A GENERAL FRAMEWORK FOR MODELING PRODUCTION*

STEVEN T. HACKMAN AND ROBERT C. LEACHMAN

School of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, Georgia 30332

Engineering Systems Research Center, University of California, Berkeley, California 94720

We introduce a general framework that guides the management scientist’s formulation of deterministic models of production processes. Using the framework, we reformulate the constraints of familiar linear programming-based planning models to specifically treat components of production lead time, thereby realizing a more accurate representation of the production process. In addition, the reformulation accommodates noninteger values for lead times as well as unequal-length planning periods. Manufacturing Resources Planning (MRP) and the Critical Path Method (CPM) are recast in terms of the framework, revealing opportunities for model generalization and extension, and their relationship to linear programming models.

(PRODUCTION PLANNING AND SCHEDULING; MODEL FORMULATION; CONTINUOUS TIME ANALYSIS)

1. Introduction

A number of researchers in production planning and scheduling have proposed frameworks to help focus, position and evaluate future research. There is little consensus, however, as to what such a framework should consist of or what needs it should address. Abraham et al. (1985) suggest a hierarchical framework of decision levels corresponding to a presumed hierarchy of decision-making. Dempster et al. (1981) propose a framework for the analytical evaluation of hierarchical planning systems. Bensoussan et al. (1985) offer a “unified mathematical treatment of production planning and production smoothing problems, in the framework of optimal control theory.” Akinc and Roodman (1986) propose a new framework for modeling aggregate production planning problems in which emphasis is placed on flexibility in specification of the decision options and the relevant cost structure. Geoffrion (1987) proposes a “structured modeling” framework to facilitate the data-base development for application of pre-specified operations research models. The foregoing frameworks consider the representation of the appropriate decision alternatives, the decision hierarchy, the cost structure, and the effort required in data-base development, which clearly are important concerns for planning and scheduling.

There is another, equally fundamental concern that attracts our research efforts and is the subject of this paper. Embedded in each planning and scheduling model is a model of production—a mathematical representation of the set of technologically feasible operations or actions within the production process. Only a very narrow range of models of production has been considered in the production literature, as well as in the above-cited frameworks. Clearly, an accurate representation of the production process is critical to the optimality and even feasibility of generated plans and schedules. In this paper, we introduce a general framework to (1) assess the accuracy or validity of the representation of a particular production process by a given production model, and (2) guide the development of more accurate production models when existing models provide inadequate representations of the production process.

The framework incorporates familiar modeling principles such as flow conservation and activity analysis, but it expresses these principles in terms of new abstract model

* Accepted by L. Joseph Thomas; received November 4, 1986. This paper has been with the authors 5 months for 2 revisions.
elements. When existing models of production are expressed in terms of these elements, extensions and generalizations become apparent. Moreover, the mathematical language and structure of the framework provides a foundation to compare and contrast seemingly unrelated models arising in such diverse environments as discrete-parts manufacturing, continuous-flow production, and project management.

The framework is introduced in §2. In the sequel, we redefine in terms of the framework three of the most familiar production planning models: standard linear programming formulations, Manufacturing Resources Planning (MRP), and Critical Path Methods (CPM). Each model highlights different abstract model elements of the framework. In §3, we apply the framework to formulate a more accurate linear programming model of production. Components of "production lead time" are categorized in terms of the model elements of the framework. Our formulation generalizes familiar formulations that do not break down lead times into components, and it is further generalized to accommodate noninteger values for the lead time components as well as unequal-length planning periods. In §4, we show that MRP techniques are based on an incomplete model of production. In §5, we show that a strict precedence relationship between CPM activities is a form of inventory balance, and we discuss how CPM can be usefully extended by incorporating activity analysis of l.p. models. We conclude the paper with a brief discussion of how the framework can guide the management scientist in formulating models of production.

2. A General Framework

The framework is a meta-model of production. It delineates precisely what is and what is not a valid production model in much the same way the "meta-model": \( \min \{ c \cdot x: A \cdot x = b, x \geq 0 \} \) delineates what is and what is not a linear program. The framework identifies the abstract model elements, or building blocks, and the relationships between the elements, that are required in a mathematical representation of a production system. Shephard (1970) developed the first axiomatic steady-state model of production. The model elements in his framework are production correspondences that abstractly model input-output possibilities of the technology. The first general activity network model was developed in Shephard et al. (1977), which introduced transfers of intermediate products between activities and activity dynamic production correspondences. Hackman (1984) extended this framework, replacing activity correspondences with activity dynamic production functions. Our presentation in this section is based in part on Hackman (1984).

2.1. The Model Elements

The framework models a production system as a directed network whose nodes represent primitive production activities, i.e., activities whose internal organization is not further modeled. (The activity network is merely an abstract representation of work flow and need not represent any physical arrangement of facilities.) Directed arcs indicate possible transfers of intermediate products. Intermediate products are outputs of one or more activities which serve as inputs to other activities. Production at each activity requires intermediate products transferred from other activities and/or system exogenous inputs. System exogenous inputs are of two types: nonstorable services and storable materials. Services include labor trades, machines and facilities, while materials include purchased parts, raw materials, fuels, etc. An exogenous material may be the same as an intermediate product of the system.

Each flow of a service or material is modeled as a bounded function of time, where time is modeled by the nonnegative part of the real line. Time 0 is a reference point defining the point in time after which flows would be determined by planning calculations using the model.

There are two fundamental types of flows. In the first and more common type, called a rate-based flow, \( x(\tau) \) represents the rate—quantity per unit time—of flow at time \( \tau \).
In the second type, called an event-based flow, $x(\tau)$ is the numerical value of an event at time $\tau$. For example, $x(\tau)$ might indicate the quantity transferred of an intermediate product at time $\tau$. Unless specifically identified as event-based, all flows in this paper are rate-based.

For each model, an index set $\Lambda = \{ t_k \}_{k=1}^\infty$ of epochs is defined. The $t_k$'s are ordered so that $0 = t_0 < t_1 < \cdots$. Each $I_k = (t_{k-1}, t_k]$ is a period, the points in $\Lambda$ are called time-grid points, and $\Lambda$ is called the time grid. Both lower and upper bounds on the lengths of the periods are assumed. For event-based flows included in the model, the set $\Lambda$ indicates the possible times of events.

It will be convenient when representing inventory balance equations to work with cumulative flows. For a flow $x$, the corresponding cumulative flow is denoted by the upper case letter $X$, and is defined by $X(t) = \int_0^t x(\tau) \, d\tau$, $t \geq 0$. Conversely, the unintegrated flow corresponding to the cumulative flow $X$ is denoted by the lower case letter $x$.

