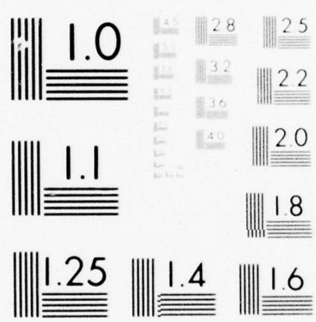
AD-A045 016 MASSACHUSETTS INST OF TECH CAMBRIDGE OPERATIONS RESE--ETC F/G 5/1
HIERARCHICAL PRODUCTION PLANNING SYSTEMS.(U)
AUG 77 A C HAX, J J GOLOVIN N00014-75-C-0556
UNCLASSIFIED TR-135 NL

1 OF 1
AD A045016

END
DATE
FILMED
11-77
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

HIERARCHICAL PRODUCTION PLANNING SYSTEMS

by

ARNOLDO C. HAX and NATHAN J. GOLOVIN

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 135	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 HIERARCHICAL PRODUCTION PLANNING SYSTEMS.		5. TYPE OF REPORT & PERIOD COVERED 9 Technical Report August 1977
		6. PERFORMING ORG. REPORT NUMBER 14 TR-135
7. AUTHOR(s) 10 Arnoldo C. Hax Jonathan J. Golovin		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-75-C-0556
9. PERFORMING ORGANIZATION NAME AND ADDRESS M.I.T. Operations Research Center 77 Massachusetts Avenue Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 347-027
11. CONTROLLING OFFICE NAME AND ADDRESS O.R. Branch, ONR Navy Dept. 800 North Quincy Street Arlington, VA 22217		12. REPORT DATE 11 August 1977
		13. NUMBER OF PAGES 33 pages 12 36p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Releasable without limitation on dissemination.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

270 720

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Hierarchical Planning Systems
Demand Forecasting

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See page 1.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

HIERARCHICAL PRODUCTION PLANNING SYSTEMS

by

ARNOLDO C. HAX

and

JONATHAN J. GOLOVIN

Technical Report No. 135

Work Performed Under

Contract N00014-75-C-0556, Office of Naval Research

Multilevel Logistics Organization Models

NR 347-027 M.I.T. OSP 82491

Operations Research Center

Massachusetts Institute of Technology

Cambridge, Massachusetts 02139

August 1977

Abstract. This ~~chapter~~ report describes the development of hierarchical planning systems to support medium range planning and operational decisions in a batch processing production environment. In this approach, higher level decisions impose constraints to lower level actions, and lower level decisions provide the necessary feedback to reevaluate higher level actions. An analysis of the existing methodology to design hierarchical production systems is given. Computational results are presented.

1. *Introduction*

Production can be defined as the process of converting raw materials into finished products. An effective management of the production process should provide the finished products in appropriate quantities, at the desired times, of the required quality, and at a reasonable cost.

Production management encompasses a large number of decisions that affect several organizational echelons. These decisions can be grouped into three broad categories: (1) strategic decisions, involving policy formulation, capital investment decisions, and design of physical facilities; (2) tactical decisions, dealing primarily with aggregate production planning; and (3) operational decisions, concerning detailed production scheduling issues. These three categories of decisions differ markedly in terms of level of management responsibility and interaction, scope of the decision, level of detail of the required information, length of the planning horizon needed to assess the consequences of each decision, and degree of uncertainties and risks inherent in each decision. These considerations have led us to favor a hierarchical planning system to support production management decisions, which guarantees an appropriate coordination of the overall decision making process, but, at the same time, recognizes the intrinsic characteristics of each decision level. A justification for this hierarchical approach and its implications for the design of a production system has been reported by Hax [11]. Early motivation for this approach can be found in the pioneering work of Holt, Modigliani, Muth, and Simon [15], and in Winters [20]. Hax [3] described an application of a hierarchical production system for a continuous manufacturing process. Hax and Meal [14], and Bitran and Hax [1] addressed the use of hiearchical systems in a batch processing environment. Armstrong and Hax [3], and Shwimer [19] analyzed an application for a job shop activity. Recent theoretical research in the field of hierarchical production planning systems has been conducted by Golovin [10], Gabbay [9], and Candea [6].

This chapter discusses the general issues associated with the design of hierarchical production planning systems. An overall description of the characteristics of such systems is given in Section 2. Section 3 analyzes the aggregate production planning decisions. Section 4 justifies the need for hierarchical planning systems. Section 5 presents the treatment of demand forecasts. Sections 6 and 7 discuss the most important methodologies proposed to disaggregate higher level decisions. Finally, Section 8 provides computational results comparing the efficiency of the various disaggregation methodologies.

2. *A Hierarchical Production Planning System*

Production decisions involve complex choices among a large number of alternatives. These choices have to be made by trading-off conflicting objectives under the presence of financial, technological, and marketing constraints. Such decisions are not trivial and model based systems have proven to be of great assistance in supporting managerial actions in this field. In fact, one could argue that, in this respect, production is the most mature field of management. A great many contributions have been made in this field by operations research, system analysis, and computer sciences. But now we believe it is both significant and feasible to attempt a more comprehensive and integrative approach to production management.

The optimal planning and scheduling of multiple products has received much attention in the operations research literature. Several attempts (Manne [17], Dzielinski, Baker, and Manne [7], Dzielinski and Gomory [8], Lasdon and Terjung [16]) have been made to formulate the overall problem as a single mixed-integer mathematical programming model to be solved on a rolling horizon basis. However, these approaches require data such as the forecast demand for every item over a complete seasonal cycle, usually a full year. When these systems involve the scheduling of several thousands of items, these data requirements become overwhelming and the resulting planning process becomes unrealistic due to the magnitude of the forecast errors inherent in such detailed long term forecasts.

The obvious alternative to a detailed monolithic approach to production planning is a hierarchical approach. The basic design questions of a hierarchical planning system are the partitioning of the overall planning problem and the linkage of the resulting subproblems. An important input to resolve these questions is the number of levels recognized in the product structure. Hax and Meal [14] identified three different levels:

(i) Items are the final products to be delivered to the customers. They represent the highest degree of specificity regarding the manufactured products. A given product may generate a large number of items differing in terms of characteristics such as color, packaging, labels, accessories, size, etc.

(ii) Families are groups of items which share a common manufacturing setup cost. Economies of scale are accomplished by jointly replenishing items belonging to the same family.

(iii) Types are groups of families whose production quantities are to be determined by an aggregate production plan. Families belonging to a type normally have similar costs per unit of production time, and similar seasonal demand patterns.

