirements for one item might be 100 per year; but, in reality, this might be in two lots of 50 pieces to support two releases in a job shop. Because of its reliance on the assumption of uniform usage, the EOQ concept was determined to be not useful as a model to perform calculations for optimizing inventory ordering policy under staggered usage.

Economic requirement batching (ERB) is a computer program designed to find the optimum delivery schedule for problems of this type, where the unit cost is a constant or a decreasing step function of shipment size. ERB begins by reading the input data for a problem—a quantity-discount schedule, a schedule of requirements, the cost of receiving an extra shipment (including set-up charges, if any), the inventory-holding–cost rate and the maximum probable flow time from the receiving dock to the place where the material is required. Since discounts may be available, ERB's total-cost function also includes the cost of material. ERB is based on a dynamic programming model similar to the one developed by WAGNER AND WHITIN[9] entitled "The Dynamic Deterministic Lot Size Model." This same model is presented by HADLEY AND WHITIN,[5] pages 336–345, and Hadley,[4] pages 386–390.

Table II can be used to illustrate the function performed by the Wagner-Whitin model.

Given the receiving-cost rate ($/shipment), holding-cost rate ($/$ worth of inventory/year) and the cost of the material ($/unit), one could easily calculate total cost for each of the four alternatives in the preceding table. However, if the requirement schedule were expanded to 10 pairs of dates and quantities, such an exhaustive combinatorial analysis would require the evaluation of 512 alternatives. [In general, with the definitions given in Table I it is not difficult to show that $n+1 \leqq W \leqq (n^2+3n+2)/2$ and $E=\sum_{r=0}^{r=n} n!/(n-r)!r!=2^n$.] The Wagner-Whitin algorithm would require no more than 55 such calculations and might complete the task after just 10 calculations.

The dynamic programming solution depends upon optimal policy

TABLE I

DEFINITIONS

a	When used as a subscript, refers to after-income-tax cost
b	When used as a subscript, refers to before-income-tax cost
C	Purchase price per unit from vendor ($/unit)
D_i	List of dates at which specific requirements for a vendor-supplied item exists
E	Number of cost calculations required to determine an optimum delivery schedule if an exhaustive combinatorial search is performed
H	Annual holding-cost rate for inventory (%/year/100)
K	Cost of capital, after tax or the cutoff rate used for capital investment decisions (%/year/100)
n	Number of intervals between discrete requirements in a delivery schedule—one less than the total number of different (D_i, Q) requirements for the same product
P	Proportion of progress payment tied to date material is received
Q	Order quantity (units/shipment); with a subscript E, this is the optimum order quantity
R	Receiving cost associated with any particular shipment ($/shipment)
s	Trigger level in trigger-type inventory ordering system
S	Desired inventory level in trigger system
T	Total cost of purchasing, receiving, and holding inventory in any commodity per year ($/year)
t	Income-tax rate (%/100)
U	The usage rate for any commodity (units/year)
W	Number of cost calculations required with the Wagner-Whitin dynamic programming algorithm to determine the optimum delivery schedule

iteration through successive states and is based upon the principle of optimality. This principle states that an optimum policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. This is applied to the inventory problem by stating that, if it is optimal to have an order arrive at the beginning of period j when there are only j periods and the on-hand inventory is to be zero at the end of period j, then it will be optimal to have an order arrive at the beginning of period j regardless of how many additional periods there are.

When unit cost is a decreasing step function of shipment size (i.e., there is a quantity-discount schedule), the Wagner-Whitin model is not entirely adequate, although it can provide a good starting point in the search for the optimum delivery schedule. If, however, the cost per unit for material does not depend upon shipment size, the Wagner-Whitin model is very efficient in locating the optimum delivery schedule.

The optimum delivery schedule is the one with the least total cost, where total cost can be defined as the cost of receiving shipments plus the cost of holding inventory. Total cost need not include the cost of material because this model is based on the assumption that unit cost is not affected by the choice of a delivery schedule (i.e., price is constant).

When total material cost is a concave function of shipment size, the total cost function in the Wagner-Whitin model can be expanded to include material cost and an optimal solution is still assured. Figure 1 illustrates

TABLE II

AN ILLUSTRATION OF THE FUNCTION PERFORMED BY THE WAGNER-WHITIN MODEL

Schedule of requirements		Exhaustive list of reasonable alternative ways to batch requirements for delivery			
Date	Quantity	1	2	3	4
D_1	Q_1	Q_1	Q_1+Q_2	Q_1	$Q_1+Q_2+Q_3$
D_2	Q_2	Q_2		Q_2+Q_3	
D_3	Q_3	Q_3	Q_3		
Number of shipments		3	2	2	1

such a function. With x defined for all positive integers, the curve $c(x)$ is concave only if the line segment joining any two points $[x_1, c(x_1)]$, $[x_2, c[(x_2)]$ lies entirely on or below the curve $c(x)$, for $x_1 \leqq x \leqq x_2$. A more general definition of a concave function can be found on page 83 of reference 4.

An essential requirement of the Wagner-Whitin model is that the material cost function be concave. When this function is not concave, the model cannot guarantee an optimum solution. Figure 2 is a sketch of a function that is not concave; note the line segment that lies partially above $c(x)$.

Unfortunately, cost functions of this form are not at all unusual. The only property found to be common to Northrop's material-cost functions was that average unit cost $C(x)/x$ did not increase with shipment size. This property can be seen in Fig. 2 by noting that $C(x)/x$ is the slope of a line connecting $C(x)$ with the origin and this slope continues to decrease or remain constant as x increases.

The Wagner-Whitin model is inadequate for this form of cost function because it treats the quantity for each requirement as an inseparable unit. That is, if ten units are required on day D_1 and 90 units are required on day D_2, the Wagner-Whitin model will consider one shipment of 100 units or two shipments of 10 and 90 units; it will not consider two shipments of 20 and 80 units or 30 and 70 units.