Unless otherwise stated, if a component of a vector of flows is not explicitly defined, then it is assumed to be the zero function.

We index the different inputs and outputs of the system from 1 to $K$ with the first $M$ indices denoting products or materials and the last $K-M$ indices denoting nonstorable services such as labor and machine time. Let $v_{ij} = (v_{ij}^1, v_{ij}^2, \ldots, v_{ij}^M)$ denote the vector of flows representing transfers of products from activity $i$, $i = 0, 1, \ldots, N$, sent to activity $j$, $j = 1, 2, \ldots, N + 1$. These transfers may not be immediately received at activity $j$.

To distinguish transfers sent from those received, the framework associates with $v_{ij}$ the vector $\delta_{ij} = (\delta_{ij}^1, \ldots, \delta_{ij}^M)$ of transfers from activity $i$ received at activity $j$.

The framework further distinguishes transfers received at an activity from inputs actually applied in production. Let $w_i = (w_i^{M+1}, w_i^{M+2}, \ldots, w_i^K)$ denote the allocation flows of system exogenous services to activity $i$ and let $y_i = (y_i^1, y_i^2, \ldots, y_i^K)$ denote the flows of inputs actually applied in production at activity $i$.

In each model, a joint domain $D$ for the allocations, applications and transfers ($\{ w_i \}$, $\{ y_i \}$ and $\{ v_{ij} \}$) reflects the domain of technologically feasible flows. The domain may involve complicated constraints linking applications at different activities. For example, suppose activities $A$ and $B$ represent production of products $A$ and $B$, respectively. Suppose each activity uses the service of the same machine which can only process one type of product at a time. Then the domain for applications should be defined to ensure that applications of the machine service at activities $A$ and $B$ are not simultaneous.

For each activity $i$, a dynamic production function $f_i$ maps input flows $y_i = (y_i^1, \ldots, y_i^K)$ applied at activity $i$ into realized output flows $f_i(y_i) = (f_i(y_i^1), \ldots, f_i^K(y_i))$. The domain of the production function $f_i$ (as derived from the joint domain $D$) is denoted by $D_i$. The dynamic production function differs from the traditional steady-state production function in two respects: First, it maps applications of inputs into realized outputs rather than allocations of inputs into possible outputs. Second, both $y_i$ and $f_i(y_i)$ are vector-valued functions of time rather than steady-state rates.

In the case of processes with time lags, activity output flows and transfers-received flows from time 0 until some time after time 0 could be functions of applications and transfers-sent which occur before time 0. We assume that all flows which are consequences of applications or transfers-sent at or before time 0 are fixed and prespecified in planning calculations.

For ease of reference, the notation for the elements of the framework is summarized in Table 1.

### 2.2. The Model of Production

Given an identification of the model elements and their domains, the framework specifies a deterministic model of production in terms of flow conservation constraints.
We refer to equations (2.1) as the **fundamental inventory balance equations**. The physical inventory at time $t$ at activity $i$ of material $m$ must be nonnegative. This inventory equals the cumulative supply of material $m$ at activity $i$—including initial stock, production and received transfers—minus the cumulative amount of product or material

---

**TABLE 2**  
Equations of the General Framework

---

**Conservation of Materials**

(a) Balance of Local Inventories. For $i = 1, \ldots, N$, $m = 1, \ldots, M$, and for all $t$,

$$I_i^m(t) = I_i^m(0) + F_i^m(Y_i(t)) + \sum_{j=0}^{N} \tilde{V}_{ij}^m(t) - \sum_{j=1}^{N+1} V_{ij}^m(t) - Y_i^m(t) \geq 0. \quad (2.1)$$

(b) Balance of System Raw Material Supplies. For $m = 1, \ldots, M$, and for all $t$,

$$\sum_{i=1}^{N} V_{0i}^m(t) \leq C^m(t). \quad (2.2)$$

(c) Demand Satisfaction. For $m = 1, \ldots, M$, and for all $t$,

$$I_{M+1}^m(0) + \sum_{i=1}^{N} \tilde{V}_{iM+1}^m(t) - U^m(t) \geq 0. \quad (2.3)$$

---

**Conservation of Services**

(a) Application Capacities. For $k = M + 1, \ldots, K$, $i = 1, \ldots, N$, and for all $t$,

$$y_i^k(t) \leq w_i^k(t). \quad (2.4)$$

(b) Allocation of System Capacities. For $k = M + 1, \ldots, K$ and for all $t$,

$$\sum_{i=1}^{N} w_i^k(t) \leq c^k(t). \quad (2.5)$$
m transferred from and used by activity i. The framework models inventories of products
at each activity allowing for completed products awaiting transfer and transferred products
awaiting application. Hence, a particular product may have multiple inventory locations.
Equations (2.2) express the conservation of system raw material supplies, while equations
(2.3) express the satisfaction of exogenous demands. The possibility for backlogging is
not included in (2.3), although the usual modifications could be made to accommodate
this case.

The conservation of nonstorable services must be expressed instantaneously, as in
equations (2.4) and (2.5). Note that the allocations of services are distinguished from
their applications in order to accommodate domain constraints on allocations that may
exist in a particular system.

Rather than repeatedly delineate assumptions common to various models, it is con-
venient to define the following model categories. In an acyclic production model, the
activity network is acyclic. In a product-generated production model, each activity pro-
duces a single product not produced by any other activity. In a discrete-time production
model, each rate-based flow is a step function, constant in each time period. In addition,
a finite horizon is assumed. In a normal production model, (1) activities do not receive
products they can not apply, and (2) activities do not transfer products they can not
produce.

A deterministic model of a specific production process is categorized and explicitly
described by identifying and defining the abstract model elements: the activity network,
the exogenous inputs, the intermediate and final products, the activity dynamic production
functions, and the domains for the allocations, applications and transfers. The formulation
of a production planning model includes constraints (2.1)–(2.5), plus domain constraints
and an appropriate objective function. While the activity network, inputs and products
are well-defined in most presentations of the models we consider in follow-on sections,
the production functions and the domains typically are treated only implicitly. We will
thus focus our attention on these model elements.

3. Linear Programming Models

3.1. Leontief Production Functions

Most of the familiar linear programming models of production employ a very simple
class of activity production functions. In particular, if the applications of inputs are
positive, then they must be proportional and therefore may be indexed in terms of one
profile. The outputs produced are also assumed to be proportional and indexed by the
same profile which indexes the inputs. We refer to the profile as the intensity of the
activity.