We have found that these three levels are necessary to characterize the product structure in many batch processing manufacturing environments we have examined. Obviously, there are some practical applications where there are some practical applications where additional or fewer levels might be needed. In the remainder of this paper we will propose a hierarchical planning system based on these three levels of item aggregation. Note, however, that conceptually the system can be extended to any number of aggregation levels by defining appropriate subproblems linking those levels.

The first step in our hierarchical planning approach is to allocate the total production capacity among product types by means of an aggregate planning model. The planning horizon of this model covers normally a full year in order to properly consider the fluctuating demand requirements for the products. We advocate the use of a linear programming model at this level. There are various advantages associated with the use of such a model that will be addressed in the next section. The major drawback is that a linear programming model does not take setup costs into consideration. The implications of this limitation will be examined in detail in Section 7.

The second step in the planning process is to allocate the production quantities for each product type among the families belonging to that type. This is done by disaggregating the results of the aggregate planning model but only for the first period of the planning horizon, thus substantially reducing the required amount of data collection and data processing. The disaggregation procedure used assures consistency and feasibility among the type and family production decisions while attempting to minimize the total setup costs incurred in the production of families. It is at this stage where setup costs are now explicitly considered.

Finally, the family production allocation is divided among the items belonging to each family. The objective of this decision is to maintain all items at an inventory level that maximizes the time between family setups. Again, consistency and feasibility are the driving constraints of this disaggregation process.

An extensive justification for this approach is provided in Hax [11] and Hax and Meal [14]. Under certain cost structures, it has been shown to be optimal (Gabbay [9]). Under more general cost structures, it has empirically been found to perform exceptionally well, as discussed in Section 7. Figure 1 shows the overall conceptualization of the hierarchi-

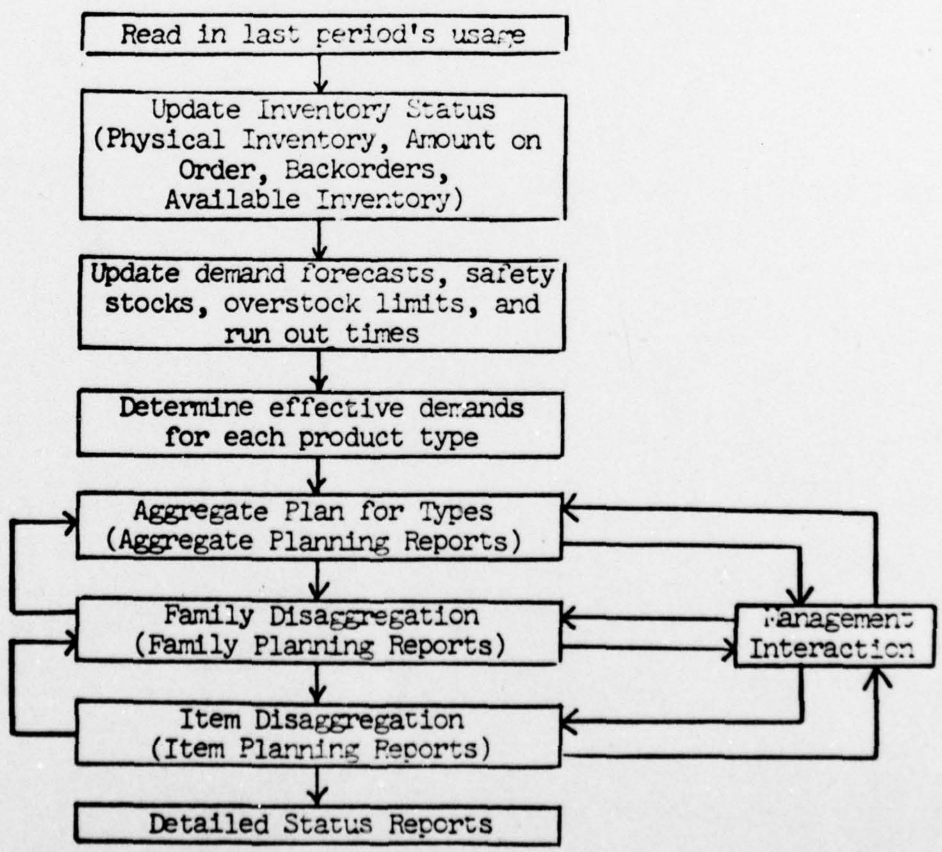


FIGURE 1. CONCEPTUAL OVERVIEW OF HIERARCHICAL PLANNING SYSTEM

cal planning effort. A computer based system has been developed to facilitate its implementation. The details of this system are reported in the next chapter of this book (Hax and Golovin [13]). Herein we will concentrate on the methodological issues associated with the system design.

3. *Aggregate Production Planning for Types*

This is the highest level of planning in the production system, addressed at the type level. Essentially any aggregate production planning model can be used as long as it adequately represents the practical problem under consideration. (For extensive discussions of possible aggregate models, see Buffa and Taubert [5], Hax [12], and Silver [18].) We consider the following simplified linear program at this level:

$$\text{Minimize} \sum_{i=1}^{I} \sum_{t=1}^{T} (c_{it}X_{it} + h_{it}I_{it}) + \sum_{t=1}^{T} (r_t R_t + o_t O_t)$$

subject to:

$$X_{it} - I_{i,t+L} + I_{i,t+L-1} = d_{i,t+L} \quad , \quad i=1,\ldots,I;\ t=1,\ldots,T$$

$$\sum_{i=1}^{I} m_i X_{it} \le O_t + R_t \quad , \quad t=1,\ldots,T$$

$$R_t \le (rm)_t \quad , \quad t=1,\ldots,T$$

$$O_t \le (om)_t \quad , \quad t=1,\ldots,T$$

$$X_{it} \ge 0 \quad , \quad i=1,\ldots,I;\ t=1,\ldots,T$$

$$I_{i,t} \ge 0 \quad , \quad i=1,\ldots,I;\ t=L+1,\ldots,L+T$$

$$R_t, O_t \ge 0 \quad , \quad t=1,\ldots,T$$

The decision variables of the model are: X_{it}, the number of units to be produced of type i during t; $I_{i,t+L}$, the number of units of inventory remaining at the end of period t+L; R_t and O_t, the regular and overtime hours used during period t, respectively.

The parameters of the model are: T, the length of the planning horizon; L, the length of the production lead time; c_{it}, the unit production cost (excluding labor); h_{it}, the inventory carrying cost per unit, per period; r_t and o_t, the cost per manhour of regular and overtime labor, respectively; $(rm)_t$ and $(om)_t$, the total availability of regular and overtime hours in period t, respectively; and m_i, the inverse of the productivity rate for type i, in hours/unit. $d_{i,t+L}$ is the effective demand for type i during period t+L. (For a definition of effective demand, see section 5.)

