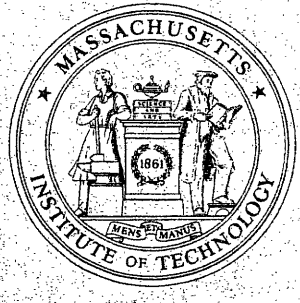
OPERATIONS RESEARCH CENTER

working paper



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

A SURVEY OF THE INVENTORY CONTROL-DETAILED SCHEDULING PROBLEM

by

Jonathan Golovin

Sloan School of Management
M.I.T.

OR 028-73 September, 1973

Supported in part by the Office of Naval Research
under contract 67-A-0204-0076

ABSTRACT

This paper surveys the inventory control problem in pure inventory systems and the detailed scheduling problem in the job shop and assembly line environments. Conditions under which inventory control systems may be substituted for production scheduling systems are briefly reviewed. The paper concludes with a discussion of the difficulties in integrating (production scheduling) inventory control systems with detailed scheduling systems.

Inventory Control

I. Introduction

This paper surveys the inventory control problem for pure inventory systems, an analogue of the production scheduling problem. The pure inventory system involves no production; instead, goods are purchased from outside suppliers, possibly repackaged or merchandised, and then sold to the concern's customers. Typical examples of pure inventory systems are wholesale and retail concerns.

The problem is to determine the order size for every item either in a discrete or continuous time frame. Optimal policies will be developed, for varying demand conditions, as well as practical solutions. Finally the use of inventory control systems in production settings will be explored.

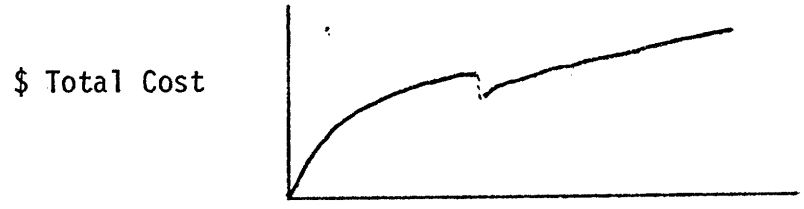
II. Cost Structure and Definitions

The multi-stage inventory and distribution system has been ably covered by Karmarkar [20]. The only addition to this area will be in the production setting and this will be at the end of the paper. In this section the single location problem is covered.

In the general case, a single location (as a plant, warehouse, or store) inventories n separate items or products. These are ordered from m outsider suppliers ($m \leq n$). Each item has a demand distribution (in each time period) and a lead time distribution.

The ordering cost structure involves a fixed cost and a variable cost dependent on the order quantity. This variable cost is usually concave and/or non-continuous (Figure 1).

Figure 1: Total Ordering Cost of Item i



In the simplest case, the variable cost is linear. However, a variety of quantity discounts are common. Some examples are:

(1) $ c/unit for the first x units ordered

$ d/unit for the next y units ordered

$ e/unit for the next z units ordered

.
.
.

$ t/unit for the next p units ordered

where $c \geq d \geq e \geq \ldots \geq t$

(2) $ c/unit if the amount ordered q is x

$ d/unit if $x \leq q < y$

.
.
.

$ t/unit if $s \leq q < w$

where $c \geq d \geq \ldots \geq t$; $x \leq y \leq \ldots \leq s \leq w$

If several items are ordered from the same supplier, there may be a shared fixed cost as well as a fixed cost per item. Price breaks may also be a function of the joint order size.

This cost structure is assumed to include both transportation and purchase costs. (Given the separate cost structures, they may be combined to yield the joint structure as assumed).

Once an order is received, there are costs associated with its storage and handling. These are referred to as inventory holding costs, encompassing costs of

obsolescence, insurance, handling, damage, interest on investment, and security to name the major components. Normally these are assumed directly proportional to the inventory investment in dollars. As a simplification, this proportion is usually given as a percentage from 12 to 25%.

The last cost structure is associated with demand satisfaction. If the final selling price of the item has been fixed, the relevant costs are all associated with unsatisfied demand. Unfilled orders may be lost, or backordered until stock is available. Again, there are several cost structures possible: the back order may involve a fixed cost (due to paperwork) and a variable cost (a loss due to a delayed cash flow and/or an implicit cost charged for customer impatience with resulting loss of goodwill and possible reduced demand in future periods). The variable cost may be per unit backordered and/or per unit time. It may be a concave or convex function.

This completes the relevant costs. In the general case demand for an item is assumed to be stochastic and possibly correlated with demand for other items. Order size may be bounded, either jointly (when storage space, dollar investment, etc., are constraints) or individually (when suppliers or transporters place limits on order quantities accepted).

III. Models

The literature is replete with solutions to specialized cases of the above problem; solutions for perishable items, multiple fixed cost structures, low demand items, and on. It would be a lengthy chore to enumerate all the special cases that have been dealth with, so instead, the most important cases will be dispatched.

A. Deterministic Models

1) Historically, the earliest result is the Wilson (or Harris) lot size formula. Under constant, continuous, deterministic demand, with a deterministic lead time and delivery rate, and fixed ordering cost with linear variable costs, the yearly total costs can be written as a function of the lot size alone (see Table 1). This is a convex function and is minimized with respect to Q, the order size. This case can be extended to include backorders (assuming a fixed cost and linear variable cost).

Differentiation of the total cost equation produces the well known square root formula. This model is not sensitive to errors in its parameters which is a desirable feature.

With a deterministic lead time, this model can be extended into an inventory control system, regulating order frequency as well as order size. (Referring to Figure 2 on Table 1). The order point is defined as the inventory level at which to place an order. For the no-backorder case, given deterministic demand, the order point is the lowest inventory level at which an order can be placed without running out of stock before replenishment arrives. This level is the lead time demand (the lead time * the usage rate). If backorders are allowed, the level of backorders desired is subtracted from the lead time demand to calculate the order point.

Now the following inventory control system exists: when the inventory level reaches the order point (O.P.), order the order quantity (O.Q.), abbreviated an (O.P., O.Q) or (Q,r) system. Note that a continuous review of inventory is implicitly assumed, though not necessary if assuming deterministic demand. Given the usage rate and the current inventory level, the date of the next order is predetermined.

This model is optimal given that the extremely restrictive assumptions are realistic.

The final complication treated in this section is quantity discounts. These are normally quoted as a step function of the order size, not as a continuous function. This means that our methodology (of differentiating a cost equation) is not applicable. Instead, direct comparison of total cost is often required. A method of solution is given in Table 2.

2) Dropping our assumption of constant, and continuous, demand (while still retaining all others), results in the dynamic lot size problem. This has been solved by Wagner and Whitin [36] and is discussed in the paper on production scheduling by Hax [14] (with extensions as noted).

3) If items are not independent but share a fixed ordering cost, or joint quantity discount, their EOQ's (economic order quantities) must be calculated jointly. In the former case, the total cost equation for all items involved includes a shared fixed cost. The cost expression is then minimized with respect to the period between orders, assuming all items are ordered jointly. Bomberger [2], Hanssmann [10], Standard and Gupta [28], and Hodgson [16] have all suggested improved ordering policies. These reduce the total cost by ordering items with relatively small demands at integral multiples of a basic review period.

When items are involved in joint quantity discounts, there are so many possible combinations of order sizes that optimization under general conditions (where each item has its own demand and cost structure) is not computationally feasible. Optimization would require dynamic programming with a state space of at least the number of items. No references on this case have been discovered.

The last joint or multiple item problem occurs when constraints exist on the use of some total resource. Common examples are constraints on total space, weight, or dollar investment. The joint constraint again precludes individual

optimization. This problem can be solved using lagrange multipliers, a standard method for constrained optimization [29].

4) Stochastic Models

a) (Q, r) systems - fixed order quantity - the Wilson lot size model is extended to consider stochastic demand. Demand is assumed to be identically independently distributed in every period. Again, an order point, order quantity system is assumed. The expression for the average annual cost is written and then minimized with respect to the order point and order quantity. The exact formulation is complicated and usually bypassed in favor of several heuristic treatments [9]. The difficulty arises from computation of the expected backorder cost. Unless a convenient demand distribution is assumed, solution is even more difficult as it requires calculation of the expected amount backordered during a lead time (similarly for inventory on hand at the end of the lead time). Computation of optimal policies requires an iterative search routine, usually a computer procedure. As in any search procedure, local minima may be mistaken for global optima.

This model has two assumptions that must be emphasized. For optimality, there must be continuous review and demand must be in single units, i.e., the order point cannot be overshot; there must be an ability to place an order precisely when the order point is reached. This usually means assuming a poisson process generating function for demand (so demand occurs in single units). If order size is also random variable it may not be optimal to order a fixed quantity each time an order is placed. In this case an (S, s) policy is required; (section II-3), a more general operating policy of which the (Q, r) policy is a special case.

Normally, two approximate models are used in place of the optimal formulation for computational ease.

The first heuristic solution assumes that an arriving order always raises the net inventory level above the reorder point. (The exact treatment allows for a large number of backorders to accumulate over the lead time; this means that arrival of outstanding orders might never bring the net inventory back up to the reorder point, so that another order would never be placed. The exact treatment defines the order point in terms of inventory position equal to net inventory (or inventory on hand) plus on order minus backordered, alleviating this problem.)

Now the expected cost of ordering, inventory holding and shortages can be calculated easily. This expression is again minimized with respect to an order quantity and order point. However, they must be solved for concurrently, as they are functions of each other. Solution is possible using an iterative method (Table 1-C) computing a value for Q, using that to compute r, substituting that value in the correct expression for Q, etc.

The second approximate method used decouples the stochastic consideration and leaves a deterministic problem. Instead of using a backorder (or stock out cost), a desired buffer stock, defined as the average stock on hand when an order arrives, is computed directly from a customer service level.

The customer service level can be specified in several forms, but usually in one of the following:

1) percent of demand backordered per year
2) percent of orders backordered year
3) probability of a stockout per cycle
4) the fraction of time stocked out

From the designated service level, a safety factor k is calculated such that k times the variance of the forecast error is the buffer stock. Then the order point is the mean demand over the lead-time plus the buffer stock, and the order quantity is the same as in Section I-1, the deterministic case.

This is appealing both computationally and for implementation purposes. It is easier for a manager to set a service level than to specify a backorder cost. His implicit backorder cost, of course, can be calculated from the service level he sets.

However, the independent determination of Q and r is unsatisfactory for situations with high implicit stockout costs, high variance in lead time demand (forecast error) and/or low fixed cost per order. The order size can be traded off against the size of the buffer stock (to decrease the percent of demand backordered) because the order size determines the number of times per year that stock-outs are possible. In these cases, the first heuristic is more appropriate.

b) (T, r) models - fixed period - in the previous model, stochastic demand was absorbed by allowing the time between orders to vary. The alternative procedure is to fix the time between orders and let the order quantity vary. A review period system no longer requires a perpetual inventory; this feature explains its popularity over (Q, r) systems. A continuous review model requires a computer system with no aggregation of events, a system that may not even be feasible if cost effective.

In a (T, R) model, the inventory level is reviewed at the beginning of each period (of length T) and an order is placed to bring inventory up to level R. Again, we have three levels of models, exact, heuristic and decoupled.

In addition, the review period may be given or can be a parameter to be optimized. Again, the average annual cost equation is written as a function of M and r and then jointly solved, using a search procedure.

If we assume that an arriving order is always sufficient to satisfy any existing backorders, we can solve the heuristic model using an iterative method (as Newton's method).

Finally, if we decouple the stochastic consideration, T is usually set equal to the annual demand divided by the deterministic EOQ (resulting in the same period as for the model in Section I-1). M is then set to the EOQ plus the mean demand over the lead time plus a buffer stock as before.

c) Comparison of (Q, r) and T, R) systems: note that for the deterministic case, these systems are identical. In the stochastic case, the real difference in costs between these two systems lies in the system support and in the buffer stock required for operation. A (Q, r) system requires a more elaborate control system; however, it has a lower inventory cost.