Casting this in the language of the framework, we say the domain $D_i$ of a production
function $f_i$ is Leontief if there are nonnegative constants $a_i^m, m = 1, 2, \ldots, K$ such that

$$D_i \subset \{ y_i | y_i^m(t) = a_i^m z_i(t), m = 1, 2, \ldots, K \text{ for some intensity curve } z_i \}.$$ 

As each curve $y_i^m$ in a Leontief domain $D_i$, may be expressed in terms of its defining
intensity curve $z_i$, we write $f_i(z_i)$ instead of $f_i(y_i)$. We say that a production function $f_i$
with Leontief domain is itself Leontief if there are nonnegative constants $c_i^m, m = 1, 2, 
\ldots, M$ such that $f_i^m(z_i) = c_i^m z_i, m = 1, 2, \ldots, M$. (The production functions are so
named due to their similarity to the steady-state functions used in Leontief 1951.) The
coefficient $a_i^m$ represents the rate input $m$ is applied to activity $i$ per unit intensity, and
the coefficient $c_i^m$ represents the rate product $m$ is produced by activity $i$ per unit intensity.

If an activity’s production process is modeled by a Leontief production function, then
the model assumes production is instantaneous and time-invariant. That is, an output
same point in time; moreover, this input/output transformation is the same at all points in time. If output at time \( t \) is actually a function of input applications over some period of time, then the Leontief function is inappropriate.

### 3.2. Basic Linear Programming Models Without Time Lags

The following assumptions about the production system lead to a linear programming model of production:

1. The system is represented by a product-generated (normal) activity network.

2. Each activity dynamic production function is Leontief and the only constraints on its domain \( D_i \) are those implied by the Leontief assumption. (This characterizes the Dynamic Linear Activity Analysis Model (DLAM) presented in Shephard et al. 1977.) As a result of assumption (1), activity intensity will be measured in units of output. Thus, \( f_i^i(z_i) = z_i, i = 1, 2, \ldots, m \).

3. There are no domain constraints on allocations or transfers (other than nonnegativity).

4. A product is held in inventory only at the activity that produces it.

5. Intermediate product transfers are "instantaneous", i.e., transfers sent equal transfers received. Combining this assumption with assumptions (3) and (4), as intermediate products are transferred, they are applied. In the language of the framework, \( v_{ij} = \beta_{ij}^j = \beta_{ij}^j = a_{ij}^j z_j \).

6. Transfers of final product are instantaneous. In view of assumptions (3), (4) and (5), transfers of final product are equated to final output requirements, i.e., \( v_{iN+1} = u_i^i \).

7. The only exogenous inputs are the nonstorables, i.e., \( v_{0i} = 0, m = 1, \ldots, M_i, i = 1, \ldots, N \).

8. Since domain constraints on allocations are assumed to be trivial, allocations are equated to applications, i.e., \( w_i^k = y_i^k, k = M + 1, \ldots, K \).

From assumptions (1)–(8), we obtain the following model of production: For each intermediate product \( i \), inventory balance equation (2.1) reduces to

\[
I_i^i(t) = I_i^i(0) + Z_i(t) - \sum_{j=1}^{M} a_{ij}^j Z_j(t) - U_i^i(t) \geq 0, \quad \text{all } t \geq 0. \tag{3.1}
\]

For each nonstorables service \( k \), the allocation constraint (2.5) becomes

\[
\sum_{i=1}^{N} a_{ik}^i z_i(t) \leq c_k(t), \quad \text{all } t \geq 0. \tag{3.2}
\]

To develop a linear programming model, we invoke the additional assumptions of a discrete-time production model with unit-length time periods \((0,1], (1,2], \ldots\). (As is customary, we refer to the interval \((t - 1, t]\) as "period \( t\).") Note that on each unit interval, all flows in and out of inventory of product \( i \) are at constant rates; hence the rate of change of the inventory level during each interval is constant. If we enforce nonnegativity of the inventory of product \( i \) at the end points of an interval, it follows that the inventory must be nonnegative during the entire interval. Hence, inventory balance can be guaranteed for all \( t \geq 0 \) simply by enforcing (3.1) at the time grid points.

Equations (3.1) and (3.2) now may be rewritten as, respectively,

\[
I_i^i(t) = I_i^i(0) + \sum_{r=1}^{t} z_{ir} - \sum_{r=1}^{t} \left\{ \sum_{j=1}^{M} a_{ij}^j z_j(r) + u_i^i \right\} \geq 0, \quad i = 1, 2, \ldots, M_i, \quad t = 1, 2, \ldots, \tag{3.3}
\]

\[
\sum_{i=1}^{M} a_{ik}^i z_i(t) \leq c_k(t), \quad k = M + 1, \ldots, K, \quad t = 1, 2, \ldots, \tag{3.4}
\]
where $z_{it}$, $u_{it}^l$, and $c_{kt}$ denote constant rates of flows during the respective time periods. In more familiar form, (3.3) and (3.4) are expressed as

$$I^i(t) - I^i(t - 1) = x_{it} - \sum_{j \neq i} a_{ij}^t x_{jt} - u_{it}^l, \quad i = 1, 2, \ldots, M, \quad t = 1, 2, \ldots, (3.5)$$

$$I^i(t) \geq 0, \quad i = 1, 2, \ldots, M, \quad t = 1, 2, \ldots, (3.6)$$

$$\sum_{i=1}^M a_{ik}^t x_{it} = c_{kt}, \quad k = M + 1, \ldots, K, \quad t = 1, 2, \ldots, (3.7)$$

where $x_{it}$ denotes the amount produced of product $i$ in period $t$, $I^i(t)$ denotes the inventory of product $i$ at time $t$, $u_{it}^l$ denotes the final demand for product $i$ in period $t$, $a_{ij}^t$ denotes the amount of service $k$ required per unit of product $i$, and $a_{ij}^t$ denotes the amount of product $i$ input per unit output of product $j$. We have just shown that physical assumptions (1)–(8) lead to the constraints (3.5)–(3.7) in the traditional linear programming multi-period multi-stage aggregate production planning model (see, for example, Hax and Candea 1985 or Johnson and Montgomery 1974).

The assumptions under which (3.5)–(3.7) comprise a valid model of production are now clear. If the transformations of activity input to output are not proportional, not time-invariant, or not instantaneous, or if transfers become significant because there are shipment lags or because transfers are not applied immediately (e.g., inspections), then (3.5)–(3.7) is no longer a valid model of production.