Figure 3 can be used to demonstrate why a requirement should not be treated as an inseparable unit. Given requirements for Q_1 units on day D_1 and Q_2 units on day D_2, it can be seen from this figure that, if two shipments of Q_1 and Q_2 units are more economical than one shipment of Q_1+Q_2 units, then it may be even more economical to increase the size of the first ship-

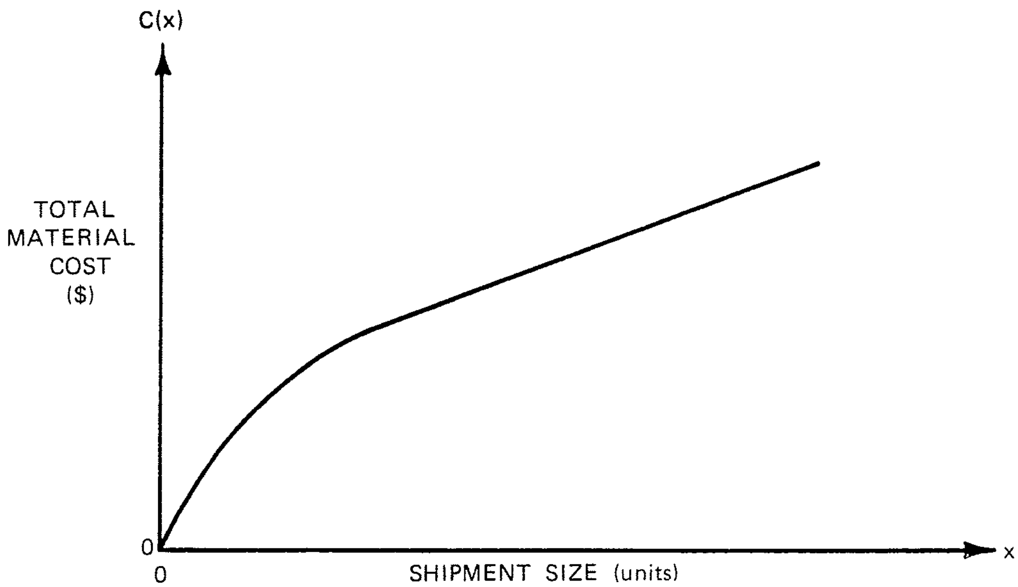

Fig. 1. Continuous material-cost function (completely concave downward).

ment to $Q_1+\Delta x$ units and reduce the size of the second shipment to $Q_2-\Delta x$ units. In other words, it may be optimum to split the second requirement into two subrequirements (Δx and $Q_2-\Delta x$) so that Δx units can be scheduled to arrive with the first shipment on day D_1 and the remaining $Q_2-\Delta x$ units can be scheduled to arrive on day D_2, the need date for the second requirement. Clearly this split will be economical when the net reduction in material cost $\Delta C_1-\Delta C_2$ is greater than the additional holding cost.

This problem is caused by the fact that the slope of $C(x)$ does not continue to decrease as x increases. Thus, it is possible for ΔC_2 to be greater than ΔC_1. However, if $C(x)$ did have a slope that continued to decrease as x increased, then $C(x)$ would be a concave function and ΔC_2 could never be greater than ΔC_1, so it would never be economical to split a requirement and the Wagner-Whitin model could be used to guarantee an optimum solution.

Economic Requirement Batching (ERB) compensates for this problem by splitting large requirements into a number of subrequirements. ERB was designed to locate the largest quantity in the schedule of requirements and restate it as two requirements, each with the same date but half the quantity of the original requirement. This process of chopping up requirements is continued until the internal computer storage capacity limitations are reached. For example, one requirement for 100 units could be treated as four requirements of 25 units each, separated by intervals of zero days and, therefore, by the seemingly illogical trick of treating one requirement on a given day as several different requirements for the same product on the

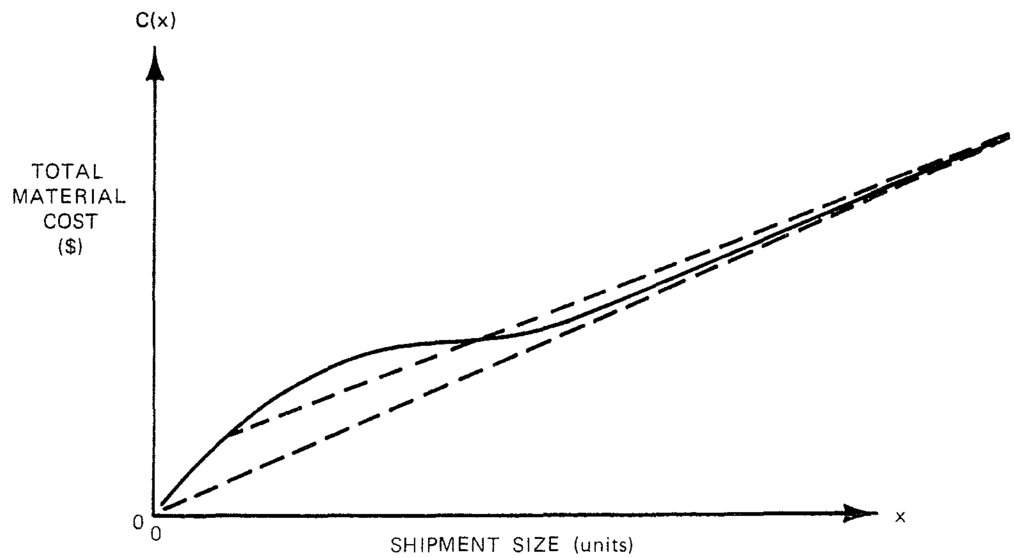

Fig. 2. Continuous material-cost function (not completely concave downward).

same day, the mathematical structure of the problem is retained so that the dynamic programming technique still functions and guarantees an optimum solution to the problem. During this process, the first requirement is not considered as a candidate for change and quantities are maintained as integers greater than zero.