The (T, R) system has an inherently longer planning interval (a lead time and review period) than the (Q, r) system (a lead time). Any decision made at the beginning of a period can not be corrected for until the next decision is made and that future order received. This longer horizon normally has greater uncertainty in usage and therefore, a larger buffer stock is required to yield equivalent service level to a Q, r) system.

d) (S, s) systems - the following system is postulated: inventory is reviewed at the start of each period; if the inventory on hand is less than s, an order of size (S - s) is placed, if greater than s, no order is placed. This system was developed by Scarf [24] building on earlier work by Arrow, Harris, and Marschak [1]. Both the (Q, r) and (T, R) systems are subsets of this basic policy; (Q, r) is the (S, s) policy for continuous review; (T, R) is the (S, s) policy when no set up charge exists (note this implies that a (T, R) policy is not globally optimal; that there is an inventory policy that is superior. However, given that a (T, R) policy is chosen for use (while the system is not optimal) optimal parameters can still be chosen for T and R).

Under the following conditions, the (S, s) policy is the optimal policy to follow. Scarf allows a more general cost structure and demand distributions than

previously considered in Sections 1 and 2. He considers an ordering cost c(z) where z is the amount purchased; a holding cost h(-) for excess inventory and a shortage cost p(·) for backordered demand.

$$c(z) = \begin{matrix} 0 & z = 0 \\ k + c \cdot z & z > 0 \end{matrix}$$

L(y) = expected holding and shortage costs in a period given an initial inventory of y

$$= \int_0^y h(y - \xi)\, \zeta(\xi)\, d\xi + \int_y^\infty p(\xi - y)\, \zeta(\xi)\, d\xi ; y \geq 0$$

$$\int_0^\infty p(\xi - y)\, \zeta(\xi)\, d\xi$$

where ξ = demand in the period and $\zeta(\xi)$ its probability density function.

The problem is to find the ordering decisions for an n period problem to minimize the total expected ordering and inventory costs (holding and shortage), $C_n(x)$ where x is the initial inventory.

$$C_n(x) = \min_{y > x} \left\{ c(y - x) + L(y) + \alpha \int_0^\infty C_{n-1}(y - \xi)\, \zeta(\xi)\, d\xi \right\}$$

(assuming no delivery lag and a discount factor for further costs). Given the above cost structure, it can be shown that $C_n(x)$ is K-convex where K convexity is defined as follows:

f(x) is K convex if $K + f(a + x) - f(x) - af'(x) > 0 \quad \forall\, a > 0, \forall\, x$

where $K \geq 0$ and f(x) is differentiable. If we further define

(a) $G_n(y) = cy + L(y) + \alpha \int_0^\infty C_{n-1}(y - \xi)\, \zeta(\xi)\, d\xi$

it is clear that it is optimal to order from inventory level x if there is some $y > x$ with $G_n(x) > K + G_n(y)$, i.e., the gain from ordering due to lower costs must be greater than the fixed charge k.

If we define S_n as the minimizing value of y in equation (a) and s_n as $G_n(s_n) = G_n(S_n) + K$, then the policy designated (S_n, s_n) is optimal. Any cost function $C_n(x)$ that is k-convex is minimized using an (S,s) policy. For the n period horizon, there will be n pairs of these numbers.

This clarifies why the (T,R) policy is not an optimal policy unless k = 0. An order is placed in every period regardless of the inventory level in relation to s. If k = 0, then $s_n = S_n$ and this ordering policy is optimal.

Scarf extends this model to consider delivery lags.

Note that the one period problem is the classic newsboy problem; when k = 0, L(y) is minimized.

Veinott and Wagner [31] explore the computation of (S,s) policies. If demands are assumed to be identical independently distributed random variables in each period then $s_n = s$ and $S_n = S$ in every period n. This is the starionary (S,s) policy. Instead of using a dynamic programming formulation, more efficient computational tools are available (as renewal theory [19] that exploit the policy property). If demand is not static but dynamic, dynamic programming can be used to calculate the optimal policy. Naturally, it is recalculated each period as demand materializes and the inventory position changes. If a computer dynamic programming routine is available the calculation of finite horizon solutions is rapid. Infinite horizon solutions are more difficult to calculate, but are more of theoretical interest than practical use; if a twenty period problem with discounted costs is solved, the effect of the 20^{th} period is negligible on the current decision; an infinite horizon seems unnecessary. However, for low cost, high volume, routine items the approximate models of Section 1 and 2 are sufficient. If 10,000 items are to be controlled, the 20 seconds computation time per item to calculate (S,s) policies may be more costly than the potential savings. This is especially true when demand is stationary and the period used large enough so that an order is always placed with either a (T,R) or (S,s) system. This situation is common in industrial settings and, therefore, the (S,s) policy is replaced by a (T,R) policy for practical implementations.

Excellent references for additional material on any of these models are Scarf's survey of inventory techniques [25], Hadley and Whitin's book on inventory systems [9], and Veinott's survey of inventory systems [33]. The latter is an extensive survey that is difficult to surpass in any respect.

Further computational experience and a comparison of (S,s) performance to approximate model performance is reported by Wagner, et al. [37]. In particular, several approximate methods for calculating S and s were explored. These were found to be computationally efficient and near optimal.

(S,s) policies are no longer optimal when the cost function is not k-convex. The simplest example of this situation is the case of price breaks; the unit cost of an item is not constant. S will depend on the current level of inventory and the price break structure. Given this more general structure for $G_n(x)$

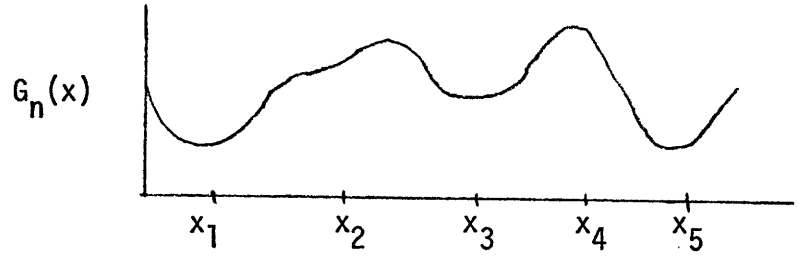


Figure 4

a more general policy may be of the form: if $x < x_1$ order to S_1

$x_1 \leq x \leq x_2$ do not order

$x_2 < x < x_3$ order to S_2

$x_3 \leq x < x_4$ do not order

etc.

5) Special Cases

The following are examples of the more pathological situations. They are often based on cases found in industry that did not conform to the usual model assumptions.

a) The first situation is declining demand; when the item has reached the end of its life cycle. Moore [23] uses the concept of an all-time requirement.

As demand is decreasing non-linearly, EOQ concepts are inadequate, resulting in overstocking. Smoothing forecasts also are inaccurate. Moore has found several functions that potentially fit the log of demand data when plotted against the log of the year (numbering the first year of sales decline as year 1).

b) Another unusual condition is the slow moving item [13]. Demand can no longer be assumed continuous. The solution to the EOQ is to find the value Q such that

$$\frac{Q(Q-1)}{2} * \frac{I}{A} \leq S \leq \frac{Q(Q+1)}{2} * \frac{I}{A}$$

where Q = EOQ; S = annual usage, I = holding cost per year per unit, A = set up charge.

c) If an unusual measure of performance is desired, the usual optimal policy may no longer be suitable. Hausman [11] discusses the situation when the measure of performance used is a backorder cost per line item (independent of the volume of the order). For the single stage, stochastic demand case, the total cost equation is calculated under the new cost structure and then optimized with respect to the order point and order quantity.

d) Demand may also be a partially deterministic and partially stochastic. A typical case arises when a part is used in assembly operations and ordered by spare parts dealers. The production schedule is known in advance giving deterministic demand while demand for spare parts is stochastic. Stockout costs may be different in each situation. This problem is a subset of work on inventory rationing policies given several classes of demand occur (Evans [7], Kaplan [18], Topkis [30], Veinott [32]). Their results show certain critical rationing levels (for each class of demand) such that demand for lower priority classes is backordered when inventory falls below those levels. Veinott has shown conditions under which the optimal rationing policy remains identical in all periods, reducing computation. However, Evans and Kaplan have demonstrated that simple rules can often capture the majority of the improvement optimal rationing policies offer with far less computation.

Hausman and Thomas [12] indicate a procedure to calculate an optimal policy for the combined deterministic-stochastic problem. The calculation involves dynamic programming, as most of the rationing problems do, at great computational cost. Instead of calculating optimal policies, they find conditions under which a (Q,r) policy would be appropriate and those for (T,R) system use.

IV. Application to Production Scheduling

The models discussed are appropriate in any situation where their assumptions are realistic. The fixed charge may represent an ordering cost or a machine set-up. Therefore, these same models can be applied to certain restricted production situations. Instead of an outside supplier, the order will be produced internally.

The models in Sections II and III are single item, single stage models. This implies that there is only one operation to be performed that transforms the raw material into the final good. That operation has a setup associated with it (possibly $0) and the final goods have a greater holding cost than the raw materials. Note that the raw materials are controlled with our previous models.

Most of the models considered were uncapacitated; there were no limits on the order size and items were considered independently. In production, this case implies excess capacity (both in facilities and manpower). If demand is nearly constant for each item, use of (O.P., O.Q) control systems to schedule production is feasible. Manpower planning is only required when demand, costs, or supplies of raw materials, are not constant. Even if seasonal planning techniques are required, they may be supported by a modified inventory control system. This approach is detailed in the survey by Hax [14]. The last consideration involves multistage production systems.

V. Multi-Stage Production Models

The following is a brief survey of multi-stage production models. Before investigating specific models, it is necessary to define several multi-stage configurations.

a) serial - each stage has at most one immediate predecessor and successor (Figure i)

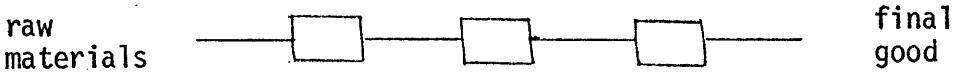


Figure i. Serial Configuration

b) parallel - each stage is single with no predecessor or successor but stages may share costs (Figure ii)

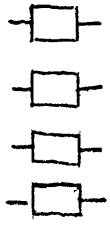


Figure ii. Parallel Configuration

c) assembly - each stage has any number of predecessors but at most one successor (Figure iii)

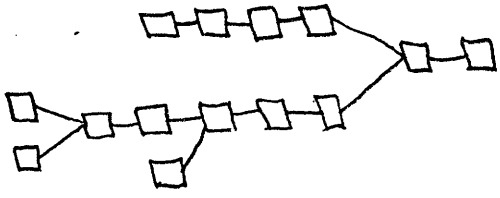


Figure iii. Assembly Configuration

d) arborescent - each stage has a single predecessor but any number of successors (Figure iv)

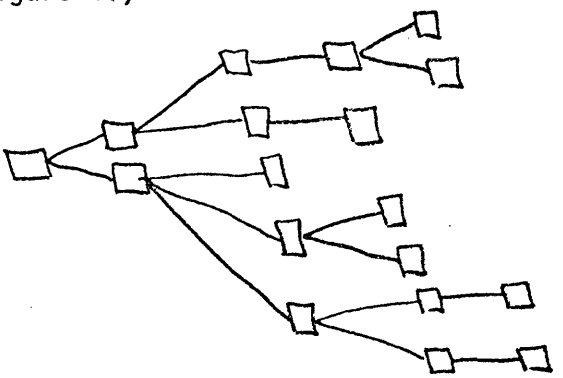


Figure iv. Arborescent Configuration

e) acyclic - each stage can have any number of predecessors and successors but, if stages are numbered, a stage numbered j can only be a predecessor of any stage p for $p>j$ (Figure v)

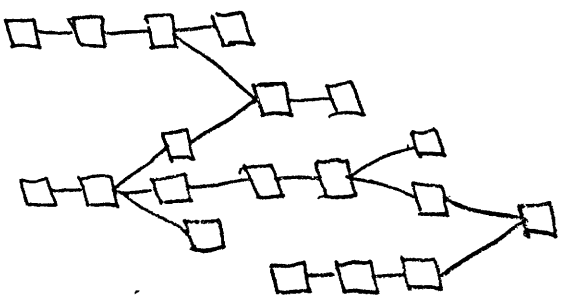


Figure v. Acyclic Configuration

f) cyclic or general - no restriction on the relationship between stages

In additional to optimal formulations, there are several practical heuristic solutions as before. The simplest is to treat each stage independently, i.e., each stage might have a (Q, r) system; when stage j reaches its O.P., it orders from stage j-1. The order point and order quantity are calculated at a stage as if the demand it faced were an independent random variable. This approach is an obvious misuse of an (O.P., O.Q.) system. The demand that stage j faces is not independent; it is sequentially dependent on the stages preceding it until final customer demand. In other words, more information exists than is being used.