Whenever the production costs c_{it} are invariant with time, no backorders are allowed, and the regular work force payroll is a fixed commitment, the terms $\sum_{i=1}^{I} \sum_{t=1}^{T} c_{it}X_{it}$ and $\sum_{t=1}^{T} r_tR_t$ are fixed and should be deleted from the objective function. In that case, the model seeks the optimum aggregate plan trading off inventory holding and overtime costs. It is straightforward to extend the model to include other cost factors and decisions, such as hiring and firing, backorders, subcontracting, lost sales, etc. Also, the constraints can represent any number of technological, financial, marketing restrictions, or other considerations.

Linear programming is a convenient model to use at this aggregate level due to both its computational efficiency and the wide availability of linear programming codes. In addition, L.P. permits sensitivity and parametric analyses to be performed quite easily. The shadow price information that becomes available when solving L.P. models can be of assistance in identifying opportunities for capacity expansions, market penetrations, introduction of new products, etc.

Notice that the manufacturing setup costs have been purposely ignored in this aggregate model formulation. In practice, we have found that setup costs have a secondary impact on the total production cost (see Section 8 for sensitivity analysis). Moreover, the inclusion of setup costs would force the model to be defined at a family level. This implies a high level of detail which invalidates all the advantages of hierarchical planning discussed in the next section. Consequently, setup costs are considered only at the second level of the hierarchical planning process.

Because of the uncertainties present in the planning process, only the first time period's production plan is implemented. At the end of each time period, new information becomes available that is used to update the model with a rolling planning horizon of length T. Therefore, the data transmitted from the type to the family level is the resulting product typce production and inventory quantities for the first period of the aggregate model. These quantities will then be disaggregated among the families belonging to each corresponding type.

4. *The Need for Hierarchical Planning*

The advantages of the aggregate approach as compared to a detailed one may now be clearer. These advantages can be divided into three distinct categories.

The first category considers the costs of data collection to support the model as well as the computational cost of running the model. A major information system may be required to collect the demand productivity and cost data as well as prepare forecasts for thousands of individual items, a more costly project than building the production planning system itself. This data must then be reviewed by management. As the number of items increases, this effort can become unwieldy, leading to deterioration of the data used in the planning process and therefore the output. In most cases, this cost of data collection and preparation will far outweigh the cost of computation. This is important to note as the cost of computation continues to decrease and it becomes feasible to solve enormous linear or non-linear programming problems. Aggregation of items can significantly reduce the cost and effort in demand forecasting and data preparation in addition to decreasing the computational costs.

The second category considers the accuracy of the data. Unless all items are perfectly correlated, an aggregate forecast of demand will have reduced variance. In general, we are able to employ more sophisticated techniques such as econometric models or auto regressive-moving average statistical models and spend more time in obtaining managerial judgment, given the smaller number of forecasts required. Since decisions on regular time, overtime, hiring and firing, and other production rate changes are based on the <u>total</u> production quantity demanded, increased forecast accuracy on total demand should improve the decision making process.

Finally, and perhaps from an implementation standpoint, most importantly, aggregation leads to more effective managerial understanding of the model's results. When ten thousand items are being planned simultaneously, the sensitivity of the results to changes in individual item demnds may be complex. There are too many combinations of changes to consider. The manager may never be able to see the overall picture but, instead, be lost in the details.

In addition, at this level of managerial planning, most marketing forecasts are made by product group and decisions made by product line or manpower class. These are budgeting decisions, not lot sizing decisions for next week. It is crucial that the decision variables and sensitivity analysis that can be carried out correspond to those with which the manager deals.

While all these arguments for aggregation are valid, they would be meaningless if it were not possible to disaggregate back to the detailed level and obtain near optimal results from this hierarchical approach. Our results, discussed in Section 8 have shown hierarchical systems to be near optimal under a variety of realistic conditions.

5. *Demand Forecasts*

Unless care is taken, the use of aggregation may lead to infeasibilities. It is important to realize that inventories and demand only have physical meaning at the item level. The concept of product types is a mere abstraction that makes possible the aggregation process. When calculating product type inventories, it is incorrect to simply add the inventories of all the items belonging to a product type. Implicitly, that practice assumes complete interchangeability of the inventories among all the items in a product type, which is not the case. To illustrate this point, consider a product type consisting of items 1 and 2, whose initial inventories and demand requirements for the next five periods are as follows:

		Demand by Period				
	Initial Inventory	1	2	3	4	5
Item 1	600	100	100	200	200	400
Item 2	100	200	200	400	400	800
Total	700	300	300	600	600	1200

By simply considering total product type demand and inventory, we would calculate net demands of 0, 0, 500, 600, and 1200. But, in fact, we will run out of item 2 in periods 1 and 2. The problem arose from assuming that we could use product type inventory held in item 1 for item 2. This problem is corrected by defining <u>effective demands</u> for each item.

Formally, if $\bar{d}_{k,t}$ is the forecast demand for item k in period t, AI_k is its corresponding available inventory, and SS_k is its safety stock, the effective demand $d_{k,t}$ of item i for period t is given by:

$$\begin{aligned} d_{k,t} &= \max\left(0, \sum_{\ell=1}^{t} \bar{d}_{k,\ell} - AI_k + SS_k\right) &&, \quad t=1,2,\ldots,t^* \\ d_{k,t} &= \bar{d}_{k,t} &&, \quad t=t^*+1,\ldots,T \end{aligned} \tag{1}$$

where t^* is the first time period in which the initial inventory is depleted, i.e.

$$\sum_{\ell=1}^{t^*-1} \bar{d}_{k,\ell} - AI_k + SS_k \leq 0 \qquad , \text{ and}$$

$$\sum_{\ell=1}^{t^*} \bar{d}_{k,\ell} - AI_k + SS_k > 0$$

The effective demand for a type i is simply given by the sum of the effective demands for all items belonging to a given type, i.e.

$$d_{i,t} = \sum_{k \in K(i)} d_{k,t} \tag{2}$$

where K(i) is the set of all items k belonging to product type i.

In our previous example, the effective demands are:

	Effective Demand by Period				
	1	2	3	4	5
Item 1	0	0	0	0	400
Item 2	100	200	400	400	800
Total Effective Demand for Product Type	100	200	400	400	1200

The hierarchical forecasting system operates as follows:

(1) An aggregate forecast is generated for each product type for each

time period in the planning horizon.