### 3.3. Linear Programming Models with Time Lags

One particularly unrealistic aspect of the assumptions leading to the previous linear programming model is the requirement that processes occur instantaneously. For example, assumption (2) insists that output from an activity depends only on input to that activity at the same instant in time. This assumption precludes operations with significant post-processing lags (e.g., time for steel to cool), and also operations with significant processing times (e.g., metal-cutting). Likewise, assumption (5) precludes situations with significant transfer times.

Considerable research has been devoted to developing practical linear programming models without these instantaneous assumptions. In all of this work, all significant time lags are lumped under the single heading “lead time”. For example, Billington et al. (1986) describe lead time as a “nonproduction lag, such as the time for paint to dry, hot metal to cool, or a batch to be physically moved between production areas.” They develop a linear programming model with capacity constraints of the form

$$I^i(t) = I^i(0) + \sum_{t=1}^i x_{it}(t-L_i) - \sum_{t=1}^M \sum_{j=1}^t a_{ij}^t x_{jt} - \sum_{t=1}^i u_{it}^l \geq 0, \quad i = 1, 2, \ldots, (3.8)$$

Here, $L_i$, the “unavoidable lead time” for product $i$, is defined so that output of product $i$ started in period $t$ may first become available in period $t + L_i$. Inherent in the definition of $L_i$ is the assumption that each unavoidable lead time is independent of the production rate and can be expressed as an integral number of time periods.

We now demonstrate that this formulation can be generalized to more correctly model the stated phenomena. To do so, we define two different types of lags contributing to lead time and formulate a more general linear programming model assuming these time lags are integral.

**Output Lag.** In some production processes, a product cannot be transferred or released to inventory for a period of time after it is “produced”. For example, after painting, a fixed amount of time may be required for the paint to dry; or after lumber is sawed, a fixed amount of time may be required to inspect and grade the output. For such processes,
assumption (2) must be modified as follows: Input applications at activity \( i \) are still proportional, but the output curve is shifted by a constant amount \( LA_i \geq 0 \). (The symbol “LA” denotes “lag after” application.) In the language of the framework,

\[
f_i^i(z_i)(\tau) = \begin{cases} 
  z_i(\tau - LA_i) & \text{if} \quad \tau > LA_i \\
  \text{prespecified} & \text{for} \quad 0 < \tau \leq LA_i.
\end{cases}
\]

Note that applications of inputs are still instantaneous, but consequent output emerges after a time lag \( LA_i \). Note also that, since applications are prespecified before time 0, output of product \( i \) must be prespecified for \( \tau \in (0, LA_i] \). Substituting

\[
F_i^i(z_i)(t) = \int_0^t f_i^i(z_i)(\tau) d\tau = \int_{-LA_i}^{t-LA_i} z_i(\tau) d\tau,
\]

(2.1) becomes

\[
I_i^i(t) = I_i^i(0) + \int_{-LA_i}^{t-LA_i} z_i(\tau) d\tau - \sum_{j=1}^{M} a_{ij}^i z_j(t) - U^i(t) \geq 0, \quad \text{all } t \geq 0. \quad (3.9)
\]

Transfer/Input Lag. There also may be a period of time from when a product is withdrawn from inventory until it is applied as an input at a follow-on activity. For example, there may be time required to transfer parts between activities, time to inspect inputs before application (where the inspection does not utilize scarce resources), etc. Suppose the delay from the time of withdrawal of product \( i \) from inventory until application as input at activity \( j \) is exactly \( LB_{ij} \) time units. (The symbol “LB” denotes “lag before” application.) Retaining our other assumptions, assumption (5) must be modified as follows:

\[
y_{ij}(\tau) = a_{ij}^j z_j(\tau) = \begin{cases} 
  v_{ij}^j(\tau - LB_{ij}) & \text{if} \quad \tau > LB_{ij}, \\
  \text{prespecified} & \text{for} \quad 0 < \tau \leq LB_{ij}.
\end{cases}
\]

Note that \( z_j(\tau) \) must be prespecified for \( \tau \in (0, LB_{ij}] \) because it corresponds to transfers sent from activity \( i \) at or before time 0. Retaining the output lags and noting that \( V_{ij}^j(t) = \int_{LB_{ij}}^{t+LB_{ij}} v_{ij}^j(\tau) d\tau \), (2.1) now becomes

\[
I_i^j(t) = I_i^j(0) + \int_{-LA_i}^{t-LA_i} z_i(\tau) d\tau - \sum_{j=1}^{M} \int_{LB_{ij}}^{t+LB_{ij}} a_{ij}^j z_j(\tau) d\tau - U^i(t) \geq 0,
\]

all \( t \geq 0. \quad (3.10)\)

The intensity of each activity \( j \) now must be prespecified for \( -LA_j < \tau \leq \max_i \{ LB_{ij} \} \).

To develop a linear programming model, we once again invoke the assumptions of a discrete-time linear programming model with unit-length time periods. If \( LA_i \) and \( LB_{ij} \) are integers, then all flows in and out of inventory of product \( i \) are constant during each interval. Recalling our earlier discussion, inventory balance will be guaranteed for all \( t \geq 0 \) if it is enforced at the time grid points. Retaining the notation \( z_{ij} \) for the constant intensity of activity \( i \) and \( u^i_t \) for the constant rate of demand for product \( i \) during \( (\tau - 1, \tau) \), the discrete-time version of (3.10) is

\[
I_i^j(t) = I_i^j(0) + \sum_{\tau=1}^{t} z_{ij(\tau-LA_i)} - \sum_{j=1}^{M} \sum_{\tau=1}^{t} a_{ij}^j z_{ij(\tau+LB_{ij})} - \sum_{\tau=1}^{t} u^i_t \geq 0, \quad t = 1, 2, \ldots. \quad (3.11)
\]

Figure 1 summarizes our lead time model.

Under the special assumptions we have made in this model, separate identification of transfer and input (e.g., inspection) lags is unnecessary. These assumptions are as follows: (1) the only domain constraints on transfers are nonnegativity, and (2) uncommitted
inventory of a product is held only where produced. Under alternatives to (1), such as discrete domains for transfers or upper bounds on transfers, it is meaningful to relax (2). In such a case, uncommitted inventory of an intermediate product would be held both where produced and where used, necessitating separate transfer and input lags as well as multiple inventory balance equations to ensure conservation of the product.

Comparison with Familiar Formulations. We now analyze familiar formulations incorporating time lags in terms of our framework, and the physical assumptions of our simple lead time model. We first consider the mathematical programming model of Billington et al. (1983, 1986), whose inventory balance constraints are given in (3.7), and whose capacity constraints are of the form (3.7). Hax and Candea (1985) also use (3.8) and (3.7).