A second alternative is to eliminate work in process inventories. Demand on the final stage n is exploded back through the system to the very first department; when Department 1 is finished, Department 2 works on the components, etc.

In this approach, production reacts directly to realized demand. All work in process inventories are eliminated at a cost of an increased number of setups and a larger buffer stock at the final stage of the finished good (since the lead time is longer). Finished goods usage dictates production so, in this case, each department is treated as completely dependent.

A third option avoids many of the disadvantages inherent to these first two. In a base stock system [22], each stage controls its ordering policy but based on information of actual customer demand. Instead of each stage reacting to its successor's ordering policy, it can produce when its own inventory level minus the customer demand reaches its order point. In more sophisticated systems, the customer demand may be lagged to indicate when that final demand will actually affect that stage, i.e., when its successor stage will be ordering its EOQ.

Since the system is driven by actual and not generated demand from secondary stocking points, uncertainty about final product demands is not amplified into uncertainty over the timing of in-system needs. This latter uncertainty inherent in the first option results in increasing variability of demand stage by stage, as the EOQ's successively increase going back toward the first stage.

If a stage faces demands from several successors (several finished good require a given component), its total demand may be less irregular and an (O.P.,O.Q.) system may be justified. However, if a finished product has many components, using an (O.P., O.Q.) system to control each component will result in a very low probability that all these items will ever be in stock simultaneously. This will increase the lead time and often lead to production congestion; several incomplete orders will sit on the floor waiting for delivery of a component.

This next section deals with optimal or near optimal results beginning with serial structures. Hanssmann[7] solves the specialized problem of one time (stochastic) demand with several classes of balking customers, given by the percentage of customers that will wait for the final good to be produced from work in process inventory at stage i; i = 1 to n. The problem is to determine the optimal stocking level at each stage. Given these percentages and the demand distribution, these are determined by constructing the total cost (profit) function, taking partial derivatives and solving the resulting set of linear equations.

Zangwill [28] considers production schedules under dynamic deterministic demand with no backorders, concave production and inventory costs, with no capacity constraints. He formulates the problem as a network flow, and then uses results for single (production) source, concave cost networks to characterize an optimal solution. These solutions are extreme flows, a flow with at most one positive input to any node. This result suggests a dynamic programming algorithm to find the optimal schedule. This approach is a further extension of the Wagner-Whitin model to multi-stage (serial) production.

Many results for more general multi-stage cases can be simplified to the serial case. The reader should assume that any specialized case of a more general situation is also encompassed.

The parallel case has been again treated by Hanssmann[7] . However, his treatment is relatively uninteresting as he defines the parallel model as one of independent single stage problems with some constraint on total inputs, outputs or inventories (i.e., the capacitated case).

Continuing to the assembly case, Crowston, Wagner and Williams [29] prove that for deterministic, constant demand, no capacity constraints, instantaneous production, no backorders, and constant marginal production costs, the ratio of lot size between stage j-1 and j (where n is the final stage) must be a positive integer. The optimal lot sizes are then solved for by dynamic programming.

Crowston and Wagner [5] extended these results to the dynamic demand case. Solution is by dynamic programming or branch and bound (for the near serial case). This problem (multi-stage with concave production and linear holding costs) is an example of a Leontief substitution system, examined by Veinott [31]. His results, in line with those of Zangwill, show that at least one optimal solution (in the non-capacitated case only) is an extreme flow; or, production can occur only if entering inventory is zero. Love [32] proves the added property for the series model that if stage j produces in period t, stage j-1 must also produce and if stage j does not produce, stage j-1 does not produce.

Crowston and Wagner's algorithm's solution time is linear with the number of stages but exponential in the number of time periods.

For arborescent networks, Kalymon [33] assumes deterministic demand with no backlogs, and linear holding and production costs (with setups). The "optimal" schedule at the first stage is then used to generate "optimal sequences" at the next stage and so on. Results from Veinott [31] are used to justify decomposing the problem in this fashion. The algorithm is exponential in the number of following echelons (or stages). Each stage is solved by Wagner-Whitin.

It may be that improved branch and bound, best bud growth, and dynamic programming techniques, as well as faster computers may aid in the computational speed.

In acylic networks, Zangwill [34], assuming deterministic dynamic demand, backlogging and no capacity constraints again constructs a Wagner-Whitin type dominant set that contains the optimal solution. The last stage's requirements are used to construct the partial dominant set which then become the requirements for stage n-1 and so on. Dynamic programming then is used to solve for the optimal solution. Computationally, the series and parallel cases may be solved.

Simpson [35] first deals with the serial base stock situation. Under stochastic demand, deterministic processing times, and for given final service time, Simpson proves that each stage will either carry no W.I.P. inventory or the full

base stock (EOQ and buffer). For the acyclic case (any number of predecessors and/or successors), the demand at any stage is the sum of the demands drawing on it.

The general case has not been dealt with extensively. The only reference is to Henshaw [36].

In the case of seasonal multi-stage production, Crowston, Hausman and Kampe [37] assume no capacity constraints, an assembly model, bayesian updating of the demand distribution each period and no setup costs. The case of end of season delivery can be solved by dynamic programming but the case of delivery requirements each period is not computationally feasible. Instead, several heuristics are compared; the majority are newsboy type with various modifications of the average cost in intermediate stages and periods. One interesting aspect covered is the problem of long lead times in predecessor stages. Production in any stage is limited by the minimum production in any predecessor stage.

In summary, the multi-stage problem has not been convincingly solved. The general case is pruned by assumptions until its structure allows solution by a chosen technique. Capacity constraints are often waived, allowing concave network solution techniques. Linear costs without setup charges allow linear programming solutions. Small problems can be attacked with non-linear search techniques, etc. In the meanwhile, there is a great gap between the optimal formulations and practical implementation.

TABLE 1

case (a) no backorders

Q = lot size (in units)
A = annual demand in units)
S = fixed ordering cost (in $)
r = holding cost per $ of inventory per year
C = purchase cost per unit (in $)
p = delivery rate (units/period)
u = sales rate (units/period)

$$T.C. = \text{(Total Cost)} = \frac{A}{Q}[S] + \left[rC\left(1 - \frac{u}{p}\right)\frac{Q}{2}\right]$$

$$\text{setting } \frac{\delta T.C.}{\delta Q} = 0 \text{ ; } Q_n^* = \sqrt[n]{\frac{2SA}{rC(1 - \frac{u}{p})}}$$

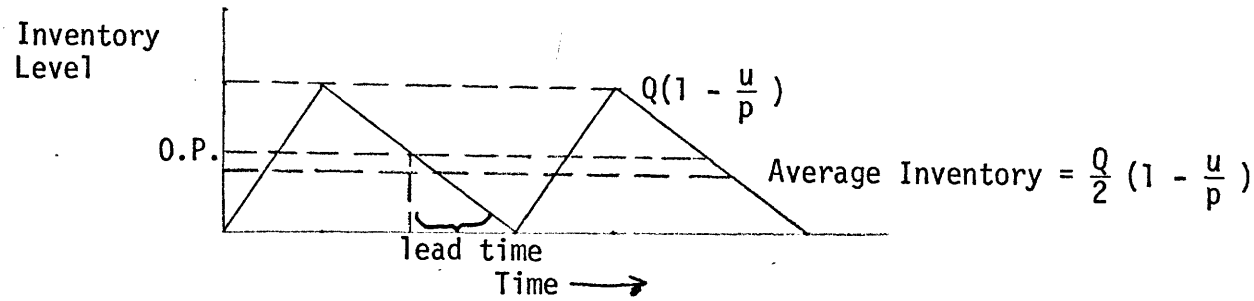


Figure 2

case (b) backorders [3]; backorder cost structure is linear and proportional to the length of the backordered period

same as above, except

I_{max} = maximum inventory level
C_s = shortage cost per unit time
C_H = holding cost per unit time

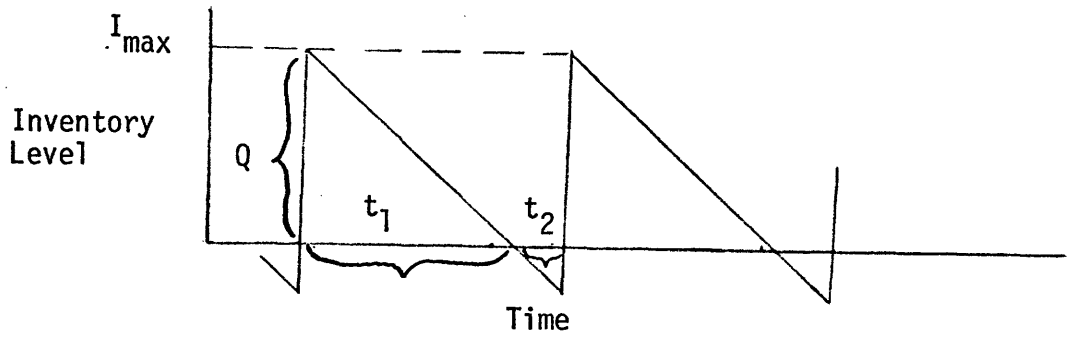


Figure 3

TABLE 1 (continued)

T.C. =

$$\frac{S.A}{Q} + C_H \frac{I_{max}^2}{2Q} + C_s \frac{(Q - I_{max})^2}{2Q}$$

as $t_2 = \frac{Q - I_{max}}{A}$, $t_1 = \frac{I_{max}}{A}$

$\therefore$ by $\frac{\delta TC}{\delta Q}$ and $\frac{\delta TC}{\delta I_{max}} = 0$

$$Q^* = \sqrt{2SA/C_H} \cdot \sqrt{\frac{C_H + C_s}{C_s}}$$

$$I_{max}^* = \sqrt{2SA/C_H} \cdot \sqrt{\frac{C_s}{C_H + C_s}}$$

case (c) stochastic (O.P.O.Q.) [8]

Let L(t) = net inventory at time t
d = lead time
f(x/d) = conditional for demand given a lead time of d
B = cost of a backorder (cost per unit backordered)
L_c = the order point
μ = mean demand over the read time