(ii) The type forecasts are disaggregated down to item forecasts. This disaggregation can be done by forecasting the proportion of the total type demand corresponding to each item. These proportions can be updated using exponential smoothing techniques, which are appropriate to apply for a short horizon at a detailed level. Notice that item and family forecasts are only required for a few time periods in the product type disaggregation models we will present.

(iii) After updating the available inventory for each item, the effective item demand is calculated by applying expression (1) above. Whenever the initial available inventory exceeds the first period's demand, expression (1) requires item forecasts for successive periods in the planning horizon. These forecasts can be obtained by making trend and/or seasonability adjustments to the initial period forecasts, again using exponential smoothing techniques.

(iv) The effective demand for types is obtained from expression (2). These demands are used in the aggregate model described in section 3. Computer programs to perform automatically the necessary calculations are discussed in the next chapter.

Note that this forecasting system is an example of a top down approach, using aggregate product type forecasts and disaggregating, rather than using detailed forecasts and summing to get the aggregate product type forecasts.

6. *The Family Disaggregation Model*

The central condition to be satisfied at this planning level for a coherent disaggregation is that the sum of the productions of the families in a product type equals the amount dictated by the higher level model (plan) for this type. This will assure consistency between the aggregate production plan and the family disaggregation process. This consistency should be achieved while determining the run quantities for each family that minimize the total setup cost among families, the remaining cost to be considered.

We will now examine four disaggregation methods which have been proposed in the literature: Hax and Meal [14], Knapsack [1], Winters [20], and Equalization of Run Out Times.

6.1 *Hax and Meal Method*

Conceptually, Hax and Meal [14] suggested a heuristic approach to family disaggregation that:

(i) schedules those families in each product type that must be run in the current planning period in order to meet the items' service requirements;

(ii) sets initial family run quantities so as to minimize cycle inventory and setup costs. (This is accomplished by setting the initial family run quantity equal to the corresponding family economic order quantity.);

(iii) then adjusts the run quantitites of the families so as to use all the production time allocated to each product type by the aggregate planning model, while observing items' overstock limits.

To implement these three conditions, Hax and Meal proposed an algorithm with the following rules:

(i) Only those families which trigger during the current planning period have to be scheduled for production. A family is said to trigger whenever the current available inventory of any of its items cannot absorb its expected demand during its production lead time plus the review period, i.e. whenever

$$AI_k < (d_{k,1} + d_{k,2} + \dots + d_{k,L+1}) + SS_k \quad \text{for any } k\epsilon K(j),$$

where K(j) is the set of all items k belonging to family j.

Equivalently, we can define the run out time of an item k to be

$$ROT_k = \frac{AI_k - SS_k}{\sum_{t=1}^{L+1} d_{k,t}}$$

A family j is said to trigger whenever its run out time for at least one item is less than one time period, i.e.

$$ROT_j = \underset{k\epsilon K(j)}{\text{Min}} ROT_k = \underset{k\epsilon K(j)}{\text{Min}} \frac{AI_k - SS_k}{\sum_{t=1}^{L+1} d_{k,t}} < 1$$

If we do not produce this family j some member item will run out, violating our assumption in the aggregate plan of no backorders.

(ii) The initial run quantity, Y_j, for a family that has triggered, is set to the minimum between its economic order quantity, EOQ_j; and the difference between the overstock limit of the family, OS_j, and its current available inventory, AI_j; i.e.

$$Y_j = \text{Min}\,(EOQ_j, OS_j - AI_j)$$

where

$$AI_j = \sum_{k\epsilon K(j)} AI_k \quad ,$$

$$OS_j = \sum_{k\epsilon K(j)} OS_k \quad \text{, and}$$

EOQ_j can be determined by the lot size formula for a family of related items (Brown [4], page 47). When an item k has a terminal demand at the end of a season, OS_k can be calculated by means of a newsboy model (see Zimmermann and Sovereign [21], page 370).

(iii) If the sum of the initial run quantities for the families belonging to a product type i does not add up to the total production time allocated to that type by the aggregate planning model, adjustments in the family run quantities are needed.

Let J(i) be the set of all families belonging to product type i, and X_i be the total production to be allocated among these families. X_i^* has been determined by the aggregate planning model, and corresponds to the optimal value of variable X_{i1} since only the first period result of the aggregate model is to be implemented. Two cases should be considered:

(a) If $\sum_{j\varepsilon J(i)} Y_j < X_i^*$, the new run quantities Y_j^* for each families are:

$$Y_j^* = \mathrm{Min}\left[\sum_{k\varepsilon K(j)} (OS_k-AI_k),\ Y_j + (X_i^* - \sum_{j\varepsilon J(i)} Y_j)\ \frac{\sum_{k\varepsilon K(j)} (OS_k-AI_k)}{\sum_{j\varepsilon J(i)} \sum_{k\varepsilon K(j)} (OS_k-AI_k)}\right]$$

This simply states that the initial run quantities are expanded in proportion to the difference between the overstock limits and the current available inventory. This difference corresponds to the maximum allowable production of each family, up to its overstock limit. If all the families that have triggered must be produced to their overstock limit, and the total production is still less than the aggregate production requirement X_i^*, one should go deeper in the run out list scheduling families belonging to product type i in order of increasing run out time until we reach the total assigned production capacity. The new families should be run up to their maximum allowable quantities.

(b) If $\sum_{j\epsilon J(i)} Y_j > X_i^*$, the run quantities are decreased in proportion to their initial assignments, i.e.

$$Y_j^* = X_i^* \frac{Y_j}{\sum_{j\epsilon J(i)} Y_j}$$

For further details about this methodology, the reader is referred to [14].

6.2 *Knapsack Method*

Bitran and Hax [1] proposed a disaggregation technique which essentially is a formalization of the heuristics developed by Hax and Meal. For every product type i the following convex knapsack problem is solved:

$$\text{Minimize} \quad \sum_{j\epsilon J(i)} \frac{s_j d_j}{Y_j}$$

$$\text{subject to:} \quad \sum_{j\epsilon J(i)} Y_j = X_i^*$$

$$\ell b_j \leq Y_j \leq ub_j, \qquad j\epsilon J(i) \tag{3}$$

where Y_j is the number of units to be produced of family j, s_j is the setup cost for family j, d_j is the forecast demand for family j (usually an annual forecast demand), ℓb_j and ub_j are lower and upper bounds for the quantity Y_j, and X_i^* is the total amount to be allocated among all the families belonging to type i.