Straightforward application of the Wagner-Whitin model was made more difficult by the particular form of the function relating material cost to shipment size. Most vendors provide quantity-discount schedules in which the average unit cost is a decreasing step function of shipment size. This form of discount schedule is preferred by vendors because it is simple to implement and readily understood by clerical personnel. One typical result, however, is that the cost function assumes illogical portions when the shipment size passes a price break in unit cost. The purchaser actually pays less money for more units.

Figure 4 is a sketch of such a material-cost function. Note that in addition to containing illogical intervals, this function does not appear to be concave.

As a result of this situation, in order to have an effective tool for cor-

DEFINITIONS

$C(x)$	=	TOTAL MATERIAL COST FOR A SHIPMENT OF x UNITS ($)
Q_i	=	SIZE OF THE iTH SHIPMENT (units)
$Q_{E,i}$	=	OPTIMUM SIZE OF THE iTH SHIPMENT (units)
Δx	=	NUMBER OF UNITS REMOVED FROM THE SECOND SHIPMENT AND ADDED TO THE FIRST SHIPMENT (units)
ΔC_1	=	INCREASE IN COST OF THE FIRST SHIPMENT ($)
ΔC_2	=	REDUCTION IN COST OF THE SECOND SHIPMENT ($)

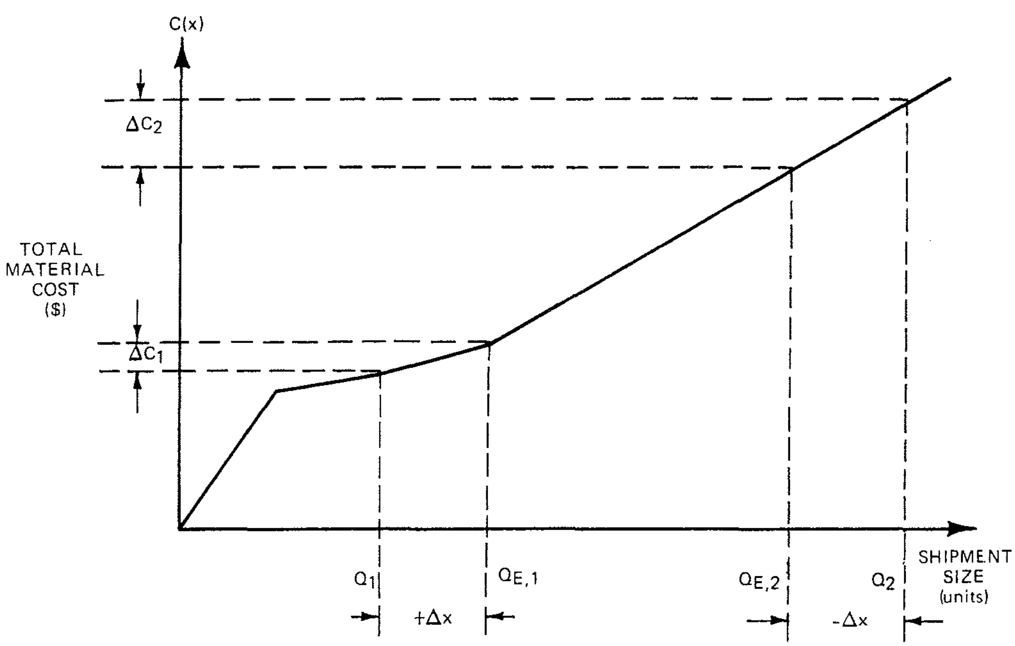

Fig. 3. Piecewise continuous material-cost function (not completely concave downward).

rectly evaluating the discrete inventory-ordering problem, it was necessary to modify the basic Wagner-Whitin algorithm to compare the material cost for each potential shipment with the cost of buying enough extra units to make the next price break. The extra units are scheduled whenever it is economical to do so and the ERB algorithm compensates for the effects these extra units have on successive decisions in the scheduling process.