Under our physical assumptions, the only (uncommitted) inventory of product \( i \) in the production system is the inventory of completed output of activity \( i \) awaiting transfer for intermediate or final uses. In this case, (2.1) for \( m = i \) takes the form

\[
I_j^i(t) = I_j^i(0) + F_j^i(Y_j(t)) - \sum_{j=1}^{N+1} V_{ij}^i(t) \geq 0. \tag{3.12}
\]

Compare (3.8) to (3.12). We conclude that \( I_j^i(t) = I_j^i(t), v_{ij}^i(t) = a_{ij}^i x_j(t), \) and \( f_j(y_j(t)) = x_j(t - L_j) \). From the capacity constraints (3.7), we observe that \( y_j^i(t) = a_{ij}^i x_j(t) \). Hence, this model assumes that the withdrawal from inventory of intermediate product inputs and application of these inputs is simultaneous, i.e., \( LB_{ij} = 0 \). If indeed \( LB_{ij} = 0 \), then (3.8) coincides with our derived model (3.11) by making the identification \( L_i = LA_i \). If, on the other hand, lags of the form \( LB_{ij} \) exist, there exists an offset between withdrawals from inventory \( v_{ij}^i(t) \) and resource applications \( a_{ij}^i x_j(t) \), i.e., \( v_{ij}^i(t) \) does not equal \( a_{ij}^i x_j(t) \). (See Figure 1.)

There are two approaches to correct (3.8): First, the variable \( x_{it} \) could be interpreted as an index of resource applications. In this case, \( y_j^i(t) = a_{ij}^i x_j(t), v_{ij}^i(t) = a_{ij}^i x_j(t + LB_{ij}) \) and \( f_j(y_j(t)) = x_j(t - LA_i) \), whereby we are led back to our model (3.11). Second, the variable \( x_{it} \) could be interpreted as a “starts” index, i.e., an index of the withdrawals from inventory of all of the intermediate product inputs needed to make a quantity \( x_{it} \) of product \( i \) available as output \( L_i \) time units later. Since the total time from the withdrawal from inventory of intermediate product input \( j \) until output of product \( i \) is \( L_i \)—independent of \( j \)—this interpretation must assume that \( LB_{ij} = LB_i \). With this assumption, \( v_{ij}^i(t) = a_{ij}^i x_j(t), y_j^i(t) = a_{ij}^i x_j(t - LB_i) \) and \( f_j(y_j(t)) = x_j(t - L_i) \), where \( L_i = LB_i + LA_i \). The correction to the (3.8)–(3.7) formulation in this case is to modify (3.7) to \( \sum_{j=1}^{M} a_{ij}^i x_j(t - LB_i) \leq c_{it} \).

Krajewski and Ritzman (1977) propose a “general model” which utilizes balance equations of the form

\[
I_j^i(t) - I_j^i(t - 1) = \sum_{j=1}^{M} a_{ij}^i z_{j(t + L_j)} - u_j. \tag{3.13}
\]
Comparing to (3.12), this model assumes that \( f_i(y_i)(t) = x_i(t) \) and \( v_j(t) = a_j^x_i(t + L_j) \). In this case, the variables \( x_i \) represent "out" instead of "start". As in the previous formulation, the assumption is implicitly made that transfer/input lags \( LB_{ij} = LB_i \). If these input lags actually vary by input type, (3.13) should be modified by replacing \( a_j^x_i(x_{i(t+L_j)}) \) by \( a_j^x_i(x_{i(t+L_j)}) \) where \( L_{ij} = LB_i + LA_j \).

Under our physical assumptions, the appropriate capacity constraints to use are of the form \( \sum_{t=1}^{M_i} a_j^x_i(x_{i(t+LA_i)}) \leq c_{ik} \).

To summarize, familiar formulations utilize one lead time \( L_i \) to measure the time between when activity output enters inventory and the time when activity inputs are withdrawn. Except for Krawiecki and Ritzman (1977), familiar formulations also assume resource loading occurs at the same point in time as when inputs leave inventory. Our formulation models the case where resource loading and withdrawals of inputs from inventory may occur at different points in time, as well as the case where different inputs of activity production may need to be withdrawn from inventory at different points in time. Distinguishing these lead times in the formulation does not require the addition of any rows or any columns to the linear program.

A Linear Programming Model with Noninteger Lead Times. The assumption that lead times are integer is quite restrictive, although it is assumed by all the familiar formulations. We now derive a discrete-time model when lead times are not necessarily integer. We retain the assumption that activity intensities, service capacities, and final demands are constant rates during unit-length time periods.

To evaluate inventory balance of product \( i \) at time \( t \), one must express the integral functions (3.10) in terms of the \( z_i^+ \) and \( z_i^- \) of \( z_i \) in a manner which accounts for the noninteger limits of integration. For a real number \( x \), we let \( x^+ \) denote the smallest integer greater than or equal to \( x \), and let \( x^- \) denote the largest integer less than or equal to \( x \). The first integral in (3.10) is expressed, for all \( t \geq 0 \), as

\[
\int_{-LA_i}^{t-LA_i} z_i(\tau) d\tau = \begin{cases} 
(t)z_i(-LA_i)^+ & \text{if } t - LA_i < (-LA_i)^+ \\
(-LA_i)^+ + LA_i z_i(-LA_i)^+ + \sum_{(-LA_i)^+ < \tau \leq (t-LA_i)^-} z_{i\tau} \\
+ [(t - LA_i) - (t - LA_i)^-]z_i(t-LA_i)^+ & \text{otherwise.}
\end{cases}
\] (3.14)

In (3.14), the coefficients of the first and last terms of the lower expression express the fractions of intensity in the first and last time periods which are included within the limits of integration; the middle term simply sums up intensities of all time periods (if any) in between. The upper expression accounts for the degenerate case when both limits of integration lie in the same time period. The second integral in (3.10) is analogously expressed, for all \( t \geq 0 \), as

\[
\int_{LB_{ij}}^{t+LB_{ij}} z_j(\tau) d\tau = \begin{cases} 
(t)z_j(LB_{ij})^+ & \text{if } t + LB_{ij} < (LB_{ij})^+ \\
(LB_{ij})^+ - LB_{ij} z_j(LB_{ij})^+ + \sum_{(LB_{ij})^+ < \tau \leq (t+LB_{ij})^-} z_{j\tau} \\
+ [(t + LB_{ij}) - (t + LB_{ij})^-]z_j(t+LB_{ij})^+ & \text{otherwise.}
\end{cases}
\] (3.15)

Finally, \( U_i(t) \) in (3.10) is expressed, for all \( t \geq 0 \), as

\[
U_i(t) = \sum_{t=1}^{t^-} u_i^- + (t - t^-)u_i^+.
\] (3.16)

Substituting the identities (3.14)–(3.16) into (3.10) provides the desired expression of \( I_i(t) \) in terms of discrete-time variables for any \( t \geq 0 \). Identifying the flows which are functions of applications or transfers-sent occurring before time 0, we note that for each

\footnote{A similar improvement may be made to MRP calculations. See §4.}
activity $i$ and for $\tau = (LA_i)^+, (LA_i)^+ + 1, \ldots, \max_j \{LB^j_i\}$, the intensities $\{z_{it}\}$ must be prespecified.