The lower bound ℓb_j, that defines the minimum production quantity for family j, is given by:

$$\ell b_j = \max \left[0, (d_{j,1} + d_{j,2} + \ldots + d_{j,L+1}) - AI_j + SS_j\right] .$$

This lower bound, ℓb_j, the minimum production to avoid backorders given current forecasts, guarantees that any backorders will be due only to forecast errors greater than the safety stock SS_j.

The upper bound ub_j is given by:

$$ub_j = OS_j - AI_j$$

where OS_j is the overstock limit of family j.

The objective function assumes that the family run quantities should be proportional to the setup cost and annual demand for a given family. This seems to be a reasonable assumption (and is the basis of the economic order quantity formulation), that tends to minimize the average annual setup cost. Notice that the total inventory carrying cost has already been established in the aggregate planning model; therefore it does not enter in the current formulation.

The first constraint:

$$\sum_{j \in J(i)} Y_j = X_i^*$$

assures the equality between the aggregate model input X_i^* and the sum of the family run quantitites. It can be shown (see Bitran and Hax [1]) that this condition can be substituted by

$$\sum_{j \in J(i)} Y_j \leq X_i^* \qquad (4)$$

without changing the optimum solution to the disaggregation problem. Intuitively, the larger the Y_j's, the smaller the objective function value and so the constraint is always met exactly.

Initially J(i) contains only those families which trigger during the current planning period. The production for these families must be scheduled in this period to avoid future backorders. All other families are put on a secondary list. These families will be scheduled only if extra capacity is available.

Bitran and Hax [1] presented an efficient algorithm to solve this problem through a relaxation procedure. Optimality and convergence proofs are given in [1] and [2]. The algorithm consists in ignoring initially the bounding constraints (3) and solving the objective function subject to the knapsack restriction (4). Then a check is made to verify if the optimum values Y_j^* satisfy the bounds (3). If they do, the Y_j^*'s constitute the optimal solution. If not, at least some of the Y_j^*'s are shown to be optimal and a new iteration takes place. The algorithm is finite because at each iteration we determine the run quantity of at least one family.

6.3 *Winters Method*

Winters [20] examined various alternatives for disaggregating the aggregate production quantities. He recommended a disaggregation procedure in which families are produced in economic order quantities, in order of their increasing run out times, until the aggregate total is reached. Unlike the Hax and Meal method, the initial run quantities are not modified, but are treated as indivisible discrete units and their release point is varied.

Using the terminology described previously, the Winters disaggregation method can be formalized as follows:

(i) Compute the run quantity for each family j:

$$Y_j = \text{Min}\,(EOQ_j,\ OS_j - AI_j).$$

(ii) Compute the run out time for each family j:

$$ROT_j = \underset{k \varepsilon K(j)}{\mathrm{Min}} \frac{AI_k - SS_k}{d_k}$$

where d_k is the forecast demand for item k.

(iii) Rank all the families belonging to a given product type by increasing run out times. This constitutes the run out list for a type.

(iv) For each product type, go down the run out list accumulating Y_j's until the total desired production X_i^* is reached; i.e., produce families 1 to n in the run out list where:

$$\sum_{j=1}^{n} Y_j \geq X_i^* \quad \text{, and}$$

$$\sum_{j=1}^{n-1} Y_j < X_i^*$$

Further discussion on this approach is provided in [20].

6.4 *Equalization of Run Out Times*

An obvious alternative disaggregation method is to allocate the production amount determined at the aggregate planning level for a given type in such a way as to equalize the run out times of all the items belonging to that type. This implies skipping the family level as a disaggregation layer. Run out time equalization is a natural disaggregation methodology to be applied at the item level and, therefore, the corresponding technical details will be presented in the next section.

It is important to mention at this point that when run out time equalization is directly applied at the item level, no consideration is given to the resulting setup costs associated with the family runs. Thus, it might be expected that this disaggregation procedure will generate fairly high setup costs, relative to the other family disaggregation methods which take explicit account of setup costs. The possible advantages of a direct item run out time equalization are the realization of a high degree of synchronization of the production planning system, and the added simplicity in implementing the hierarchical system.

7. *The Item Disaggregation Model*

For the current planning period, all the costs have been already determined in the former two levels and any feasible disaggregation of a family run quantity has the same total cost. However, the feasible solution chosen will establish the initial conditions for the next period and therefore will affect future costs. In order to save setups in future periods it seems reasonable to distribute the family run quantity among its items in such a way that the item run out times coincide with the run out time of the family. A direct consequence is that all items of a family will trigger simultaneously, minimizing remnant stock, the remaining inventory held in the items in that family.

7.2 *A Heuristic Approach*

Hax and Meal [14] proposed a heuristic algorithm to equalize the run out times of the items belonging to each family. The essence of this approach is to allocate the family run quantity, Y_j^*, so as to maximize the expected time until an item in that family runs out. Any item running out requires scheduling the entire family again and, therefore, should be deferred as long as possible within the constraints of item overstock limits and the total family production quantity determined at the family disaggregation level.

An initial run quantity is determined for each item by means of the expression:

$$Z_k = d_k \left[\frac{Y_j^* + \sum_{k \epsilon K(j)} (AI_k - SS_k)}{\sum_{k \epsilon K(j)} d_k} \right] + SS_k - AI_k \qquad (5)$$

where Z_k is the number of units to be produced of item k; AI_k and SS_k, are, respectively, the available inventory and safety stock of item k; d_k is the forecast demand for item k; K(j) is the set of indices of all the items belonging to family j; and Y_j^* is the total amount to be allocated for all items belonging to family j. Y_j^* was determined by the family disaggregation model.

Notice that the new run out time for item k will be:

$$ROT_k = \frac{Z_k + AI_k - SS_k}{d_k}$$

and, by equation (5), this is equal to

$$ROT_k = \frac{Y_j^* + \sum_{k \epsilon K(j)} (AI_k - SS_k)}{\sum_{k \epsilon K(j)} d_k}$$

which is constant for every item k. This equalizes the expected runout time for all the items in the family. Moreover, summing each side of Equation (5) over all k values gives us:

$$\sum_{k \epsilon K(j)} Z_k = Y_j^*$$

and, therefore, guarantees that the total amount allocated to the family, Y_j^* has been allocated among the items belonging to that family.

The resulting run quantities must be tested for negativity and against the overstock limits for each item. If the item run quantity does not lie between these limits it is set to zero or to the overstock limit, as appropriate. The normalizing constant,

$$\frac{Y_j^* + \sum_{k\epsilon K(j)} (AI_k - SS_k)}{\sum_{k\epsilon K(j)} d_k}$$

is appropriately modified by eliminating that item from the summations and the procedure is repeated again for the remaining items.