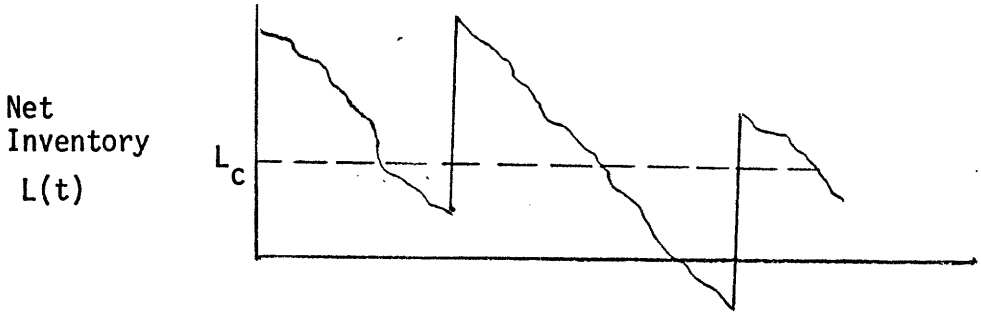


Figure 4

TABLE 1 (continued)

$$T.C. = \frac{AS}{Q} + rC\left[\frac{Q}{2} + L_c - \mu\right] + \frac{AB}{Q} \cdot \int_{L_c}^{\infty} (x - L_c)\, f(x/d)\, dx$$

$$\frac{\delta TC}{\delta Q} = \frac{AS}{-Q^2} + \frac{rC}{2} - \frac{AB}{Q^2} \int_{L_c}^{\infty} (x - L_c)\, f(x/d)\, dx = 0$$

$$\frac{\delta TC}{\delta L_c} = rC + \frac{AB}{Q} \frac{\delta}{\delta L_c} \left[\int_{L_c}^{\infty} (x - L_c)\, f(x/d)\, dx \right] = 0$$

an iterative solution technique is given in Hadley and Whitin [9]

case (d) stochastic (T,R) [8] T = period length (fraction of a year)

$$T.C. = \frac{S}{T} + rC\left[M - \mu - \frac{AT}{2}\right] + \frac{B}{T} \int_{M}^{\infty} (x - M)\, f(x/d + T)\, dx$$

if T is fixed, then

$$\frac{\delta TC}{\delta M} \Rightarrow \int_{M}^{\infty} f(x/d + T)\, dx = \frac{rCT}{B}$$

if T can also vary, the iterative procedure of section c is again required

TABLE 2[13]

FINDING OPTIMAL ORDER QUANTITIES WITH QUANTITY DISCOUNTS

Frequently purchases quantities must be chosen not only with ordering and carrying costs but also with vendor's discount schedules considered. Suppose, for example, we have the following price ranges:

Quantity Purchased	Unit Price
$0 \leq Q < u_1$	C_1
$l_2 \leq Q < u_2$	C_2
.....	.....
$l_i \leq Q \leq u_i$	C_i
.....	.....

where $C_i > C_{i+1}$ and $u_i = \ell_{i+1}$

The optimum purchase quantity can be determined as follows. First compute the order quantity Q_i for each price range by means of the expression

$$Q_i = \begin{cases} l_i & \text{if } Q_i^* < l_i \\ Q_i^* & \text{if } l_i \leq Q_i^* < u_i \\ u_i & \text{if } Q_i^* \geq u_i \end{cases}$$

where Q_i^* is given by:

$$Q_i^* = \sqrt{\frac{2AS}{rC_i}} \tag{3}$$

Then, proceed as follows:

(1) Choose the highest value of i for which $Q_i = Q_i^*$ (there will always be at least one such Q_i); call it Q_k.

(2) Test all other discount levels C_j, $J > k$, by computing the total inventory costs associated with that level TC(j) and compare it with TC(k), where

$$TC(j) = C_jS + \frac{rC_jQ_j}{2} + \frac{SA}{Q_j} \tag{4}$$

and

$$TC(k) = C_kS + \frac{rC_kQ_k}{2} + \frac{SA}{Q_k} \tag{5}$$

The optimum discount level j corresponds to that value of j such that TC(k) - TC(j) is the maximum positive value. If all TC(j) are greater than TC(k), then the level k is the optimum.

REFERENCES

1. Arrow, Harris and Marschak, "Optimal Inventory Policy," Econometrica, XIX, 1951.

2. Bomberger, "A Dynamic Programming Approach to a Lot Size Scheduling Problem," Mgt. Sci., Vol. 12, No. 11.

3. Buffa and Tauber, Production-Inventory Systems, Irwin, 1972.

4. Crowston, Wagner and Williams, "Economic Lot Size Determination in Multi-Stage Assembly Systems," Sloan Working Paper, MIT, #566-71.

5. Crowston and Wagner, "Dynamic Lot Size Models for Multi-Stage Assembly Systems," Sloan Working Paper, MIT

6. Crowston, Hausman, Kampe, "Multi-Stage Production for Stochastic Seasonal Demand," Sloan Working Paper, MIT, #587-72.

7. Evans, "Sales and Restocking Policies in a Single Item Inventory System," Mgt. Sci., Vol. 14, No. 7.

8. Groff and Muth, Operations Management, Irwin, 1972.

9. Hadley and Whitin, Analysis of Inventory Systems, Prentice-Hall, 1963.

10. Hanssmann, Operations Research in Production and Inventory Control, Wiley, 1962.

11. Hausman, "Minimizing Customer Line Items Backordered in Inventory Control," Mgt. Sci., Vol. 15, No. 12.

12. Hausman and Thomas, "Inventory Control with Probabilistic Demand and Periodic Withdrawals," Mgt. Sci., Vol. 18, No. 5

13. Hax, "Technical Note on Inventory Control," unpublished teaching note, MIT.

14. Hax, "Aggregate Capacity Planning," Operations Research Center Working Paper, MIT, September, 1973.

15. Henshaw, "Multi-Stage Inventory Models," master's thesis, MIT, 1969.

16. Hodgson, "Addendum to Standard and Gupta's Note," Mgt. Sci., Vol. 16, No. 7.

17. Kalymon, "A Decomposition Algorithm for Arborescence Inventory Systems," O.R., Vol. 20, No. 4.

18. Kaplan, "Stock Rationing," Mgt. Sci., Vol. 15, No. 5.

19. Karlin, "Steady State Solutions," Chapter 14 in Studies in the Mathematical Theory of Inventory and Production, edited by Arrow, Karlin and Scarf, Stanford University Press, 1958.

20. Karma kar, "Mathematical Analysis of Distribution Systems," (A Survey), Operations Research Center Working Paper, MIT, September, 1973.

21. Love, "A Facilities in Series Inventory Model with Nested Schedules," Mgt. Sci., Vol. 18, No. 5.

22. Magee and Boodman, Production Planning and Inventory Control, McGraw-Hill, 1967.

23. Moore, "Forecasting and Scheduling for Past-Model Replacement Parts," Mgt. Sci., Vol. 18, No. 4.

24. Scarf, "Optimality of (S, s) Policies," Chapter 13 in Mathematical Methods in the Social Sciences, edited by Arrow, Karlin & Suppes, Stanford University Press.

25. Scarf, "A Survey of Analytical Techniques in Inventory Theory," Chapter 7 in Multi-Stage Inventory Models and Techniques edited by Scarf, Gilford and Shelly, Stanford University Press, 1963.

26. Shwimer, "Interaction Between Aggregate and Detailed Scheduling in a Job Shop," Ph.D. thesis, MIT, 1972.

27. Simpson, "In-Process Inventories," O.R., November, 1958.

28. Standard and Gupta, "A Note on Bomberger's Approach," Mgt. Sci., Vol. 16, No. 7.

29. Teichroew, Introduction to Management Science, Wiley, 1964, pp 295-321.

30. Topkis, "Optimal Ordering and Rationing Policies," Mgt. Sci., Vol. 15, No. 3.

31. Veinott and Wagner, "Computing Optimal (S, s) Inventory Policies," Mgt. Sci., Vol. 11, No. 5.

32. Veinott, "Optimal Policy for a Multi-Product Dynamic Non-Stationary Inventory Problem," Mgt. Sci., Vol. 12, No. 3.

33. Veinott, "The Status of Mathematical Inventory Theory," Mgt. Sci., Vol. 12, No. 11.

34. Veinott, "Minimum Concave Cost Solution of Leontief Substitution Models of Multi-Facility Inventory Systems," O.R., Vol. 17, No. 2.

35. Von Lanzenauer, "A Production Scheduling Model by Bivalent Linear Programming," Mgt. Sci., Vol. 17, No. 1.

36. Wagner and Whitin, "Dynamic Version of the Economic Lot Size Model," Mgt. Sci., Vol. 5, No. 1.

37. Wagner, O'Hagan and Lundh, "An Empirical Study of Exactly and Approximately Optimal Inventory Policies," Mgt. Sci., Vol. 11, No. 7.

38. Zangwill, "A Deterministic Multi-Period Production Scheduling Model with Backlogging," Mgt. Sci., Vol. 13, No. 1.

39. Zangwill, "A Backlogging Model and a Multi-Echelon Model of a Dynamic Lot Size Production System - A Network Approach," Mgt. Sci., Vol. 15, No. 9.

1. INTRODUCTION

The detailed scheduling decision involves the assignment of men and machines to specific operations during a given time interval. The scheduling of a large system involving hundreds of employees and work centers and thousands of jobs is incredibly complex, at least to academics. Surprisingly, Pounds [50] was unable to find anyone in industry who was responsible for detailed scheduling and recognized that he had a scheduling problem. Pounds infers that: "The job-shop scheduling problem is not recognized by most factory schedulers because for them, in most cases, no scheduling problem exists." Obviously, there are alternatives to precise, detailed, optimal scheduling.

The detailed scheduling problem is imbedded in two mutually exclusive environments, the job shop and the assembly line. Intermediate situations, such as a flow shop or batch processing on an assembly line will use methodology from one of these two settings.

In either setting, however, we will assume that there is a job generating process. This may be an inventory control system, a production scheduling system, or just customers walking in off the street.

2. THE JOB SHOP

Our initial section will examine job shop scheduling. The following definitions are required (Conway et al, [12]) A job is a collection or set of operations with a precedence ordering on the operations. An operations has three attributes; it is associated with a job, a machine, and a real number representing the processing time of the operation on the machine (or possibly a probability distribution). A machine is simply a time scale with intervals available for processing. A job shop is the set of all machines. Sequencing is determining the ordering of operations on a single machine. Scheduling is

assigning each operation of each job onto the time scale of a machine, within the precedence relations postulated, with no overlap of operations in an interval.

There are also several implicit assumptions which will be stated here and then taken as given. All operations are well defined and known for a job. All jobs must eventually be performed; the resources and facilities are entirely specified. The precedence relations are known.

The specific scheduling problem is classified by six attributes: the number of jobs and number of operations/job to be processed, the number and type of machines in the shop, the disciplines restricting assignment, the criteria for schedule evaluation, the arrival process of jobs, and whether operations are assumed to have deterministic or stochastic processing times.

Some further notation:

r_i = release time; for each job i, the time the job is released to the shop floor

d_i = due date (when the last operation should be completed)

$a_i = a_i - r_i$ = allowance for time in the shop

Each job has a set of g_i operations where

$$\begin{matrix} m_{i,1} \\ \vdots \\ m_i, g_i \end{matrix} \qquad \begin{matrix} p_{i,1} \\ \vdots \\ p_i, g_i \end{matrix}$$

where

$m_{i,j}$ = machine number to do the j^{th} operation of job i

p_{ij} = the corresponding processing time

$\sum_j P_{ij} = P_i$ is independent of our scheduling decisions and is assumed to include setup and teardown times. This means that the changeover time is independent of the sequence chosen (on a machine).

W_{ij} = waiting time before j^{th} operation of job i

C_i = completion time of job i = $r_i + \sum_j p_{ij} + \sum_j W_{ij}$

F_i = flow time of job i = $C_i - r_i$

L_i = lateness = $F_i - a_i = C_i - r_i$
(this can be positive or negative)

T_i = lateness = maximum $(0, L_i)$

E_i = earliness = maximum $(0, -L_i)$

Most theoretical works use simple measures of performance (M.O.P.) to increase the prospect of finding a solution. These are usually minimize the average or maximum of completion time, flow time, lateness, or tardiness. More complex criteria consider weighted sums of simple criteria, or involve the variance of these measures.

The details of a scheduling problem are usually abbreviated in a four parameter notation A/B/C/D where A describes the job arrival process, B the number of machines in the job shop, C the flow pattern in the shop, and D the criterion for evaluation or measure of performance.