If the time lags are not integer, ensuring inventory balance (3.10) only at the integer time grid points is not enough to ensure feasibility. Consider the example shown in Figure 2. Here, $U^t = 0$, $LA_i = 1.7$, $LB_i = 1.7$, and $a_j^t = 1$. A simple check shows that (3.10) is satisfied for $t = 1, 2, 3$. However, at $t = 1.7$ cumulative output of $i$ is zero, yet two units are required by $j$ at this time—a clear infeasibility. This phenomenon can happen since the flows $f'_j(z_i)$, $v_{ij}$, and $\hat{v}_{ij}$ are no longer constant on the given time intervals. These flows are still step functions, but the time points where rates of flow may change fall in between the integer time points.

In order to apply a formulation with integral lead times, one might try to overcome this difficulty by either rounding lead times to integer amounts or by subdividing the natural time period so that lead times become integral. The first approach either underestimates lead time, potentially leading to infeasibilities, or overestimates lead times, potentially leading to excessive work-in-process. The second approach greatly increases problem size.

To ensure inventory balance (3.10) for all time, we claim it is necessary and sufficient to check (3.10) at all time points when the rates of final demand flows, activity output flows or intermediate product transfer flows may change. For product $i$ let $T_i$ denote this collection of time points. A point in time $t \geq 0$ is an element of $T_i$ if, and only if, $t - LA_i$ or $t + LB_i$ or $t$ itself is an integer. In the example,

$$T = \{0, 0.3, 0.7, 1.0, 1.3, 1.7, 2.0, 2.3, 2.7, 3.0, \ldots\}.$$

For proof of this claim, note that on the intervals defined by successive points of $T_i$, all flows in and out of inventory of product $i$ are at constant rates; hence, following our
earlier discussion, the inventory level at any time point within such an interval will be nonnegative if the inventory level is enforced to be nonnegative at the end points.

This formulation is not only accurate but it requires significantly fewer constraints than would be included in a model employing a time grid fine enough to make all lead times integer. In the example, our formulation requires 3 balance equations per period for product $i$. To realize integer lead times a grid with periods of length 0.10 is required for product $i$, making for 7 extra balance equations within each period. Worse, if one time grid is used to enforce inventory balance for all products, an even finer grid might be required. The width of the required grid would be the least common divisor of the points in $\bigcup_{i=1}^{M} T_i$. On the other hand, the precision of the lead times does not affect the number of balance equations required in our proposed approach.

A Linear Programming Model with Unequal-Length Time Periods. Another restrictive assumption of familiar formulations is that all time periods have equal length. However, unequal-length periods are desirable, for example, when natural time periods such as weeks include varying numbers of working days because of vacations or holidays. We now modify our previous formulation to handle the case when the lengths of the periods with constant activity intensities are not necessarily equal. Let $S$ denote the set of epochs marking the end points of these periods. To ensure feasibility, it is necessary and sufficient to evaluate (3.10) at those time points $t$ such that $t - LA_i$ or $t - LB_i$ or $t$ is an element of $S$ (thus modifying the definition of $T_i$). The expressions of the integrals in (3.14), (3.15) and (3.16) under these general conditions must be modified since the coefficients of intensity variables and demand rates were derived assuming a period length of one. To obtain the correct discrete-time version of (3.10), we modify (3.14), (3.15) and (3.16) as follows. First for a real number $x$, redefine $x^+$ (redefine $x^-$) to mean the smallest (resp. largest) epoch $t_k \in S$ not less than (resp. not greater than) $x$. Next, substitute the ratios

$$\begin{align*}
\frac{t}{(-LA_i)^+ - (-LA_i)^-}, & \quad \frac{(-LA_i)^+ + LA_i}{(-LA_i)^+ - (-LA_i)^-} \quad \text{and} \quad \frac{(t - LA_i) - (t - LA_i)^-}{(t - LA_i)^+ - (t - LA_i)^-}
\end{align*}$$

(3.17)

for the coefficients $(t)$, $[(-LA_i)^+ + LA_i]$ and $[(t - LA_i) - (t - LA_i)^-]$ in (3.14), respectively. (We define $0/0 = 0$.) The denominators of the ratios in (3.17) express the lengths of the time periods in which the limits of integration lie; the ratios thus express the fraction of intensity in those periods which lies within the limits of integration. Likewise, we substitute the ratios

$$\begin{align*}
\frac{t}{(LB_i)^+ - (LB_i)^-}, & \quad \frac{(LB_i)^+ + LB_i}{(LB_i)^+ - (LB_i)^-} \quad \text{and} \quad \frac{(t + LB_i) - (t + LB_i)^-}{(t + LB_i)^+ - (t + LB_i)^-}
\end{align*}$$

(3.18)

for the coefficients $(t)$, $[(LB_i)^+ - LB_i]$ and $[(t + LB_i) - (t + LB_i)^-]$ in (3.15), respectively. Finally, we modify (3.16) to become, for all $t \geq 0$,

$$U_i(t) = \sum_{r=1}^{t^-} u_{ir} + \frac{(t - t^-)}{(t^+ - t^-)} u_{i t^+}.$$  

(3.19)

Incorporating the modifications (3.17), (3.18) and (3.19), inventory balance constraints (3.10) enforced at the set of epochs $T_i$ for product $i$, $i = 1, \ldots, M$, and capacity constraints (3.4) enforced for each epoch in $S$ form a more general linear programming production model. The overall time grid for this discrete-time model is $\Lambda = \bigcup_{i=1}^{M} T_i$.

In sum, we have developed an accurate yet parsimonious linear programming model for product-generated networks with two different types of lead times for each activity. A time grid (with possibly unequal-length periods) for resource allocation is assumed to be given. Activity input applications are constant on these intervals according to Leontief
domains. Based on the lead times, distinct sets of epochs for each product are established for enforcing inventory balance. The decision variables of the linear program are basically unchanged in our reformulations, and so no changes are proposed to the usual objective functions for various planning applications.