The output from this modified version of the Wagner-Whitin model

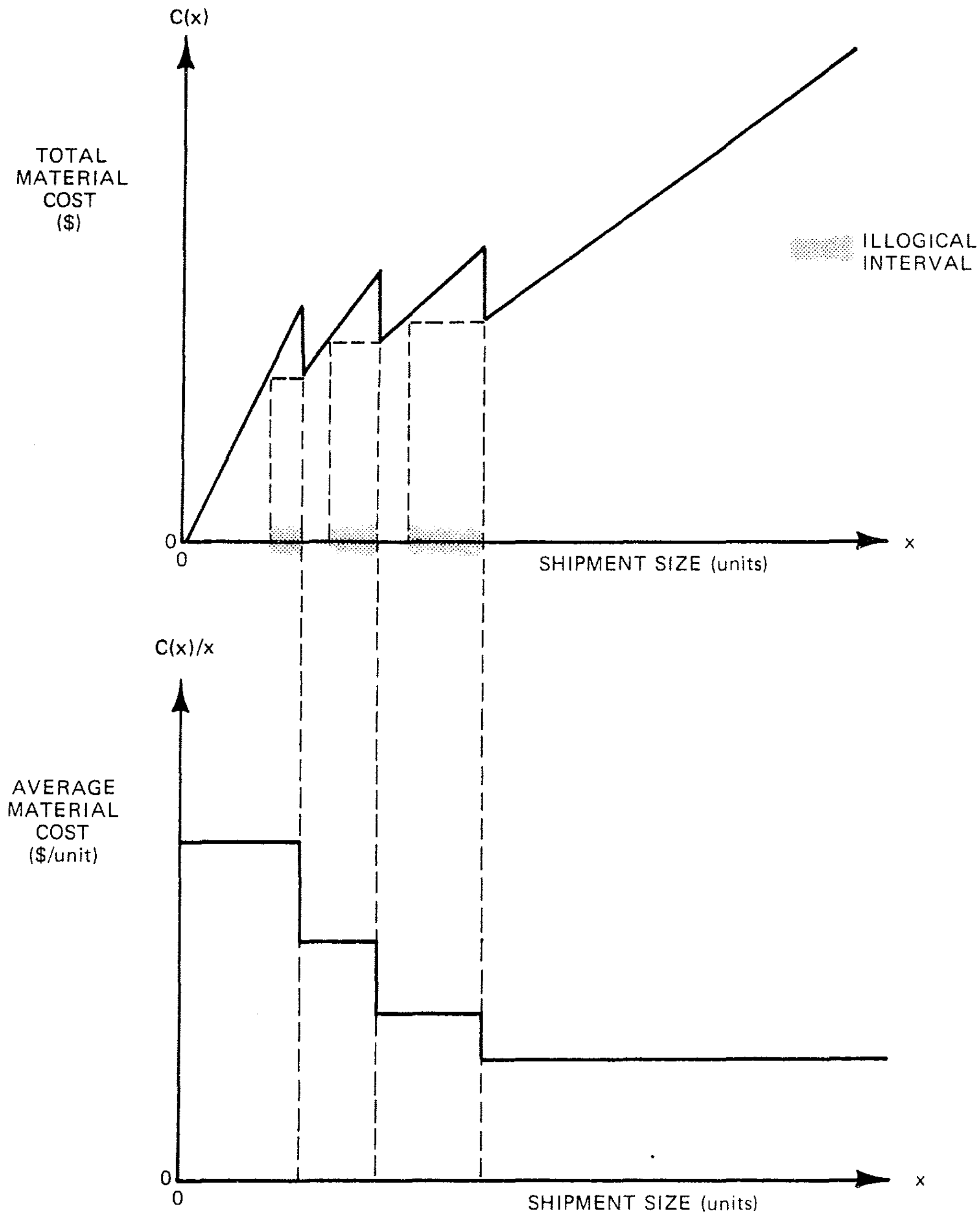

Fig. 4. Discontinuous material-cost function (inherent when quantity discount schedules are used).

is not necessarily optimum because, typically, each requirement is only divided into two or three subrequirements. However, the resulting delivery schedule is usually close enough to be optimized by a heuristic adjustment that generates and evaluates proposals for changes in the delivery schedule. When the evaluation of a proposal indicates that it will lead to a reduction in total cost, the delivery schedule is changed to the improved version. Although the delivery schedules thus obtained are not guaranteed to be optimum, they were found to be highly efficient, and in all cases explored no more economical solutions were found.

The inputs for this program are very similar to those in ADSE and consist of:

1. A receiving cost per shipment (i.e., receiving cost plus any setup charges).
2. The holding-cost percentage.
3. A flow time from the dock to the place required.
4. A schedule of requirements.
5. The quantity-discount schedule, listing intervals of units per shipment and the associated price per unit for each interval.

To summarize, ERB was developed as a program that would find an optimum delivery schedule to meet any schedule of requirements for a part that experienced intermittent usage and was subject to quantity discounts. With ERB, it became possible to develop optimum delivery schedules for any purchased material whose price was constant or varied only with changes in shipment size.

ALTERNATIVE DELIVERY-SCHEDULE GENERATOR (ADSG)

AT FIRST it was thought that the combination of ADSE and ERB would be adequate for solving most problems associated with intermittent usage. It very readily became apparent, however, that, as general as ERB seemed to be, it was not general enough to solve a problem situation that arose in a large number of cases. The critical assumption inherent in ERB was that the vendor's price varied only with the quantity in each shipment. In many purchase contracts, especially those of high-value items, this turned out to be an incorrect assumption. ERB only performed well for products with relatively well defined price schedules. These could be either stock vendor shelf items or items built to special specifications that only required relatively short production runs. When a vendor's production rate is rather low and his associated run time long for a special-specification item, the price associated with the unit often depends on the structure of the delivery schedule because the vendor uses the delivery schedule as a basis for his production schedule.

Typically, in this situation, there was no knowledge available concerning an exact price of the ordered material, and in fact, what was needed was a group of reasonable delivery schedules that could be given to the vendor to bid on. Each of these alternative delivery schedules would have meaning to the vendor; he would be able to deduce the number of setups required in order to meet any schedule and accordingly determine the total cost for each schedule. If a technique was determined for generating an efficient set of alternatives, then it would be possible to have the vendor bid on each of these alternatives and use the Alternative Delivery-Schedule Evaluator (ADSE) to pick the best one for the Norair Division. Therefore, the third program that was developed had the objective of determining a small but highly efficient set of alternative delivery schedules such that we could be reasonably certain that one of them would be optimum for the purchaser.

The information input required for ADSG was:

1. Receiving cost—dollars per shipment.
2. Holding cost percentage.
3. Flow time from dock to the place required.
4. Required shipment quantity in units per shipment.
5. The required shipment interval in days between shipments.
6. The approximate unit value.
7. Schedule of Norair requirements.

Input requirements 4 and 5 were usually available in this type of situation. These could be obtained from the vendor and were usually phrased in the following sense: "While I am manufacturing these units, I will ship 100 at a time at intervals of one month between shipments." The approximate unit value was just a rough figure to enable the program to perform estimates on holding costs. The ADSG program used heuristic logic to develop a list of schedules from these data inputs. This logic was rather complex and required 1,300 lines of FORTRAN programming to implement, but it is roughly outlined below.