7.2 *Knapsack Approach*

Bitran and Hax [1] formalized the heuristic approach by formulating the run out time equalization problem as the following strictly convex knapsack problem for each family j:

$$\text{Minimize } \tfrac{1}{2} \sum_{k\epsilon K(j)} \left[\frac{Y_j^* + \sum_{k\epsilon K(j)} (AI_k - SS_k)}{\sum_{k\epsilon K(j)} \sum_{t=1}^{L+1} d_{k,t}} - \frac{Z_k + AI_k - SS_k}{\sum_{t=1}^{L+1} d_{k,t}} \right]^2$$

subject to:

$$\sum_{k\epsilon K(j)} Z_k = Y_j^*$$

$$Z_k \leq OS_k - AI_k$$

$$Z_k \geq \max \left[0, \sum_{t=1}^{L+1} d_{k,t} - AI_k + SS_k \right]$$

The first constraint of this problem requires consistency in the disaggregation from family to items. The last two constraints are the upper and lower bounds for the item run quantities. These bounds are similar to those defined for the knapsack family disaggregation model in the previous section.

The two terms inside the square bracket of the objective function represent, respectively, the run out time for family j, and the run out time for an item k belonging to family j (assuming perfect forecast). The minimization of the square of the differences of the run out times will make those quantities as close as possible. The term ½ in front of the objective function is just a computational convenience.

An algorithm to solve this problem follows very closely the logic presented in the family disaggregation algorithm. Details are given in Bitran and Hax [1] and will not be presented here.

8. *Computational Results*

We conducted a series of experiments to examine the performance of the hierarchical system under various conditions including size of forecast errors, capacity availability, magnitude of setup costs, length of the planning horizon, and disaggregation methodology used from product type to family levels.

The data used for these tests were obtained from a major manufacturer of rubber tires. The product structure characteristics and other relevant information are given in Figure 2. Table 1 exhibits the demand pattern for both product types. Product type 1 had a terminal demand season (corresponding to the requirements of snow tires), and consisted of 2 families and 5 items. Product type 2 had highly fluctuating demand throughout the year and consisted of 3 families and 6 items. Families were groups of items sharing the same molds in the tire curing presses, and therefore, sharing a common setup cost. Items, for instance, were white wall and regular wall tires of the same class. Families and items have the same cost charac-

Product Type 1: P1 | Product Type 2: P2

Product Type 1: P1	Product Type 2: P2
Family setup cost = $90	Family setup cost = $120
Holding cost = $.31/unit a month	Holding cost = $.40/unit a month
Overtime cost = $9.5/hour	Overtime cost = $9.5/hour
Productivity factor = .1 hr/unit	Productivity factor = .2 hrs/unit
Production lead time = 1 month	Production lead time = 1 month

Regular Workforce Costs and Unit Production Costs are considered fixed costs.

Total Regular Workforce = 2000 hrs/month
Total Overtime Workforce = 1200 hrs/month

FIGURE 2. PRODUCT STRUCTURE AND OTHER RELEVANT INFORMATION

TIME PERIOD t	PRODUCT TYPE 1 P1	PRODUCT TYPE 2 P2
1	12,736	6,174
2	7,813	2,855
3	0	4,023
4	0	4,860
5	0	7.131
6	0	9,665
7	1,545	17,603
8	7,895	14,276
9	10,982	11,706
10	15,782	15,056
11	16,870	8,232
12	15,870	7,880
13	9,878	10,762
TOTAL	99,371	120,223

TABLE 1. DEMAND PATTERNS OF PRODUCT TYPES

teristics and the identical productivity rates of their corresponding product types.

The experiments applied various hierarchical production planning systems under varying conditions for a full year of simulated plant operations. Production decisions were made every four weeks at which time a report was generated identifying aggregate as well as detailed decisions. The model was then updated and rerun using a one year rolling planning horizon. This process was repeated 13 times. At the end of the simulation, the total setup costs, inventory holding costs, overtime costs, and backorders were listed. A summary of eleven different simulation runs is provided in Table 2. The simulations were implemented on the Computer Based Operations Management System (COMS) developed at M.I.T. (see Hax and Golovin [13]).

Run 1 can be regarded as the base case: no forecast errors, a planning horizon of one year divided into 13 periods of 4 week durations each, "normal" capacity (defined as 2000 hours of regular time and 1200 hours of overtime per period), "normal" setup costs ($90 for families belonging to product type 1, and $120 per family belonging to product type 2). All the other runs varied some characteristics of Run 1.

8.1 *Difference in Performance of Family Disaggregation Methodologies*

We tested the performance of hierarchical planning systems using four different family disaggregation methodologies: Hax and Meal, Knapsack, Winters, and Equalization of Run Out Times. At the item level, we limited ourselves to use the heuristics approach proposed by Hax and Meal for equalization of run out times. A careful analysis of Table 2 indicates that no significant differences of perofrmance seem to exist among the four tested methodologies. Hax and Meal, and Knapsack have a slight

FAMILY DISAGGREGATION METHODOLOGY	COST COMPONENTS	1 BASE CASE NO FORECAST ERROR	2 10% FORECAST ERROR	3 30% FORECAST ERROR	4 HIGH SETUP COST CASE I P1: 5000,50 P2: 400,400,1000 NO FORECAST ERROR	5 HIGH SETUP COST CASE II P1: 6000, 4500 P2: 400,5000,3000 NO FORECAST ERROR	6 TIGHT CAPACI 1660 Reg NO FORECAS
HAX-MEAL	Setup	5580	5580	5580	67050	106400	4590
	Holding	72507	72795	75954	72451	72060	115154
	Overtime	81092	81415	82365	81072	81957	117430
	Total	159179	159790	163899	220593	260417	23717
	Backorders	0	2398	6460	0	0	0
KNAPSACK	Setup	5360	5360	5250	67050	104800	4480
	Holding	72510	73611	76577	72374	72597	115072
	Overtime	81111	81425	82365	81111	81102	117439
	Total	158981	160396	164192	220535	258449	236991
	Backorders	2	1513	6243	10	4	144
WINTERS	Setup	5800	5690	5470	68850	107000	5690
	Holding	73382	74915	76988	72861	71889	115058
	Overtime	81643	81406	82384	81102	82821	117420
	Total	160825	162011	164842	222813	261710	238168
	Backorders	0	17	10618	0	0	0
EQUALIZATION OF RUN OUT TIME	Setup	5910	5910	5910	68850	107000	5910
	Holding	72533	73047	75447	72533	72533	115069
	Overtime	81102	81415	80912	81102	81102	117439
	Total	159545	160372	162269	222485	260635	238418
	Backorders	14	2398	9965	14	14	145