When lead times are integral with respect to the given time grid, our formulation increases model accuracy with no increase in computational cost. When lead times are noninteger, our formulation provides an accurate model with less constraints than the approach of subdividing the time grid enough to make all lead times integral. (Of course, modeling the case that lead times are noninteger with respect to the given grid generally requires more constraints and computational effort than the case that lead times are integral.)

Our more general formulation is by no means the most general linear programming production model. The lead times in the foregoing model serve to relax the assumption of instantaneousness of intermediate product applications, outputs, and transfers that is assumed by linear programming models without time lags. The assumption of instantaneous applications of service resources also can be relaxed, i.e., the assumption of Leontief domains is not necessary for obtaining a linear programming model. For example, Krajewski and Ritzman (1977) propose a formulation with resource input coefficients that distribute resource application over an integral number of time periods between intermediate product application and realization of output. Using the framework, their model can be extended to admit noninteger durations for resource applications.

As a final remark, if there are significant processing times (i.e., noninstantaneous service applications), these times also must be accounted for in the lags separating intermediate product input applications and consequent output of each activity. It is not correct to delete processing times from the lags appearing in inventory balance equations, as suggested in Billington et al. (1986). Moreover, setup times introduce more breakpoints in inventory input and output flows; if setup times are fractional, enforcing inventory balance only at integer times does not guarantee inventory balance throughout continuous time. A full development of linear programming models for the case of activity production functions with noninstantaneous service applications is beyond the scope of this paper; however, linear programming models for particular noninstantaneous cases of interest are developed in Leachman and Boysen (1985), Leachman (1986) and Hackman and Leachman (1989).

4. Manufacturing Resources Planning

Consider an acyclic, product-generated, normal, discrete-time production model with unit-length time periods. There are no exogenous materials in the model. (All raw materials of interest are included as products in the network.) The activities (products) are ordered such that if \( A_i \) supplies intermediate product to \( A_j \), then \( i < j \). Final transfer requirements \( v_{i,N+1} \), \( i = 1, 2, \ldots, N \), are specified for the system. (Since the model is product-generated, we suppress the superscripts on the flows.) Let \( V_i(t) = \sum_{j=1}^{N+1} V_{ij}(t) \) denote the cumulative total transfer requirement of activity \( i \) at time \( t \). Let \( R_i(t) = \max \{ 0, V_i(t) - I_i'(0) \} \). In MRP parlance, \( v_i(t) \) is the "gross requirement" for product \( i \) at time \( t \), while \( r_i(t) \) is the "net requirement" for product \( i \) at time \( t \). Final transfer requirements \( \{ v_{i,N+1}(t) \} \) are termed the "master production schedule". In MRP systems, gross and net requirements, and the master production schedule are expressed as event-based flows on an equal-length discrete time grid such as weeks.

MRP systems include a material requirements planning (mrp) module to determine requirements for products and materials. Coefficients \( \{ a_{ij} \} \) are presupposed, indicating the amount of product \( i \) required as input to produce one unit of product \( j \). So-called "lead times" \( \{ L_{ij} \} \) are also specified. Each item is produced in batches of size \( \{ B_{ij} \} \).
between the event when a transfer requirement for product \( i \) is due and the event when intermediate product inputs used to produce the requirement are withdrawn from inventory by activity \( i \). For \( i = N, N - 1, \ldots, 1 \) and for all \( t \), the following computations are performed recursively:

\[
V_j(t) = \sum_{i=t+1}^{N+1} V_j(t),
\]

\[
R_i(t) = \max \{ 0, V_i(t) - I_i^t(0) \},
\]

\[
V_{ji}(t) = a_i^t R_i(t + L_i), \quad \text{all } j < i.
\]

An additional step is sometimes included between (4.2) and (4.3) to modify the flow \( R_i(t) \) to reflect desirable lot sizes. The (modified) flows \( \{r_i\} \) are the result of mrp calculations; each flow \( r_i \) defines order quantities of product \( i \) due at the time grid points.

MRP systems incorporate an incomplete model of production. Activity application flows, activity production functions and domain constraints are not defined. In general, the requirements curve \( R_i \) for activity \( i \) is not the cumulative output curve \( F_i(Y_i) \) of activity \( i \); similarly, the transfer curve \( V_i \) is not the cumulative curve \( Y_i^t \) of application of product \( j \) at activity \( i \). In order for MRP to be valid, fundamental inventory balance at each activity \( i \) must be ensured, i.e.,

\[
F_i(Y_i) \geq R_i \quad \text{and} \quad Y_i^t \geq V_{ji}.
\]

Instead of specifying \( Y_i \) and \( F_i(Y_i) \), MRP systems replace (4.4) with the relationship (4.3). Attempting to ensure feasibility solely using (4.3) may require excessive work-in-process inventory, as is well-known. To avoid such excess inventory, application flows and conservation of service resources must be made explicit in the model, resulting in a formidable mathematical programming model (Billington et al. 1983, 1986).

Short of analysis of service applications, a simple improvement to the mrp calculation can be made that is analogous to the improvement we suggest for l.p. formulations incorporating time lags. Consider application of MRP to a production system satisfying the assumptions of §3.3. The mrp parameter \( L_i \) must account for time to transfer inputs from predecessors and time to inspect inputs (\( LB_{ji} \)), time to produce (\( LA_i \)), as well as time spent in inventory \( I_i^t \) (i.e., time resulting from differences between actual output and requirements curves). The mrp parameter \( L_i \) in (4.3) must include an allowance for max, \( LB_n \) in addition to other factors. In the case of multiple activity inputs with different lags, some reduction in lead time (and excess inventory) is obtained by substituting \( (LB_{ji} + LA_i) \) for \( L_i \) in (4.3).

To summarize, MRP systems incorporate an incomplete model of production in which all flows are expressed in terms of intermediate product transfers. In contrast, familiar l.p. models express all flows in terms of applications.

5. Critical Path Models

In familiar critical path models (CPM), an acyclic network of activities \( A_1, \ldots, A_N \) is given. Each \( A_i \) is assumed to operate in some uninterrupted interval of time with integer length \( d_i > 0 \). In the activity-on-node format, an arc from \( A_i \) to \( A_j \) means there is a strict precedence relationship between \( A_i \) and \( A_j \). (\( A_i \) cannot be started until after \( A_j \) has finished.)

In the standard application of resource-constrained CPM, resource use by an activity is assumed constant during its duration (Modor et al. 1985).