The flow chart in Fig. 5 outlines the basic logic and subroutines for implementing the ADSG scheduling algorithm while the chart in Fig. 6 provides a convenient tool for graphically visualizing its operation.

After loading the input data, the required shipment size and interval are used to calculate an implied production rate for the vendor's manufacturing facility. Subroutine SCHED then proceeds backwards in time through the schedule of requirements scheduling deliveries to arrive as late as possible. This is done with no restriction on shipment size but in strict accordance with the implied production rate. In other words, a delivery of Q units ties up the production facility for $Q \cdot P$ days, where P is the production rate in days per unit. When the production facility becomes free, as the date is shifted back in time, cumulative requirements are compared to cumulative deliveries and a shipment, corresponding to the difference, is scheduled if cumulative deliveries are less than cumulative requirements. The purpose of subroutine SCHED is to find the quantity and latest possible start date for each production run.

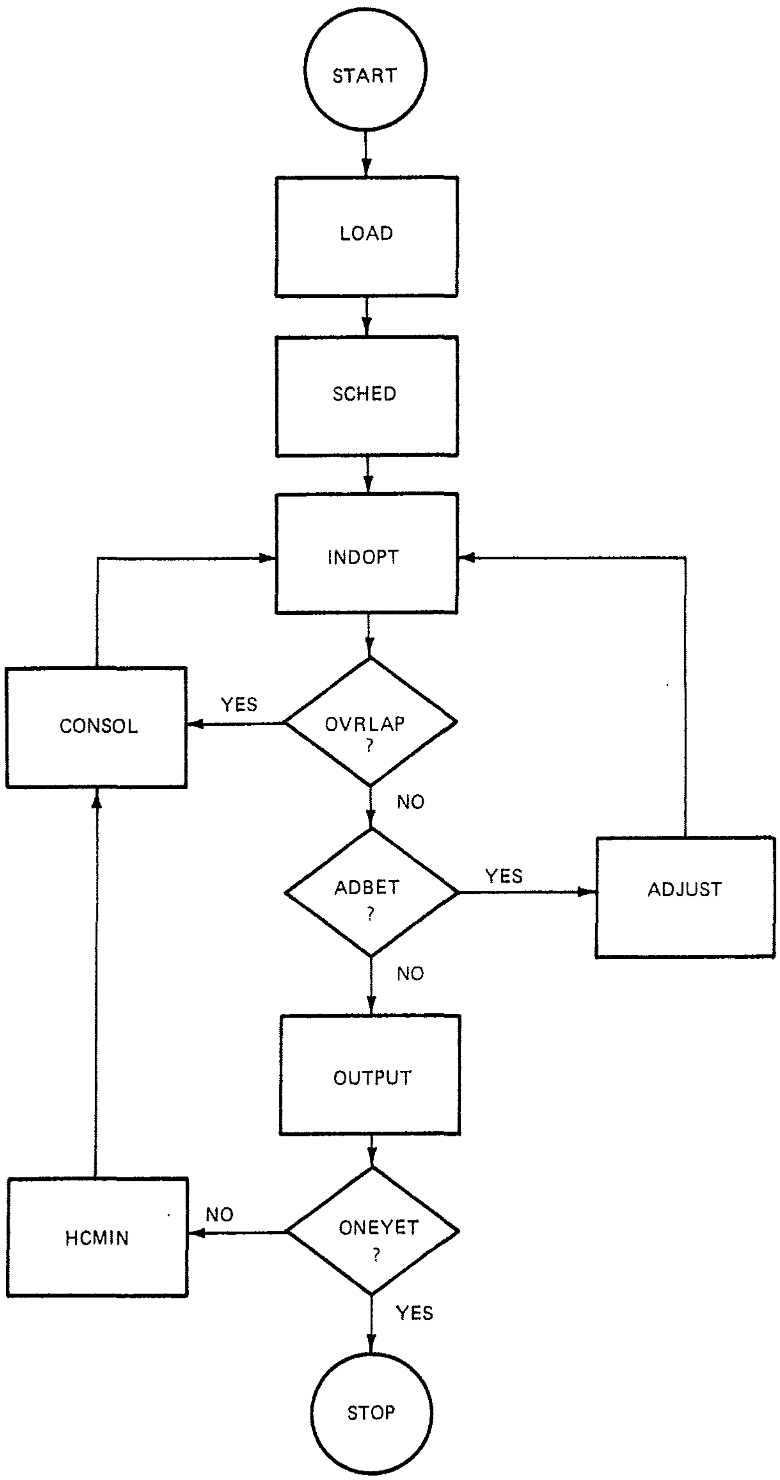

Fig. 5. Logic flow for alternative delivery-schedule generator.

Subroutine INDOPT is then called for each production run. The function of INDOPT is to optimize independently the delivery schedule for one production run. This is done by varying the size of the first shipment over as many as twenty different values and then shifting the sequence of deliveries as late as possible for each value. The delivery date and quantity that minimizes total cost is optimum. The deliveries treated by INDOPT must all conform to the required shipment size unless they happen to be the first or last from a production run. Figure 6 presents a typical schedule for this point in the program.