A flow pattern describes the job transfer matrix, the percent of jobs transferred from machine i to j. A flow shop has positive entries in any row i only for $j > i$; a job shop (theoretically) has a completely dense matrix; the general shop has an arbitrary pattern. These are lettered F, R, and G respectively (figure 1).

The job arrival process is classified as static or dynamic. In the static case, all the jobs arrive simultaneously at time = T. This is signified by setting A to the number of jobs (usually 1, 2 or n, the general case). In the dynamic case, A identifies the probability distribution for the interarrival times of jobs.

The job shop may or may not be attached to a larger facility. If it is, or has regular customers, it often is a closed job shop. This indicates that operation masters (documents giving the job's routing, material requirements and specifications) already exist. Demand for certain products can be forecast with accuracy. This allows production to final goods inventory. In the open job shop, each job may be unique. A new operation master is written for each job. Instead of forecasting final good usage, machine work loads must be estimated.

Figure 1

(A) FLOW SHOP JOB TRANSFER MATRIX (F)

from machine \ to machine	1	2	3	4	5
1		.5	.3	.1	.1
2			.7	.2	.1
3				.6	.4
4					1.0
5					

(B) JOB SHOP (R)

from machine \ to machine	1	2	3	4	5
1		.2	.1	.4	.3
2	.1		.4	.4	.1
3	.2	.3		.4	.1
4	.1	.10	.5		.3
5	.2	.3	.2	.3	

(C) GENERAL SHOP (G)

from machine \ to machine	1	2	3	4	5
1			.4		.6
2	.2			.8	
3	.6	.1		.1	.2
4		.6	.4		
5	.1	.1	.1	.7	

3. JOB SHOP RESEARCH

Job Shop Research separates into analytical solution of "simple" models and simulation of realistic models.

A. The Sequencing Problem (Giffin, [24]). The problem that has been given the greatest theoretical attention is the one machine shop. Under varying conditions of job arrival, measures of performance, and assumptions on regular and overtime costs, this problem has been solved convincingly.

(Note that the rationale for using minimize $\bar{F}$, the mean flow time, as a measure of performance in sequencing research is that it minimizes work-in-process inventory.)

1) The static, n job, deterministic processing time, minimize mean flow time ($\bar{F}$) problem. This is the classic, and original, sequencing problem. The total elapsed time to complete the n jobs is sequence-independent, but arranging jobs in order of shortest processing time (S.P.T.) minimizes $\bar{F}$. If each job has a different \$ value, and therefore a different holding cost per unit time, the jobs should be arranged by weighted S.P.T. to minimize weighted flow time; or

$$(1) \quad \frac{P_{[1]}}{u_{[1]}} \leq \frac{P_{[2]}}{u_{[2]}} \leq \cdots \leq \frac{P_{[n]}}{u_{[n]}}$$

where $P_{[i]}$ is the processing time for the job in the i^{th} position and $u_{[i]}$ is its weight or \$ cost/unit time.

If the total processing time is sequence dependent (i.e., as in color changes in injection molding), we are faced with the traveling salesman problem and SPT is no longer appropriate. For this version, the usual M.O.P. is total processing time (F_{max}) which is equal to the sum of the processing times and the setup times)

$$(2) \quad F_{max} = F_{(n)} = \sum_{i=1}^{n} S_{(i-1)(i)} + \sum_{i=1}^{n} P_i$$

Since P_i is fixed, we are minimizing $\sum_{i=1}^{n} S_{(i-1)(i)}$ to min F_{max}. This problem has been solved by Little et al [39] using branch and bound.

2) The sequence-independent problem can be solved for intermittant arrivals (the dynamic case). More terminology is required. Pre-emption refers to interrupting a job on the machine to process another. Pre-empt-resume means that a job has the same total processing time despite being interrupted. Pre-empt-repeat means that all processing up to the interrupt is lost.

If pre-empt resume is assumed, SPT is still optimal. The partially completed job is treated as a new job with a processing time equal to its remaining time. When a new job arrives, the jobs are reordered and the current job pre-empted if appropriate. Under pre-empt-repeat, no general results exist. Information on impending arrivals is necessary for general rules.

3) If we further consider regular time, overtime and work shifts, an interesting situation arises; weighted SPT is no longer optimal in minimizing $\bar{F}$ (Gelders and Kleindorfer [23]). An infinite time horizon is now divided into periods of regular time, overtime, and down time. Using weighted SPT may mean that the next job can not be completed before the end of the regular time period, while a job with a higher weighted SPT could be completed in the remaining period. n! permutation schedules must be examined to find the optimal schedule.

4) Weighted tardiness as a measure of performance (in the static and dynamic cases): McNaughton [43], Schild and Fredman [53, 54], Held and Karp [27], Elmaghraby [19], Emmons [21], Srinivasan [56] and Shwimer [57], as well as others, have produced scheduling algorithms.

5) Two further model improvements are consideration of stochastic processing times and job priority classes [24]. In the K-class system, class 1 jobs have highest priority and are processed ahead of all jobs of a higher class (in general, jobs in class i are processed before jobs of class j, $j>1$, independent of waiting times). These models are amenable to queuing theory analysis, if the

service time and interarrival time distributions are judiciously chosen. When $\bar{F}$ with measure of performance (or work in process inventory or mean waiting time) the optimum priority assignment within a class is

$$\frac{E(t_1)}{C_1} \leq \frac{E(t_2)}{C_2} \leq \cdots \leq \frac{E(t_j)}{C_j}$$

where $E(t_i)$ is the expected processing time of the job in the i^{th} position and C_i its delay cost per unit time.

6) A further complication is required precedence among jobs (Conway et al, [15]). In the most general case, job b is required to precede job c, but b and c are not required to be adjacent. Conway, Maxwell and Miller have solved this problem for the static case to minimize mean flow time.

7) The following are examples of current research papers. Merten and Muller [44] consider minimizing the variance of flow time as a M.O.P. They show that the schedule that minimizes the variance of flow time is the antithesis of the schedule that minimizes the variance of waiting time (where a schedule $R = [i_1, i_2 \ldots i_n]$ has an antithesis schedule $R^1 = [i_n, i_{n-1}]$ where the jobs are reversed in sequence).

However, a procedure for finding minimum variance schedules is not obvious.

Lawler [37] proposes an algorithm to find optimal schedules for a sequencing problem with arbitrary job precedence constraints. Each job has a cost function based on its flow time and the sum of costs is to be minimized.

Balut [4] has solved the sequencing problem under stochastic setup and processing times with an objective of maximizing the number of early jobs. Surveys of the sequencing problem can be found in [24] and [25].

B. Parallel Machines [15]. The shop structure is enlarged to m identical parallel machines. Arrival is static. There are n jobs. Assume a job must be processed on a single machine. The problem is to partition n jobs into m subsets and determine the sequence for processing within each subset. Not surprisingly, the jobs can be ordered in terms of increasing processing time and then simply assigned to machines 1 to m in rotation (operationally, as soon as a machine becomes free, assign it to the job with minimum processing time). This rule minimizes mean flow time.

C. The Flow Shop [15]. This is the next level of complexity in scheduling. As from the definition, there can be several paths through the shop. However, all movement between machines within the shop must be in a uniform direction.

It is worthwhile noting that at this point analytical solution begins to pale before the combinatorial problems inherent to the multi-machine, multi-job case. Simulation becomes the major and most useful recourse.

1) The two machine flow shop - (a ubiquitous reference in any scheduling bibliography). Johnson [34] solves this problem under static arrival to minimize F_{max} of n jobs. Conway, et al [15] suggest that the importance of this work stems not from the actual algorithm which is intuitive, but first, from using F_{max} as a M.O.P. and, second, from proving optimality. Let A_i be the processing time (including setup) of the first operation of the i^{th} job. Similarly, B_i for the second operation. Then F_{max} is minimized when job j precedes job j + 1 if $\min (A_j, B_{j+1}) < \min (A_{j+1}, B_j)$. The rationale is to put the smallest A_i first so the second operations can begin as soon as possible and the smallest B_j last so the total processing can be completed as soon as possible after the last operation on machine 1 is finished since

$$F_{max} \geq \sum_{i=1}^{n} A_{[i]} + B_{[n]} \quad \text{and}$$

$$F_{max} \geq A_{[1]} + \sum_{i=1}^{n} B_{[i]}.$$

2) The same problem when $\overline{F}$ is used as the criterion becomes very difficult. Johnson's procedure is not optimal and no constructive algorithm comparable to Johnson's is known. Ignall and Schrage [29] have applied branch and bound technique to the problem. Unfortunately, their solution method doubles in computational difficulty each time n is increased by one. However, their solution also allows solution of the n job 3 machine problem with F_{max} as the M.O.P.

3) The m machine shop-Analytical work is scarce once beyond the three-machine flow shop. At this point, complete enumeration, branch and bound and integer programs are computationally ineffective. A problem with n jobs and m machines has $(n!)^{m-2}$ possible schedules ! For m = n = 6, this is a mere 2.7×10^{11} schedules. This brings us near the realm of difficulties unleased by the general n job m machine job shop problem.

The flow shop assumption reduces the search in that one only need consider schedules in which the same job order is followed on the first two machines (Conway et al, [15]). This is proved by contradiction; if not, a schedule could be improved by attaining this state. If F_{max} is the M.O.P., a stronger result is true; that only schedules with the same job order on machine 1 and 2, and m-1 and m have to be considered. The proof is similar.

Dudek and Smith [17], extending the work of Dudek and Teuton [18], have proposed an algorithm to minimize F_{max} for the n-job m-machine flow shop. However, only <u>permutation</u> schedules are considered; this is optimal under certain restrictive assumptions but not in general. (A permutation schedule has n possible first jobs, n-1 second, etc. for n! possible schedules.) Their algorithm begins with a "presequence" of scheduled jobs and then looks to extend the sequence by dominance tests. If a job passes all tests, it can be added to the sequence; otherwise, several sequences must be carried for further tests. Comparison between the number of sequences generated and the number generated in a total

enumeration shows the algorithm to be very effective. However, the problem formulation, again, is restrictive.

number of machines m	number of jobs n	number of problems worked	algorithm	enumeration (n!)
3	3	15	2.067	6
	4	20	5.000	24
	5	20	11.100	120
	6	20	18.350	720
	7	15	15.100	5040
	8	4	38.000	40320
5	3	15	2.333	6
	4	17	7.882	24
	5	18	21.444	120
	6	15	54.200	720
	7	3	80.670	5040

TABLE I [18]

Campbell, Dudek and Smith [8] have developed a heuristic to generate approximate solutions to the above (restricted) problem. Their algorithm is not computer-bound, adding to its desirability. The procedure generates (m-1) n-job two-machine problems, then solvable by Johnson's procedure, as follows: for machines 1 through K sum the processing times for each job; do the same for machines K + 1 to m. Then using Johnson's algorithm, find the best solution for F_{max}. Change K and repeat. Find the lowest F_{max} (K) and use that sequence.

This algorithm, again, is effective, giving an average error of 2.54% in sequence time (compared to optimal sequences) for 340 problems ranging in size from n = 3 and m = 3 to n = 7 m = 7. For larger problems (n and $m \geq 20$), the algorithm was superior to Palmer's heuristic [49]. Calculation time by hand varied from a few minutes for the n = 3, m = 3 problem to ~32 minutes for the n = 10, m = 15 problem. For permutation flow shop schedules, this is an effective heuristic solution procedure.

The general flow shop problem, however, remains combinatorially locked.

D. The General Job Shop (Conway et al, [15]). The most general problem is the scheduling of m jobs on n machines where each job may require processing on any number of machines in any sequence. A complete enumeration for the 5 machine 5 job problem is $(5!)^5$ schedules.