TABLE 2. SUMMARY OF COMPUTATIONAL RESULTS WITH DIFFERENT HIERARCHICAL PRODUCTION PLANNING SYSTEMS

ST) 3000 RROR	6 TIGHT CAPACITY 1660 Reg. Hrs. NO FORECAST ERROR	7 LOOSE CAPACITY 2500 Reg. Hrs. NO FORECAST ERROR	8 6 MONTH PLANNING HORIZON- ORIGINAL NO FORECAST ERROR	9 6 MONTH PLANNING HORIZON- TIGHT CAPACITY NO FORECAST ERROR	10 6 MONTH PLANNING HORIZON- NORMAL CAPACITY 10% FORECAST ERROR	11 1-1-1-1-3-6 PLANNING HORIZON- NORMAL CAPACITY NO FORECAST ERROR
	4590 115154 117430 237174 0	5910 56029 48650 110589 0	5910 67172 88578 161660 0	5910 63272 94250 163432 40603	5910 64969 90773 161652 2956	5580 77878 75430 158888 383
	4480 115072 117439 236991 144	5910 56002 48507 110419 0	5960 67212 88597 161499 0	5030 83983 103332 192345 16658	5960 64971 90773 161434 2951	5250 78052 75440 158742 73
	5690 115058 117420 238168 0	5800 55233 49115 110148 3677	5910 68030 88588 162528 0	4920 67400 108927 181247 28765	5910 66637 90839 163386 17	5910 80250 75886 162046 0
	5910 115069 117439 238418 145	5910 56122 48517 110549 0	5910 67095 88597 161602 14	5910 83980 103332 193222 16662	5910 64969 90773 161652 2956	5910 72533 81102 159545 14

advantage over Winters and Equalization of Run Out Times, but the differences in cost do not constitute major gains. The Equalization of Run Out Times procedure gives the highest setup costs, as expected. It is not possible to infer, with this limited amount of experimentation, whether or not a given disaggregation methodology could offer some specific advantages under certain given conditions. Research in progress is designed to cast some light on these issues.

8.2 *Sensitivity to Forecast Errors*

Runs 1, 2, and 3 show the impact of forecast errors on the production planning decisions. Forecast errors are uniformly distributed in intervals of the type [-a,+a] and are introduced in all three levels. Moreover, at the family and item levels we guarantee that the demands of families in the same product type and the demands of items in a family add to the demand of the product type and family respectively. As one would have expected, the quality of the decisions deteriorates under increasing forecast errors. Both cost and size of backorders increase when forecast errors begin to escalate. However, the system performs reasonably well even under forecast errors of up to 30%, included in Run 3. (The 6243 units backordered in Run 3 of the Knapsack case represent a 97% service level.) This is an important justification for the hierarchical approach since, obviously, aggregate forecasts can be more accurate than detailed forecasts.

8.3 *Sensitivity to Changes in Setup Costs*

The values imputed to the setup costs in the base case (Run 1) were realistic measures of the actual setup costs incurred in the normal manufacturing operations. They included direct setup costs (manpower and materials), as well as opportunity costs for having the machines idle while performing the changeover. We wanted to test the system's performance under extreme conditions which represented unusually high setup costs. With this purpose in mind we made two different runs, Runs 4 and 5, with the following setup cost characteristics:

	Setup Costs				
	Type 1		Type 2		
	Family 1	Family 2	Family 1	Family 2	Family 3
Run 4	5000	50	400	400	1000
Run 5	6000	4500	400	5000	3000
Base Case - Run 1	90	90	120	120	120

Naturally, the total cost associated with Runs 4 and 5 increases significantly. It can be observed that Runs 1, 4, and 5 are almost identical in terms of inventory holding costs and overtime costs, which indicate that the overall production strategies for these runs do not change much. This is to be expected as the aggregate plan does not see the increase in setup costs. This could be a limitation of this particular hierarchical approach when applied to situations with extremely high setup costs, since, under these conditions, one could have expected higher inventory accumulation to obtain a better trade off between inventory and setup costs.

8.4 *Sensitivity to Capacity Availability*

Runs 6 and 7 evaluate the performance of the system under different capacity conditions. Run 6 decreases the regular capacity to 1660 hours per period; Run 7 expands the regular capacity to 2500 hours (as opposed to 2000 hours in the base case). As one could see from the results in Table 2, the system's performance is quite sensitive to capacity changes. Under tight capacity, there is a significant increase in both costs and backorders; the opposite is true under loose capacity. Clearly, the system can be useful in evaluating proposals for capacity expansion.

8.5 *Sensitivity to Changes in Planning Horizon Characteristics*

Runs 8, 9, 10, and 11 experiment with changing the length of the planning horizon under different conditions. Shortening the planning horizon from 13 periods to 6 periods did not affect the system's performance under normal capacity conditions. (Compare Runs 1 and 8, and Runs 2 and 10). However, as one would have expected, the size of backorders began to increase significantly when the planning horizon is shorter under tight capacity. Run 11 deals with an aggregation of time periods in the planning horizon. The length of the planning horizon is still a full year but it is divided into only six time periods of uneven lengths. The first four periods have 4-week duration each, the fifth period covers 12 weeks (aggregation of three 4-week periods), and the sixth period covers 24 weeks (aggregation of six 4-week periods). Run 11 shows a performance quite similar to the base case. This result might indicate that this type of aggregation of the planning horizon could be useful in many situations, since it improves the forecasting accuracy in more distant time periods and reduces the associated computational time, without experiencing a decline in performance.

8.6 *Degree of Suboptimization*

Although our proposed hierarchical planning system provides optimum solutions to the subproblems that deal with individual decisions at each level, obviously it is not an overall optimum procedure. As we have pointed out, setup costs are ignored at the aggregate planning level, thus introducing suboptimization possibilities. To analyze how serious this suboptimization problem was, we developed a mixed integer programming (MIP) model at a detailed item level to identify the true optimal solution to our test problem. The MIP model was implemented by means of IBM's MPSX/MIP code, which is a general purpose branch and bound algorithm.

Due to the computational cost of solving MIP models, we limited our comparisons between the hierarchical planning system and the MIP model to those situations containing no forecasting errors. In those cases, we could solve the MIP model only once, and obtain the optimum yearly cost. (If forecast errors would have been introduced we would have had to solve the MIP model 13 times for each run, which was prohibitively expensive.)