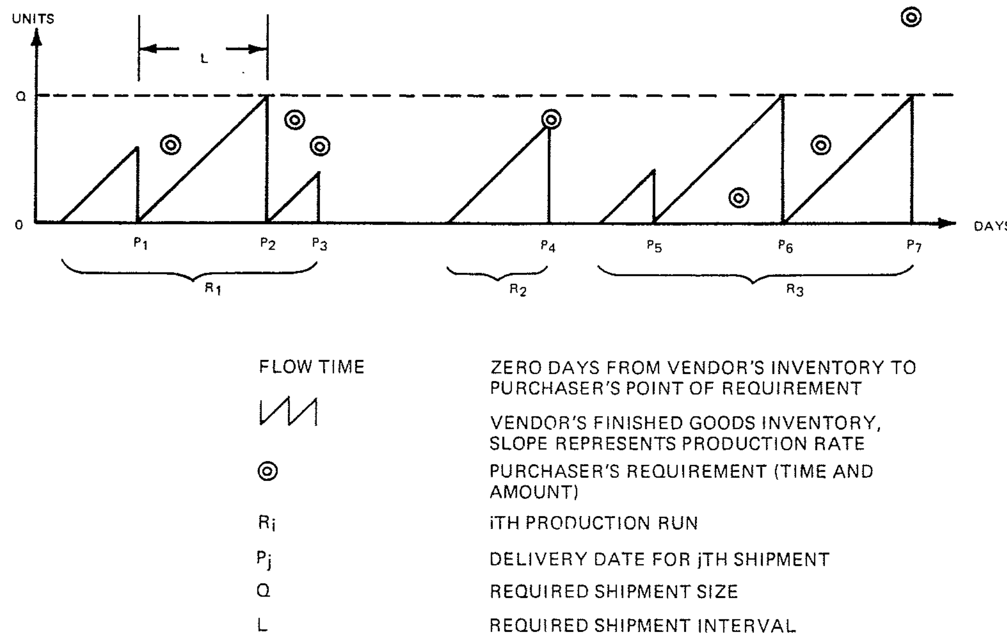

Fig. 6. Example of a feasible schedule created by the alternative delivery-schedule generator.

Occasionally, the shifting operation performed by INDOPT will create some overlap between closely adjacent production runs. Subroutine OVRLAP conducts a quick test for this condition. When overlap is found, subroutine CONSOL consolidates the adjacent runs. The enlarged production run is then independently optimized before returning to the beginning of the overlap test.

When all overlap has been eliminated, subroutine ADBET searches for opportunities to reduce total cost by adjusting terminal partials between production runs. For example, the partial shipment in the third production run shown in Fig. 6 (P_5) could be distributed among the partials in the two preceding runs (P_3, P_4). This would increase the purchaser's holding cost but eliminate the cost of receiving one shipment. If the net effect would be a cost reduction, then subroutine ADJUST would be called to implement the adjustment, and then INDOPT would be called to independently optimize the deliveries affected before returning to the beginning of the overlap test.

Subroutine OUTPUT is reached when all overlap has been eliminated, all runs are independently optimum, and there are no more opportunities for economical adjustment of terminal partials between the delivery schedules for the various production runs. When OUTPUT has completed printing an alternative, ONE YET asks whether we have reached the point where there is only one production run. If 'yes,' then the problem is complete. If 'no,' then HCMIN is called to locate the production run that could be shifted back in time, to combine with its immediate predecessor, for the smallest possible increase in the purchaser's holding cost.

CONSOL then consolidates the production runs chosen by HCMIN. Again, the enlarged production run is independently optimized before returning to the beginning of the overlap test.

The output from the program performing the above logical manipulation is a small number, usually 2 to 5, of alternative delivery schedules. These schedules, along with any others that may be determined either by the vendor or the purchaser (using either formal or informal methods), are bid upon by the vendor. The results from this bidding are then input to ADSE which prints out the best of all of the alternatives and the opportunity loss accruing to each of the other alternatives. Therefore, the combination of ADSG and ADSE results in a powerful package that enables a rather general inventory-cost analysis to be made in the very difficult situation where the price of a unit is not exactly known and depends on how the vendor produces the shipments.

ECONOMIC ORDER QUANTITY (EOQ)

THE MOST commonly described inventory model is the basic (s, S)—oriented analysis of inventory ordering, economic order quantity. One of the strongest assumptions inherent in the analysis of this model is a relatively uniform usage of inventory after it is acquired. While it was determined that this assumption was too restrictive for production-oriented items purchased at Norair, there was, nonetheless, one area of applicability for the EOQ model. This area encompassed those parts in 'Industrial Supplies and Equipment' typified by items such as nuts and bolts, bar and sheet stock, and light bulbs. Accordingly, there was some use for an EOQ model and an analysis was made that resulted in a short computer program that performed EOQ calculations. The trigger level of the (s, S) policy was determined by an analysis of historical patterns of usage in combination with desired safety levels. The reorder quantity, S-s, was the basic EOQ amount. A very short program, which included a provision for quantity-discount price breaks, was written to solve this problem.

In a situation where price-break discounts are given by a vendor for quantity ordering of a part or product, it is impossible to derive the opti-