1) The two machine, two operations per job problem. Jackson [32] has extended Johnson's results to the general job shop. This is the only analytical solution in this section. Again, the M.O.P. used is F_{max} (for n static jobs). The jobs are partitioned in four groups; those with only one operation, on machine 1; those similarly on 2; those with 2 operations and sequence 1, 2; those similarly on 2, 1. The latter two are each ordered by Johnson's procedure, as if they were the entire work load. Ordering of jobs within the first two sets is arbitrary as it won't affect F_{max}. The optimal schedule is: on machine 1 run the jobs in [1, 2] before the jobs in [1], before the jobs in [2, 1]; on machine 2 [2, 1], [2], [1, 2]. This minimizes idle time on both machines 1 and 2. This is emphasized by imagining that there were no jobs in [2, 1]; the jobs in [1], [2] and [2] are still optimally ordered.

2) The Integer Programming Formulation of the General Job Shop. The formulation shown, also applicable to sections A, B and C, is by Manne [41]. Earlier formulations by Bowman [5] and Wagner [61] are more complicated.

Variables and constants:

a) P_{ik} = processing time of job i on machine k

b) r_{ijk} = 1: if j^{th} operation of job i requires machine k
0: otherwise

c) T_{ik} = starting time of job i on machine k

d) Y_{ijk} = 1: if job i precedes job k on machine k (not necessarily directly)
0: otherwise

Constraints:

a) only one job can be in process on a machine at any instant, i.e., either $T_{ij} - T_{jk} \geq P_{jk}$ or $T_{jk} - T_{ik} \geq P_{ik}$; so using the Y_{ijk} variables,

(1) $$[M + P_{jk}]\, Y_{ijk} + [T_{ik} - T_{jk}] \geq P_{jk}$$

(2) $$[M + P_{ik}]\, [1 - Y_{ijk}] + [T_{jk} - T_{ik}] \geq P_{ik}$$

where M is a large number such that only one constraint from 1 and 2 will be binding.

b) operational precedents are stated by observing that $\sum_k r_{ijk}\, T_{ik}$ is the starting time of the j^{th} operation of job i. Therefore, for all but the last operation of a job

(3) $$\sum_k r_{ijk}\, [T_{ik} + P_{ik}] \leq \sum_k r_{i,j+1,k}\, T_{ik}$$

This formulation gives:

nm variables T_{ik}

$\frac{m(n)\,(n-1)}{2}$ variables Y_{ijk}

(m-1)n equations of type a

$\frac{2(m)(n)(n-1)}{2}$ equations of type b

therefore, for the four machine, 10 job problem we have 220 variables, 390 constraints.

Objective functions:

a) minimize $\bar{F}$: this is the same as minimizing the sum of the start-times of the last operation of each job or minimize $\sum_i \sum_k r_{imk}\, T_{ik}$

b) minimize F_{max}: this requires an additional constraint

$$\sum_k r_{imk}\, (T_{ik} + P_{ik}) \leq F_{max}$$

and then the objective is to minimize F_{max}

c) minimize mean tardiness: the equation $T_i - E_i = F_i - a_i$ is added and the objective is to minimize $\sum_i T_i$.

This tool has not been used frequently due to the lack of fast I.P. codes. The size of the I.P. problem increases so rapidly as n and m increase that I.P. and branch and bound would not seem to be effective scheduling tools in the short run. A theoretical survey of the static general job shop problem (to minimize F_{max}) is given by Bakshi and Arora [3].

3) This brings us to the most effective and predominant research method in job shop scheduling investigations -- simulation. (Refer for the meaning of acronyms to Table I.)

Since the determination of a complete schedule for the n job general job shop is seemingly impossible, the problem is partitioned into sequencing problems at each machine. The sequencing problem involves assigning a priority to each job in a queue and then processing, in order of priority. These priorities move the jobs through the shop instead of a schedule. A dispatching rule (or priority assignment rule, for example, might be SPT. Most of the simulation research has been devoted to evaluating possible dispatching rules.

Jackson [33] has proved that the following are sufficient conditions to allow such decomposition <u>without</u> loss of optimality:

(1) the arrival of jobs is poisson

(2) the routing of a job depends only on a probability transition matrix

(3) the service time distribution of an operation is exponential

(4) the dispatching rule at a machine is independent of a job's routing and processing times

Unfortunately, these are slightly restrictive assumptions.

More terminology is required; a <u>local</u> dispatching rule requires information available at a single machine, a global rule requires information beyond that available at a single machine.

TABLE I

I. Usual Measures of Performance (either the mean, maximum, or variance of:)

a. flow time
b. tardiness
c. number of orders completed
d. percent of machine capacity utilized

(a. and b.'s relative importance may be gauged by a quotation from a job shop foreman: "I'd get reprimanded for high work-in-process inventory, but I'd get fired for too many late jobs.")

II. Several Dispatching Rules

a. SPT - shortest processing time
b. FCFS - first come, first served
c. SS - static slack -- the slack remaining when the job arrives at that machine where slack is defined as the (due date - present date)
d. variations of SS; SS/PT - static slack/remaining processing time; SS/RO - static slack remaining/number of operations
e. LCFS - last come, first served (first at the queue)
f. DS - dynamic slack - defined as [due date - (expected remaining processing time + present date)] and variants DS/PT, DS/RO
g. FISFS - first in the system (or shop), first served
h. COVERT - a rule that uses the ratio of delay cost to processing time, or c/t (c-over-t) to retain the benefits of SPT but reduce extreme lateness
i. RANDOM - priority assigned at random (used as a control)
j. DDATE - priority assigned on the basis of due date
k. LPT - longest processing time
l. MWKR - most work remaining (as Σ processing time)
m. WINQ - work in the next queue (the first global rule) - priority assigned on the basis of the sum of the processing times of the jobs in the queue that each job will next enter

(a more complete list (with rule rationales) is in Conway et al, [15], Chapter 11)

While simulation permits relaxation of almost all assumptions (as: no transit times from job to job, unlimited labor, several sequences possible) traditionally these have not been relaxed. While the labor-limited shop has been explored by Nelson [44], the majority of research assumes unlimited manpower, or a machine constrained shop.

Early research [7] was performed by Rowe [51, 52] (1958), Baker and Dzielinski [2] (1960), Conway, Maxwell, and Johnson [13] (1960). These initial studies confirmed the feasibility of using dispatching rules. An interesting note is the explanation for the simplicity of these initial models [15]. The original simulations were programmed in absolute machine language and then in symbolic assembly language, "a nontrivial programming task." When specific simulation languages became available in the early 1960's, larger studies followed. Two of the more massive were by Conway at RAND [11] and Nanot [46] at UCLA.

Fairly comprehensive results of simulation research are available in [15], Chapter 11 (the most complete survey up to 1967), and [7], Chapter 14.

To summarize some of these findings (under the assumption of unlimited labor):

1. SPT: SPT was found to consistently minimize mean flow time. However, since it discriminates against jobs with operations with long processing times, it has a high variance of flow time (several jobs wait for long periods).

Several schemes have been advanced to correct this deficiency. Conway and Maxwell [61] tested three approaches. They alternated the SPT rule with a

low variance rule (as FCFS) to "clean out the shop." This failed as the disadvantage of introducing the alternated rule (in increased flow time) were judged greater than its advantage (a decrease in variance of flow time).

prop. of time using SOT	mean flow time	variance:flow time
0 (FCFS)	244.5	30,423
.20	230.3	67,375
.40	223.7	60,550
.60	215.8	67,757
.80	211.3	85,072
1.00 (SPT)	205.9	88,695

Results for a pure job shop with 6 machines, a sample size of 2,000 jobs (20,000 for FCFS). From Conway and Maxwell [61].

TABLE III

The second approach was to truncate the SPT rule by imposing a limit on the waiting time for an operation (or job). This was moderately successful as seen in Table III.

rule	$\bar{F}$ (mean flow time)	variance, flow time
TS, 100	236.1	36,264
TS, 300	229.3	51,417
TS,	220.4	75,984
SPT	218.2	125,461

Results for a pure job shop with 6 machines, a sample size of 2,000 jobs. From Conway and Maxwell [12]. (TS means truncated SPT; the number is approximately the maximum waiting time in a queue.)

TABLE IV

Their third attempt (with Oldziey) [59] was to use a composite (global) rule that considered the job's due date, processing time and the congestion at all other queues. The result was a large decrease in mean tardiness. However, this rule, as most global rules, requires a tremendous amount of global information and the researchers conclude that possible gains may not be worth the computational and hardware (system) costs necessary to support the use of the rule.

Carroll [10] also tried to adjust the SPT rule to eliminate oultiers, extremely late jobs. This rule uses a ratio of tardiness or delay cost to processing time. The higher the potential delay cost, the higher the priority. The shorter the processing time, the higher the priority. This rule was superior to truncated SPT (T.S.) in reducing mean tardiness, though it increased $\overline{F}$ over T.S. results.

2. Global versus local rules: global rules allow consideration of the entire shop status (queues at other machines, the total amount of processing in the shop for each machine, the number of jobs and their processing times that will be arriving at each machine, and so on). If the shop is well utilized, global rules will not decrease machine idle time (as it will be zero or near zero already). However, under low utilization, these rules may decrease idle time. They can decrease congestion at queues by increasing the priority of jobs whose next operation will go to a shorter queue. While these (and other) benefits exist, global rules require an information retrieval and processing system (as noted in [12]) that may be more costly than can be justified by all the benefits achievable.

Local rules also imply decentralized control. The foreman or machine operator chooses the next job to be processed. Global rules mean centralized priority computation with greater dependency on equipment that can fail (or be sabotaged) -- usually a CPU and possibly remote data entry stations or terminals.

3. Multiple criteria (Buffa and Taubert [7]). Instead of using a single criterion, several weighted measures of performance can be totalled to rate a dispatching rules' performance. LeGrande [38] simulated a labor-limited shop using actual data (from Hughes Aircraft Co.) to compare six rules on the basis of 10 criteria. When all were equally weighted, SPT was superior, followed by DS/RO, FCFS, FISFS and then RANDOM.

DS/RO had the minimum number of orders completed late and the smallest variance of flow time. SPT lead in the categories of number of orders completed, average number of orders waiting in the shop (least), average waiting time of orders ($\bar{F}$), percent of labor utilized, percent of machine capacity utilized, and mean of the distribution of completions. DS/RO was more attractive than SPT if order completion criteria were most heavily weighted.

4. A cost based composite rule: this multiple criteria rule assigns costs or cost indices to each measure of performance to allow cost minimization for a shop. Work in process inventory, tardiness, facilities utilization and mean setup time are each translated into operational costs (or indices to denote relative cost structure).

The job priority is then the sum of these costs. The value of the approach is to change the relative weights into a relative cost framework in multiple criteria rules.

5. Labor limited shops. Nelson [47] has developed labor assignment rules and tested them in conjunction with dispatching rules. The three dispatching rules used were FCFS, FISFS, and SPT. The labor assignment rules were:

(a) random assignment of idle labor to any machine with work in queue

(b) assignment to the machine with the most jobs in queue

(c), (d), (e) assignment according to the labor- and machine-limited systems counterpart of the (FISFS, FCFS, SPT) queue discipline for machine-limited systems (i.e., send the labor to the queue with the highest priority job under the machine-limited rule used)

Labor assignment was controlled by a parameter that varied the frequency of assignment (from whenever a worker has no jobs remaining at his present machine, to after completion of each operation at a machine).

The combination (b) - SOT had minimum mean flow time while (b) - FISFS has a lower variance and maximum flow time.

The use of full central control (after each operation) decreased mean, maximum and the variance of flow time over less frequent reassignments.