We computed MIP solutions to three of our previous runs: the base case (Run 1'), the first high setup cost run (Run 4'), and the tight capacity run (Run 6'). The MIP results are given in Table 3. The existing limits on the node tables of the branch and bound code used did not allow us to determine the true optimum in the MIP runs. Therefore, the solutions reported in Table 3 might still be improved. Table 3 also provides the continuous lower bounds obtained at the time in which the computations

	1'	4'	6'
RUN / COST COMPONENTS	BASE CASE NO FORECAST ERROR	High Setup Cost CASE I P1: 5000,50 P2: 400,400,10000	TIGHT CAPACITY 1600 Reg. Hrs.
SETUP HOLDING OVERTIME	4,590 75,953 77,796	48,050 79,880 75,430	3,930 115,872 117,430
TOTAL COST (Best known solution)	158,339	203,360	237,232
LOWER BOUND	153,926	162,783	233,665

TABLE 3. SUMMARY OF COMPUTATIONAL RESULTS WITH MIXED INTEGER PROGRAMMING MODELS

were interrupted. For all practical purposes, we could consider the solutions corresponding to Runs 1' and 6' to be optimal. Possibly Run 4' could still be improved.

By comparing the total costs of the three runs for the Knapsack case we have:

	Knapsack Hierarchical System	Best known MIP solution
Base Case	158,981	158,339
High Setup Cost	220,535	203,360
Tight Capacity	236,733	237,232

We see that the hierarchical planning system was extremely efficient. It appears that only under abnormally high setup cost the system's perofrmance begins to depart significantly from the overall optimal solution.

In summary, the hierarchical systems seems to perform near optimum when setup costs are moderate. The base case cost, which reflected the operating data of the tire industry, is only .4 percent higher than the best known optimum solution obtained by the MIP formulation. However, the cost of each run of the hierarchical system was approximately $5, while the corresponding MIP run cost near $50. In addition, the MIP approach would be computationally impossible to carry out for larger problems. Moreover, the hierarchical system appears to offer coherent solutions under varying forecast errors, capacity availabilities, and planning horizon lengths.

Extremely high setup costs could affect the performance of the system. In practice, families with very high setups are candidates for continuous production (as opposed to batch production) if they have a high level of demand. In such a case, those families can be handled independently of the hierarchical system. In situations where there are few high setup families with low demand, special constraints can be imposed on the family disaggregation model to produce those families in large enough quantities. This can be accomplished by setting the lower bound of the family to its unconstrained economic order quantity. When all the families in the product structure have high setup costs and low demand levels, it might not be desirable to eliminate setup costs at the aggregate level. In that situation, we could eliminate the aggregate planning model for porduct types and allocate production quantities at the family level by using an approach similar to that proposed by Lasdon and Terjung [16]. We would then apply the item disaggregation model to allocate the family production quantities among items.

References

1. Bitran, G. R. and A. C. Hax; "On the Design of Hierarchical Production Planning Systems", Decision Sciences 8 (1), 28-55 (1977).

2. Bitran, G. R., and A. C. Hax; "On the Solution of Convex Knapsack Problems with Bounded Variables", Massachusetts Institute of Technology, Operations Research Center, Technical Report No. 129, April 1977.

3. Bradley, S. P., A. C. Hax, and T. L. Magnanti; Applied Mathematical Programming, Addison Wesley, 1977. Chapter 6 based on the technical paper "Integration of Strategic and Tactical Planning in the Aluminum Industry" by A. C. Hax, Operations Research Center, M.I.T. Working Paper 026-73, September 1973; Chapter 10 based on the paper by Robert J. Armstrong and Arnoldo C. Hax, "A Hierarchical Approach for a Naval Tender Job-Shop Design", Operations Research Center, M.I.T., Technical Report No. 101, August 1974.

4. Brown, R. G.; Decision Rules for Inventory Management, Holt, Rinehart, and Winston, 1967.

5. Buffa, E. S., and W. H. Taubert; Production-Inventory Systems: Planning and Control, Richard D. Irwin, 1972.

6. Candea, D.; "Issues of Hierarchical Planning in Multi-Stage Production Systems", Massachusetts Institute of Technology, Operations Research Center, Technical Report No. 134, July 1977.

7. Dzielinski, B. P., C. T. Baker, and A. S. Manne; "Simulation Tests of Lot Size Programming", Management Science 9 (2) 229-258 (1973).

8. Dzielinski, B. P., and R. E. Gomory; "Optimal Programming of Lot Sizes, Inventory and Labor Allocations", Management Science 2 (9) 874-890 (1965).

9. Gabbay, H.; "A Hierarchical Approach to Production Planning", Massachusetts Institute of Technology, Operations Research Center, Technical Report No. 120, December 1975.

10. Golovin, J. J.; "Hierarchical Integration of Planning and Control", Massachusetts Institute of Technology, Operations Research Center, Technical Report No. 116, September 1975.

11. Hax, A. C.; "The Design of Large Scale Logistics Systems: A Survey and an Approach", in Modern Trends in Logistics Research (ed.: W. Marlow), M.I.T. Press, 1976.

12. Hax, A. C.; "Aggregate Production Planning", in Handbook of Operations Research (ed.: J. Moder and S. E. Elmaghraby), Van Nostrand Reinhold, to appear.

13. Hax, A. C., and J. J. Golovin; "Computer Based Operations Management System (COMS)", in Studies in Operations Management (ed.: A. C. Hax), North Holland-American Elsevier, 1978.

14. Hax, A. C., and H. C. Meal; "Hierarchical Integration of Production Planning and Scheduling", in Studies in Management Sciences, Vol. I, Logistics (ed.: M. A. Geisler), North Holland-American Elsevier, 1975.

15. Holt, C. C., F. Modigliani, J. F. Muth, and H. A. Simon; Planning Production, Inventories and Work Force, Prentice Hall, 1960.

16. Lasdon, L. S., and R. C. Terjung; "An Efficient Algorithm for Multi-Item Scheduling", Operations Research 19 (4) 946-969 (1971).

17. Manne, A. S.; "Programming to Economic Lot Sizes", Management Science 4 (2) 115-135 (1958).

18. Silver, E. A.; "A Tutorial on Production Smoothing and Work Force Balancing", Operations Research 15 (6) 985-1010 (1967).

19. Shwimer, J.; "Interactions Between Aggregate and Detailed Scheduling in a Job Shop", Massachusetts Institute of Technology, Sloan School of Management, Unpublished Ph.D. Thesis, June 1972.

20. Winters, P. R.; "Constrained Inventory Rules for Production Smoothing", Management Science 8 (4) 470-481 (1962).

21. Zimmermann, H. J., and M. G. Sovereign; Quantitative Models for Production Management, Prentice Hall, 1974.