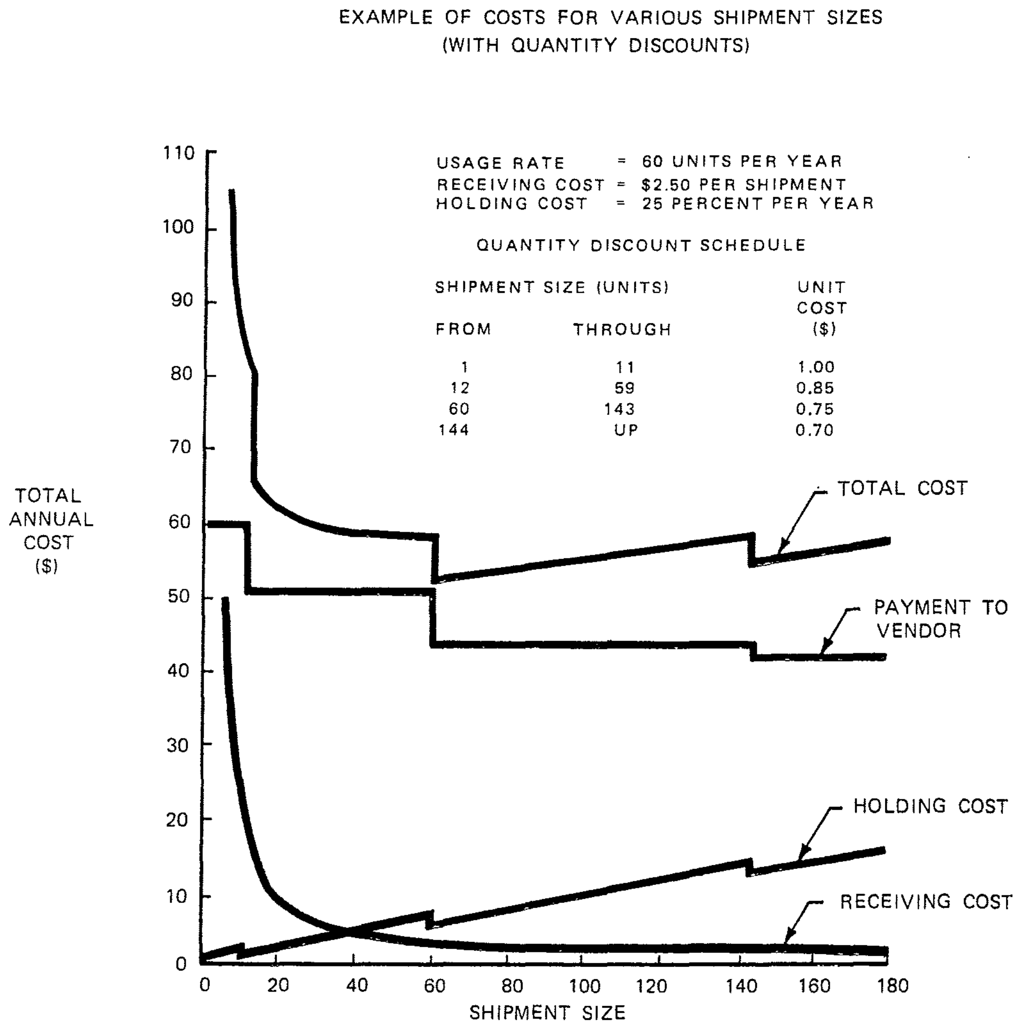

Fig. 7. Example of cost function type handled by the economic order quantity model.

mum reorder quantity of that part by one single calculation. All important points on the total-cost curve of Fig. 7 must be computed and the Q corresponding to the minimum T selected as the optimum amount.

Total cost is here defined as

$$T=R(U/Q)+H(Q/2)C+UC, \tag{1}$$

with the definitions given in Table I. Note that this formulation of total cost must also include the cost of the material itself in addition to the traditional provisions for holding and receiving costs.

The general EOQ total-cost curve consists of several pieces of U-shaped total-cost curves. The discontinuous points on this curve are the points at which quantity discounts come into effect.

When only one price is included and no quantity discounts are available, the optimum reordering quantity can be obtained by simple differential calculus and is given by the following formula:

$$Q_E = (2RU/CH)^{1/2}. \tag{2}$$

When more than one C exists, there is a Q_E that can be calculated for each value of cost per unit. In this case, it is necessary to calculate Q_E for each C. This calculation can result in one of three posible outcomes for each cost interval: the optimal Q_E can fall within the interval, or above or below it. If the Q_E for any price level falls within this range (I_1) as a result of the EOQ calculation, then it is clear that this order quantity is the optimal for the corresponding price level and it is necessary to calculate T for this value in order to compare it with other T's so that the Q_E corresponding to a minimum T may be chosen. If, however, this Q_E value falls outside of the corresponding quantity interval, then we can argue as follows. If the recommended Q_E is on a lower interval, then the price-break point that was the lowest value of the original interval would be less expensive than any other point on that original interval because total cost is a monotonically increasing function of Q within this interval. If, on the other hand, the Q_E value falls in an interval that recommends purchases greater than those in I_1, then it is clear that no total-cost calculation corresponding to this Q_E is necessary. Since C decreases in the next higher interval and, all other things being equal, a decrease in C implies an increase in Q_E, we can be certain that the true minimum T does not lie in I_1, even at its upper limit. Therefore, we conclude that the calculation of total cost corresponding to all Q_E values that lie within their own intervals and total cost corresponding to all price breaks where EOQ is below the lower limit will provide sufficient information to determine an optimal EOQ policy.

The input format for the EOQ program was similar to the others and essentially consisted of:

1. A receiving cost.
2. Holding-cost percentage.
3. A usage rate.
4. A quantity-discount schedule.

The output, however, was much simpler, consisting of only a basic

EOQ quantity describing the units per shipment. Incidentally, it should be mentioned that EOQ and the ERB model gave the same answer when the ERB model was fed an input schedule of requirements that reflected uniform usage (i.e., a long list of relatively small, identical requirements). However, in situations which were of this form, the input format for the EOQ program was much easier and, accordingly, EOQ was used.

IMPLEMENTATION

Two BASIC problems were encountered in implementing these programs.

1. The first problem was that management personnel had difficulty comprehending the scope and usefulness of the four programs; they did not fully understand the differences in each model and therefore did not always know which program to apply. To solve this problem, management was first given a formal oral presentation of the capabilities of these programs. Then, at the conclusion of this presentation, it was suggested that the division supply an individual who could be trained in the basic logic of the various models so that he could act as resident expert in the proper application of the scheduling programs. This approach was felt to be the most efficient solution to the problem, since it would have been a very difficult job to train all of the purchasing personnel so that they each would be able to recognize a scheduling problem and formulate it in terms applicable to the correct computer model. This approach was generally successful.