Fryer [22], in an important extension, has further clarified the importance of labor assignment rules. Increasing the organization complexity, he investigated a job shop composed of three divisions, each division consisting of four work centers. The policy decisions involved transferring men to other divisions as well as intradivisional reassignment among work centers. Two dispatching rules (SPT, FCFS) were used to sequence jobs. Fryer found that the interdivisional reassignment policy (flexible versus restricted) had a greater effect on mean flow time than any other policy choices. Decision rules concerning to which specific division to reassign an eligible worker had little effect on performance measures. The intradivisional reassignment policy (flexible versus restrictive) had a major affect on flow time variance. Again, the decision rules on which particular work center to assign a worker were relatively unimportant in M.O.P. reduction.

This study clarifies the importance of allowing labor reassignment independent of the specifics of the reassignment rule. The effects of dispatching rules were consistent with previous research.

In essence, the total job shop scheduling problem should include both a queue and labor assignment discipline.

Alternatives to Global Scheduling

There are several alternatives to scheduling besides use of dispatching rules. Prior to use of dispatching rules, Gantt charts were employed. These were simply time lines for each machine upon which the jobs were laid out, operation by operation, until a feasible schedule was reached. These schedules suffered from swift obsolescence.

Other alternatives are keeping average delivery times long or renegotiating due dates when necessary. This obviates the need for global rules or loading exercises. The shop can also work at low utilization so bottlenecks rarely occur. Lastly, the job status system can be improved so that expeditors can push jobs through as necessary.

Most of these alternatives result in either lower utilization of equipment and labor or higher work in process inventory (or both), but full scheduling is not always possible (due to complexity) or desirable (due to its cost). One example of both conditions is found in the hospital, where "rational scheduling rules" are hindered by the uncertainties of patient care, the complexity of the shop (hospital facilities) and the potential cost of scheduling research and equipment.

Prerequisites for "Successful" Scheduling

Desirable measure of performance levels depend not only on detailed scheduling methodology, but on many higher level decisions. In the case of the closed shop, capacity (or aggregate) planning is required for seasonal items. Proper release of jobs to the shop by use of the run-out-time list is assumed. If these two higher levels are ignored or the decisions made in error, the lower level must suffer accordingly. If jobs "suffocate" the shop, it may be a result of excessive job releasing or insufficient manpower allocated. The opposite is also possible.

In the case of the open job shop, loading (to infinite or finite capacity) is requisite for the planning of job due dates and release times. Aggregate planning is possible (if the shop has a seasonal workload) by using forecasts of machine loads in hours.

II. The Assembly Line

1. Definitions: The assembly line is the extreme case of the flow shop; machines are arranged to manufacture one product or product-type. The line can then be considered as a single machine, the balance determining its output rate. This rate can only be changed by rebalancing the line. Machines are often physically contiguous and capacities of manufacturing stages set to allow near continuous uniform product flow.

The following definitions will be useful (Kilbridge and Webster [35]:

1) A work station is a location where one or more assigned tasks are performed by one or more operators.

2) A task is an indivisible work activity; an activity that could not be split between two operators; each task has a processing time associated with it (specifying the amount of time required to perform the task).

3) Precedence relationships define the allowable processing sequences of tasks.

4) Zoning constraints and special constraints restrict groupings of tasks and limit specified tasks to given work centers respectively.

5) The aggregate production rate gives the required output of the entire line in units per hour of the product to be manufactured.

6) The natural cycle time is calculated from 5) as the maximum production time/per unit that still fulfills the aggregate output rate.

7) The work content of a product is the sum of its tasks' processing times (similarly for work station job content).

8) The cycle time is the maximum of all the work centers' job contents; the cycle time sets the production rate of the line; one unit of the product is finished each cycle time.

The classical assembly line balancing problem is to partition tasks among work stations within all precedence relationships, special constraints and zoning constraints to minimize or maximize some criterion. The maximum potential daily output of the line must also be greater than or equal to the aggregate production rate per day.

2. Measures of Performance: The usual criteria are to minimize either idle time or the number of work stations. Less common criteria are to minimize the variance of work station job contents or the total labor cost. In published research, the first criterion, idle time [defined as (the number of work stations)* (the cycle time) - (the work content of the product)] is often discussed, but rarely used (Ignall [30]). Instead, the number of work stations is minimized assuming that the cycle time must be less than or equal to the natural cycle time. In practice, operators at work stations with smaller job contents than the cycle time will not actually stand idle; they will work continuously at a slower pace. The effect, however, in terms of labor cost is the same as if they were idle part of the time and worked at their normal pace during the remainder of the cycle time.

The reason all other criteria are shunned by most researchers is quite practical; as was found in job shop scheduling, the judicious choice of M.O.P. can measurably improve the prospect of finding a solution. This problem, again, has a combinatorial nature. If the objective is to minimize the number of work stations (given a cycle time), rather than to minimize idle time, considerably less search is necessary (Ignall [30]). To illustrate: if U1→UN are tasks and A and B two partial balances, where A = |U1 U2 U6| |U4 U5|... and B = |U1 U2 U6 U5| |U4 U3|... (|-| denotes a work station), than B dominates A when minimizing the number of work stations. B has one less task to assign to its remaining work stations; therefore A does not have to be fully evaluated.

However, if idle time is considered, A must be fully explored. Similarly, when considering variance of both idle time and work content, there are more possible balances to evaluate.

To minimize cost, idle time is found for a range of k = 1 to n work stations (n is bounded by the processing time of the largest task). Normally, a small number of work stations yields a lower idle time. There are more tasks per station increasing the chance for an excellent fit. If the cycle time for a balance is greater than the natural cycle time (N.C.T.), a second shift or overtime is required. If it is less than the N.C.T., the line must shut down for some fraction of the day. The costs for these adjustment may be added to the regular time labor cost for each number of stations and the minimum total is selected.

In practice, the number of work stations is always minimized, also minimizing the number of operators required for one shift. If the solution has high idle time, tasks, precedence relations, or constraints may be redefined. Helgason and Birnie have found that in their experience any balance can be improved upon by an experienced industrial engineer. This implies that a good starting balance at low computational cost may be superior to an excellent balance at high computational cost.

3. Solution Techniques

Solution techniques can be partitioned into five classes. These are: complete enumeration, integer programming, heuristic procedures, branch and bound (or best bud) and dynamic programming.

a) Complete enumeration - this procedure is severely limited by the dimensions of the problem. If there are N tasks and r precedence relations, there are approximately $N!/2^r$ feasible sequences of tasks. This is usually too large a number to consider (even though further reduced by the fact that interchanging certain elements within a work station results in the same balance but a different

feasible sequence). The number of sequences is definitely finite but, so far, no simpler method for assembly line balancing has been suggested. Jackson [31] constructs all feasible first work stations, eliminates those dominated, then for each remaining first work station constructs all feasible second station combinations, etc. This procedure is optimal but soon bogs down for problems with 30 to 40 tasks with few precedence relations.

b) Integer Programming - another optimal but computationally infeasible procedure for large scale problems. Bowman [6] has developed two formulations; however, a five-task illustrative problem has 20 inequality constraints and 10 variables. The number of variables and constraints unfortunately increases nonlinearly with the number of tasks.

c) Heuristics - as in job shop research, heuristic solutions have proved effective and numerous. Besides their computational efficiency, they often allow less restrictive assumptions than optimal techniques. Often optimal techniques are modified into heuristic solutions (as by Held, Karp and Shareshian [28]). The following are several of the more famous heuristics:

I. Kilbridge and Webster [35] - their method allows line balancing without computer assistance. The precedence relationships are translated into columns; the first column has all tasks with no precedessor, the second their immediate follower tasks, and so on. In addition, each task in a column, has the maximum column number that it could be moved into without changing any precedence relationships. Any tasks that would also have to be moved are similarly listed. The heuristic then adds the elements within column I, II and on until the total task time is as close as possible to the cycle time desired. If there is a gap, assigned tasks are moved into higher numbered columns and/or more tasks are added from the present or next column. There are suggested procedures for moving and selecting tasks. This procedure also allows consideration of zoning constraints and other "special" conditions. While tedious, it is one of the few

techniques that is both effective in general situations and does not rely on computer availability.

II. Arcus [1] - (COMSOAL - Computer Method of Sequencing Operations for Assembly Lines). Arcus uses three lists; list A with each task and its number of immediately preceding tasks, list B (the available list) - a list of all tasks from A with no immediate precedessors and list C (the fit list), those tasks from list B whose processing time is less than or equal to the time remaining at the work station being assigned tasks.

Tasks are selected from list C by a biased sampling procedure, and lists A and B updated until a balance is obtained. (When C is empty and tasks remain on B, a new work station is started.)

Balances are generated with little computational effort, allowing a great number to be generated at low cost. If r% of all balances are good, then the probability of generating a good one is $[1 - (1 - r)^n]$ where n is the number of trials (assuming a new balance is generated each trial). Obviously, as n gets very large, the probability approaches 1.

The sampling is biased by certain rules that produce "better" balances; for example, giving larger tasks and tasks with many successors a greater probability of being selected from the C list. Arcus's method also permits consideration of more complex problems (as Kilbridge and Webster's method.

III. Tonge [59] - Tonge extends Arcus's method by changing the probability of choosing a rule to choose tasks for inclusion in a station. Successful application of a rule increases its probability of future selection and vice versa. Some of the nine rules are: choose the task with the largest time; the greatest number of successors; at random. This learning theoretic approach does not seem superior to Arcus's method, however.

While there are other heuristic techniques, Mastor's study [42] (of 16 methods under varying problem size (number tasks), number of precedence

relations, and line length (number of stations)) indicates that for large complex problems Arcus's method is the most effective.

d) Dynamic Programming - In the same study [42], the technique that proved most effective for moderate sized problems with more restrictive assumptions was dynamic programming. Held, Karp, and Sharesian [28] use a dynamic programming formulation for small problems and a heuristic incorporating dynamic programming to solve subproblems of larger problems.

In the exact formulation, a subset is defined as a grcup of tasks where for any task in the subset, all its necessary precedessors are also in that subset. A sequence is an ordered set where the tasks in that subset are ordered feasibly for execution. There may be several sequences per subset possible. The optimal balance for each subset can be found and the states increased until the entire problem is solved.

In the heuristic procedure, tasks are grouped by certain rules and then these groupings are treated as tasks in the exact solution method. Such heuristics are necessary as the problem size increases; in the most severe case of no precedence restrictions, the number of alternatives to evaluate for K tasks would be $K \cdot 2^{k-1}$ (the number of pairs([subset S],[subset S with one task deleted]) used in Held's et al recurrence relation $= \sum_{t=1}^{k} t\binom{k}{t} = k2^{k-1}$

Another approach in this vein is shortest route techniques. Klein [36] and Gutjhar and Nemhauser [26] both use this formulation with a solution procedure akin to dynamic programming. And again, computationally, these approaches are not practical for large balancing problems.

e) Branch and Bound - a variant of this technique, best bud, has been used by Nevins [48] to solve large problems. While branch and bound is optimal, computationally, it would be infeasible. Best bud is not optimal but, in practice, has proved extremely successful. Its approach is similar to branch and bound,

partitioning a problem into subproblems. Instead of continuing along a path (partial balance) until the path is fathomed (rejected as above the upper bound or completely evaluated and producing a new upper bound), the path with the best "bud score" is followed when a path's score is reevaluated (after "growing" a new work station). It is followed if still low score or left incomplete if an alternate partial path has a better score. A path's score is equal to $(T^* - T)/(N^* - N)$ where T^* is the total work content; T the sum of assigned tasks to that path; N the number of work stations already assigned, N^* the number of work stations allowed in our balance. This score represents the average time that must be assigned to the remaining work stations to obtain a solution.

4. The Real Problem

This completes a short survey of solution techniques to the classical assembly line balancing problem. However, while this problem is seemingly solved, it is not the real line balancing problem, just as the static sequencing problem fell short of solving the real job shop problem. The analogy is appropriate because the same two elements are absent: consideration of stochastic processing times and a product mix (instead of a machine mix). Moodie [45] was the first to consider stochastic processing times; he minimizes $\sum_k (C - E(S_k) + r\sqrt{v(S_k)})$ for a k station balance where C is the cycle time, r is a constant chosen as a safety factor; S_k is the work content of the k^{th} station (V_k its variance and $E(S_k)$ its mean); task processing times are assumed to be independent normal random variables. This reduces the problem into the deterministic framework.