2. The second problem was to get an agreement on which values should be used for the input data required by the programs. This problem included three areas of controversy:

a. The corporation as a total believed its cost of capital to be 15 per cent per year before taxes. However, because of historical traditions, the divisions were charged an annual rate of 6 per cent for working capital by the corporate office. Consequently, it was understandably difficult to motivate division management to use the higher cost-of-capital figure that the corporate office desired in evaluating investment decisions.

b. Another controversy involved the treatment of progress payments. Contractors in the aerospace/defense industry are frequently in a position to receive progress payments on cash outlays for contracts. If a company receives a 70 per cent progress payment reimbursement from the US Government for all cash outlays made on material (after a constant time lag), then, in effect, the marginal amount of company capital tied up in the project is only 30 per cent of the invoice cost and the cost of capital should be lowered accordingly.

c. The final area of controversy in the second problem required the resolution of differences between the various before- and after-tax costs to be used as inputs. These differences are properly handled in accordance with the following relation: (after-tax cost) = (1-tax rate) · (before-tax cost).

All three components of the second problem were solved simultaneously by preparing a short program to simulate the operations of a firm that has two assets on its balance sheet—earning assets and inventory. This program first reads nine constants that describe the structure of the firm:

1. Interest rate.
2. Return on investment in earning assets.
3. Debt-to-asset ratio.
4. Income-tax rate.
5. Receiving cost.
6. Material cost.
7. Material usage rate.
8. Ratio of total assets to annual usage.
9. Per cent of progress payments tied to date material is received.

It then asks (when used in the conversational mode) for a holding-cost rate to use in the EOQ formula that controls the allocation of total assets between inventory and earning assets. When the holding-cost rate is supplied by the operator, the program calculates and prints a set of financial statements. Whenever a holding-cost rate other than the optimum, $(1-P)K_a/(1-t)$, is used the resulting income statement and balance sheets show up unfavorably. Therefore, this program was used to help convince skeptics that the value for H in the EOQ formula

$$Q_E = (2R_b U / C_b H)^{1/2} \tag{3}$$

should be

$$H = (1-P)K_a/(1-t). \tag{4}$$

ADVANCED EFFORTS

WHILE IT is felt that the four management-science programs described in this article provided a valuable increment to management's decision-making capability, the programs themselves are not perfect products and are subject to continued refinement and improvement. For example, one possibility is that, instead of computing property tax as a percentage and including it in holding cost, one could segregate and analyze it according to the exact dates that property taxes are assessed. This type of improvement would result in a more closely optimum schedule of shipment deliveries. Another possibility for improvement could be to further develop the RAX terminal on-line programs in a conversational inquiry mode so that a person unfamiliar with the models would be automatically led to the correct program by a conversational-mode computer program that would ask him necessary pieces of input data and accordingly provide the correct calculations.

CONCLUSION

BY LISTENING closely to problems as stated by operating personnel and by adjusting an operations-research effort to model existing conditions closely, an operations-research study led to the development and implementation of some computer programs that were helpful in inventory management.

In the three years since this work was first done, each of the authors has attempted to implement the programs at several installations. Some of these attempts were successful, while others were not, although all were made in an environment where success was feasible and technically desirable. In each case, the human element reflected through the attitude of the people who were to use the application was responsible for the success or failure of the program.

From an observation of these successes and failures, the following empirically observed 'laws' are presented without proof:

First law. The successful implementation of any operations-research project is inversely related to the number of authority-possessing persons associated with the project as follows: $P(S_i)=1.0-F_i$, where $P(S_i)=$the probability of a successful implementation on the ith OR project, and $F_i=$the fraction of people in the organization with authority to comment on the implementation of the ith OR project.

Second law. The time required to implement an operations-research project's results successfully is directly proportional to the time required to complete the operations-research work and the number of people with anything to say about its implementation: $E(T_{2,i})=K\cdot T_{1,i}\cdot F_i$, where $K=$a constant of proportionality (i.e., 'infamous K'), and $E(T_{2,i})=$the expected time required to establish Project i as a success or failure in implementation terms, and $T_{1,i}=$the time required to develop an OR model for the ith project.

Third law. When there exists a set of conditions such that the operations-research model is presented as a subordinate component of a new data-processing system desperately needed to relieve intolerable clerical or operating burdens, $P(S_i)$ becomes arbitrarily close to 1.0 and $E(T_{2,i})$ approaches 0.0.

The most successful implementations of the four inventory scheduling programs resulted when the power of the third law was brought to bear in such a manner as to override the effects of large F_i. Conversely, good success was also achieved whenever F_i could be kept very small, regardless of the size of the other variables.

REFERENCES

1. Arrow, K. J., et al, *Studies in the Mathematical Theory of Inventory and Production,* Stanford University Press, Stanford, 1958.
2. Churchman, C. W., et al, *Introduction to Operations Research,* Part IV, Wiley, New York, 1957.
3. *Computer Management Science in an Inventory Control System at Sandia Corpora-*

tion, International Business Machines Corporation Publication E20-0165, White Plains, New York.

4. HADLEY, G., *Nonlinear and Dynamic Programming*, Addison-Wesley, Reading, Massachusetts, 1964.
5. ——— AND WHITIN, T. M., *Analysis of Inventory Systems*, Prentice-Hall, Englewood Cliffs, 1963.
6. HANSSMANN, F., "A Survey of Inventory Theory from the Operations Research Viewpoint," in ACKOFF, R. L., *Progress in Operations Research*, Volume I, Wiley, New York, 1961.
7. *IMPACT—Inventory Management Program and Control Techniques*, International Business Machines Corporation Publication E20-8105, White Plains, New York, 1962.
8. WAGNER, H. M., *Statistical Management of Inventory Systems*, Wiley, New York, 1962.
9. ——— AND T. M. WHITIN, "Dynamic Version of the Economic Lot Size Model," *Management Sci.* **5**, 89–96 (1958).