The more pressing problem of several products sharing a line remains; the single product line is the exception rather than the rule. This problem has recently received attention in the work of Thomopoulos [57,58] and Macaskill [40]. The method of solution is similar to that for single product lines. However, work elements are assigned to stations on a daily or shift by shift basis rather

than on a cycle time basis. Usually individual model task assignments are not considered; as there are different quantities of each model to be scheduled that shift, more aggregate "tasks" are considered. A task time becomes the total task work time (for that quantity of model i to be assembled) for task $i = t_i \times N_j$ where N_j = number of units of model j to be completed.

"Tasks" are then assigned to stations within all precedence relationships to minimize the sum of the idle time at each station over all stations. Idle time at a station is equal to the length of the shift minus the work content.

Further improvements are possible to smooth the flow of each model along the line, since the above procedure may lead to very uneven flows of work on any individual model.

Industrial concerns, particularly those in the appliance and automotive industries were forerunners in developing computer programs to handle mixed line balancing. Such companies include International Harvester Corporation (Capretta, [9]), and Whirlpool Corporation (Moodie [45]).

The final difficulties are at a higher level of decision making: when to use an assembly line instead of a job shop setting; when to redesign products to take advantage of manufacturing possibilities. These questions are crucial and much harder to answer. They involve factors which are not easily quantified (as employee satisfaction) but which must be considered. A further discussion is beyond the scope of this survey, but the problem has been noted.

Other complications also outside this survey's scope are local regulation of the assembly line speed (for conveyer belt type lines) by workers, unequal labor abilities among workers, and redefining tasks or precedence relationships to improve balances.

REFERENCES

1. Arcus, "An Analysis of a Computer Method of Sequencing Assembly Line Operations," Ph.D. Thesis, University of California, Berkely, 1963.

2. Baker and Dzielinski, "Simulation of a Simplified Job Shop," Mgt. Sci., Vol. 6, No. 3, April, 1960.

3. Bakshi and Arora, "The Sequencing Problem," Mgt. Sci., Vol. 16, No. 4.

4. Balut, "Scheduling to Minimize the Number of Late Jobs when Set-up and Processing Times are Uncertain," Mgt. Sci., Vol. 19, No. 11.

5. Bowman, "The Schedule-Sequencing Problem," O.R., Vol. 7, No. 5.

6. Bowman, "Assembly Line Balancing by Linear Programming," O.R., Vol. 8, No. 3.

7. Buffa and Taubert, Production-Inventory Systems: Planning and Control, Irwin, 1972.

8. Campbell, Dudek, and Smith, "A Heuristic Algorithm for the n-Job m Machine Sequencing Problem,: Mgt. Sci., Vol. 16, No. 10.

9. Capretta, "Operations Analysis in Four Manufacturing Problems," Thirteenth Annual Conference and Convention-Proceedings of the American Institute of Industrial Engineers, 1962.

10. Carroll, "Heuristic Sequencing of Single and Multiple Component Jobs," Ph.D. Thesis, MIT, June, 1965.

11. Conway, "An Experimental Investigation of Priority Assignment in a Job Shop," RAND Corp. Memo. RM-3789-PR, Feb., 1964.

12. Conway and Maxwell, "Network Scheduling by the SPT Discipline," OR, Vol. 10, 1962, 51-73.

13. Conway, Maxwell and Johnson, "An Experimental Investigation of Priority Dispatching," J.I.E., Vol. 11, No. 3, May, 1960.

14. Conway, Maxwell and Oldziey, "Sequencing Against Due-Dates," Proceedings of IFORS Conference, Cambridge, Mass., Sept., 1966.

15. Conway, Maxwell and Miller, Theory of Scheduling, Addison-Wesley, 1967.

16. Day and Hottenstein, "Review of Sequencing Research," NRLQ, Vol. 17, No. 1, March, 1970.

17. Dudek and Smith, "A General Algorithm for Solution of the n-Job, m-Machine Sequencing Problem of the Flow Shop," O.R., Vol. 15, No. 1.

18. Dudek and Teuton, "Development of M-Stage Decision Rule for Scheduling n Jobs through m Machines," O.R., Vol. 12, No. 3.

19. Elmaghraby, "The One Machine Sequencing Problem with Delay Cost," Journal of Industrial Engin., Vol. XIX, No. 2, Feb., 1968.

20. Elmaghraby, "The Machine Sequencing Problem - Review and Extensions," NRLQ, Vol. 15, June, 1968.

21. Emmons, "One Machine Sequencing to Minimize Certain Functions of Job Tardiness," O.R., Vol. 17, No. 4, July-Aug., 1969.

22. Fryer, "Operating Policies in Multiechelon Dual-Constraint Job Shops," Mgt. Sci., Vol. 19, No. 9.

23. Gelders and Kleindorfer, "Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop: I-theory," MIT, Sloan School Working Paper, April, 1972.

24. Giffin, Introduction to Operations Engineering, Irwin, 1971.

25. Groff and Muth, Operations Management: Analysis for Decisions, Irwin, 1972.

26. Gutjhar and Nemhauser, "An Algorithm for the LIne Balancing Problem," Mgt. Sci., Vol 11, No. 2, 1964.

27. Held and Karp, "A Dynamic Programming Approach to Sequencing Problems," Journal of the Society for Industrial and Applied Mathematics, Vol. 10, No. 1, March, 1962.

28. Held, Karp and Shareshian, "Assembly-Line Balancing -- Dynamic Programming with Precedence Constraints," O.R., Vol. 11, No. 3, 1963.

29. Ignall and Schrage, "Application of the Branch-and-Bound Technique to Some Flow Shop Scheduling Problems," O.R., Vol. 13, No. 3.

30. Ignall, "A Review of Assembly LIne Balancing," J.I.E., Vol. 16, No. 4.

31. Jackson, "A Computing Procedure for a Line Balancing Problem," Mgt. Sci., Vol. 2, No. 3, 1956.

32. Jackson, "An Extension of Johnson's Results on Job-Lot Scheduling," NRLQ, Vol. 3, No. 3.

33. Jackson, "Jobshop-Like Queuing Systems," Research Report 81, Management Sciences Research Project, UCLA, Jan., 1963.

34. Johnson, "Optimal Two- and Three-State Production Schedules with Setup Times Included," NRLQ, Vol. 1, No. 1, March, 1954.

35. Kilbridge and Wester, "A Heurstic Method of Assembly-Line Balancing," J.I.E., Vol. XII, No. 4, July-Aug., 1961.

36. Klein, "On Assembly Line Balancing," O.R., Vol. 11, No. 2, 1963.

37. Lawler, "Optimal Sequencing of a Single Machine Subject to Precedence Constraints," Mgt. Sci., Vol. 19, No. 5.

38. LeGrande, "The Development of a Factory Simulation System Using Actual Operating Data," Mgt. Technology, Vol. 3, No. 1, May, 1963.

39. Little, Murty, Sweeny and Karel, "An Algorithm for the Traveling-Salesman Problem," O.R., Vol. 11, No. 6.

40. Macaskill, "Production-Line Balances for Mixed-Model Lines," Mgt. Sci., Vol. 19, No. 4.

41. Manne, "On the Job-Shop Scheduling Problem," O.R., Vol. 8, No. 2.

42. Mastor, "An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques, Ph.D. Thesis, UCLA, 1966.

43. McNaughton, "Scheduline with Deadlines and Loss Functions," Mgt. Sci., Vol. 6, No. 1, Sept., 1959.

44. Merten and Muller, "Variance Minimization in Single Machine Sequencing Problems," Mgt. Sci., Vol. 18, No. 9.

45. Moodie, "A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work-Element Times," Ph.D. Thesis, Purdue, 1964.

46. Nanot, "An Experimental Investigation and Comparative Evaluation of Priority Disciplines in Job Shop-Like Queueing Networks," Ph.D. Thesis, UCLA, 1963.

47. Nelson, "Labor and Machine Limited Production Systems," Mgt. Sci., Vol. 13, No. 9.

48. Nevins, "Assembly Line Balancing Using Best Bud Search," Mgt. Sci., Vol. 18, No. 9, May, 1972.

49. Palmer, "Sequencing Jobs through a Multi-Stage Process in the Minimum Total Time - a Quick Method of Obtaining a Near Optimum," ORQ, Vol. 16, No. 1.

50. Pounds, "The Scheduling Environment," Industrial Scheduling, Chap. I.

51. Rowe, "Sequential Decision Rules in Production Scheduling," Ph.D. Thesis, UCLA, 1958.

52. ____, "Towards a Theory of Scheduling," J.I.E., Vol. 11, March, 1960.

53. Schild and Fredman, "On Scheduling Tasks with Associated Linear Loss Function," Mgt. Sci., Vol. 7, No. 3, April, 1961.

54. _____, "Scheduling Tasks with Non-Linear Loss Functions," Mgt. Sci., Vol. 9, No. 1, Sept., 1962.

55. Shwimer, "On the N-Job, One-Machine, Sequence Independent Problem with Tardiness Penalties: A Branch and Bound Approach," Mgt. Sci., Vol. 18, No. 6, Feb., 1972.

56. Srinivasan, "A Hybrid Algorithm for the One-Machine Sequencing Problem to Minimize Total Tardiness," Mgt. Science Report No. 225 G.S.I.A. (Carnegie-Mellon University, Nov., 1970).

57. Thomopoulos, "Line Balancing - Sequencing for Mixed Model Assembly," Mgt. Sci., Oct., 1967.

58. _____, "Mixed Model Line Balancing with Smoothed Station Assignments," Mgt. Sci., May, 1970.

59. Tonge, "Assembly Line Balancing Using Probabilistic Combinations of Heuristics," Mgt. Sci., Vol. 11, No. 7, 1965.

60. Wagner, "An Integer Linear-Programming Model for Machine Scheduling," NRLQ, Vol. 6, No. 2.

INTEGRATING INVENTORY CONTROL WITH DETAILED SCHEDULING IN THE PRODUCTION ENVIRONMENT

In the section on detailed scheduling, a demand generating process was assumed, independent of the scheduling algorithm utilized. For the closed job shop and/or the assembly line, this generating process may be an inventory control system. Jobs are then released in order of their run-out-time.

Separation of the two decision processes may be suboptimal. Run-out-time releasing ignores the job shop status (farsighted decision process). Job shop scheduling ignores the inventory status of all "jobs" not yet released (shortsighted decision process). At what cost can a happy medium be attained?

Von Lanzenauer [35] has formulated a joint model to determine the production at each stage for each product in each period for a multi-stage production facility to minimize the cost of set-ups, inventory and shortages. The actual formulation is similar in spirit to Manne's in the job shop scheduling section. However, the problem is restricted by dividing a machine's time scale into discrete intervals or periods. At most one product can be processed in a period on a machine and a machine must operate for an entire period or not at all. This restriction is only realistic when periods are short, increasing the number of variables drastically in an already infeasible computationally integer program.

Work force smoothing is ignored in this formulation; the machines are the only constraining resource. Shwimer [26] formulates a more complete integrated model, including work force smoothing. However, he quickly points out its computational drawbacks and, instead, concentrates on its structure. The result (further expounded in Hax [14]) is an iterative model where production scheduling decisions are input to a detailed scheduling simulation. Tardiness, inventory levels, flow times, machine utilization and other measures of performance are evaluated for the released load. These are then used to alter the production scheduling decision. Iterations continue until some criterion is met.

In the short run, this approach is more realistic than integrated models. However, the integrated models yield structural insights that may be valuable in deciding partitioning procedures for joint problems, while the heuristics give benchmark solutions to be bettered by alternate approaches.