Resource-constrained CPM embeds a model of production as follows. The activity network is the CPM activity-on-node network, exclusive of the source for exogenous inputs and the sink for final outputs. The underlying production system has the following characteristics.
(1) The production model is acyclic and normal with a finite horizon.

(2) Each activity produces a vector of outputs, whereby a distinct intermediate product is supplied to each follower. Activities with no followers produce a final product transferred to a sink node. Since we may identify uniquely the products of the system with arcs of the network, we replace the notation \( m \) for product with the notation \((i, j)\).

(3) Exactly one unit of each product is produced.

(4) Transfers \( y_{ij}^{(k)} \) are event-based flows. Quantity transferred equals 0 or 1. We take the time grid \( \Lambda \) to be the set of nonnegative integers.

(5) Activity production functions have a restricted Leontief domain. Input applications to \( A_i \) are indexed by a "box" curve of the form

\[
z_i(\tau) = \begin{cases} 
  1/d_i & \text{if } \tau \in (S_i, S_i + d_i], \\
  0 & \text{otherwise},
\end{cases}
\]  

(5.1)

where \( S_i \in \Lambda \) corresponds to the start time of \( A_i \). Domain constraints on allocations are assumed to be trivial.

(6) Cumulative output of an activity is measured in terms of the fraction of required resources that has been applied. We define \( f_i^{(i,j)}(z_i) = z_i \) for each \( j \) such that \((i, j)\) corresponds to an arc in the network. Note that an activity simultaneously produces one product for each follow-on activity.

It is a simple matter to verify that strict precedence is implied by the fundamental inventory balance equations and the assumptions (1)–(6). Hence strict precedence is simply a particular form of inventory balance that arises when there are event-based transfers.

The representation (1)–(6) of CPM in terms of the framework changes according to one's conventions for describing the physical phenomena. Assumption (6) is based on a particular convention for output measurement, e.g., when 50% of the required effort has been made, an activity is considered 50% done. A second representation arises if activity output is viewed as a discrete "lump" emerging exactly when resource applications are complete. In this representation, we relax the integer constraint on \( y_{ij}^{(k)} \) in (4) and replace (6) by

\[
f_i^{(i,j)}(z_i)(\tau) = \begin{cases} 
  1 & \text{if } \tau = S_i + d_i \text{ and } (i, j) \text{ is an arc,} \\
  0 & \text{otherwise.}
\end{cases}
\]  

(5.2)

Now, each production function has Leontief domain but is no longer Leontief.

A third representation arises if instead of the output or transfers being discrete, the follow-on applications of intermediate product are viewed as discrete. In this case, we maintain the original assumption (6), relax the domain for transfers in (4) to simple nonnegativity, and alter (5) such that

\[
y_i^{(i,j)}(\tau) = \begin{cases} 
  1 & \text{if } \tau = S_j \text{ and } (i, j) \text{ is an arc,} \\
  0 & \text{otherwise.}
\end{cases}
\]  

(5.3)

Now, the domains for the production function are no longer Leontief; only service resource applications \( y_k^k \) are indexed by an intensity function (5.1).

In fact, a strict precedence model may be formulated without event-based flows. In this fourth model, there is no inventory of product \((i, j)\) at \( A_i \), i.e., (4) becomes \( y_{ij}^{(k)} = f_i^{(i,j)}(y_j) \). Each production function has restricted Leontief domain exactly as in (5). With these assumptions, we can maintain strict precedence if we relax (6) here.
\[ f_{i,j}^{(i,j)}(z_i)(\tau) = \begin{cases} \frac{1}{d_j} & \text{if } \tau \in (S_i + d_i, S_i + d_i + d_j] \quad \text{and} \quad (i, j) \text{ is an arc,} \\ 0 & \text{otherwise.} \end{cases} \]

(5.4)

The production function defined in (5.4) maps the intensity curve for \( A_i \) into the earliest intensity curve for \( A_j \) consistent with the assumption of strict precedence. That is, the production function determines the appropriate bound on the choices for intensity of \( A_i \). This last representation is the only one of the four which is preserved under aggregation of resource-constrained CPM networks; using this model, a computationally tractable approach to multi-project aggregate planning has been developed (Hackman and Leachman 1989, Leachman and Boysen 1985). See Figure 3 for a pictorial representation of the four alternative models.

Comparing CPM to the basic l.p. model in §3.2, note that both models utilize the same production function, except that application flows in CPM have the severely restricted domain defined by (5.1). It is natural to consider relaxation of these domain constraints to allow flexibility in activity operations akin to that allowed by the activity analysis model in linear programs.

A number of authors have developed extensions in this regard, which we now categorize in terms of allowed domains for application flows. Wiest (1967) and Talbot (1982) have considered resource-constrained scheduling when there are discrete alternatives for the

---

**Figure 3.** Alternative Representations of CPM.
duration (and associated resource requirements) of each activity. However, from the point of view of the framework, each alternative still corresponds to a "box" intensity curve for each activity, i.e., a constant rate of resource application from start to finish of an activity. Leachman (1983) develops a technique for resource leveling when there is a continuous range of possible intensity levels for each activity, i.e., (5.1) is replaced by

\[
z_i(\tau) = \begin{cases} 
\frac{1}{d_i} & \text{if } \tau \in (S_i, S_i + d_i] \text{ for some } S_i \in \mathbb{R}_+ \text{ and } \frac{1}{d_i} \in [z_i, \bar{z}_i], \\
0 & \text{otherwise.}
\end{cases}
\tag{5.5}
\]

Weglarz (1981) and Leachman et al. (1989) relax this domain even further in considering resource-constrained scheduling. They develop scheduling algorithms that exploit the assumption that intensity may vary during the duration of an activity, i.e., (5.1) is replaced by

\{z_i: \text{For some } S_i, F_i \in \mathbb{R}_+, z_i \text{ is nonzero only on} \}

the interval \([S_i, F_i], \text{ and for } \tau \in [S_i, F_i], z_i \leq z_i(\tau) \leq \bar{z}_i\}. \tag{5.6}

6. Summary and Suggested Use of the Framework

A model of production is a set of constraints on the decision variables defined for controlling a production process that ensure that values proposed for the variables are technologically feasible. A framework has been introduced for methodically constructing the constraints from explicit physical assumptions about the production process. The constraints express conservation through continuous time of all services and materials of interest. We have used the framework to generalize and improve the accuracy of the simple models of production embedded in linear programming, MRP and CPM planning techniques.

The framework serves as a useful tool promoting a scientific approach to formulating models of real production processes. It is an axiomatic system delineating elements and relationships that should be included in any model of production. To use the framework, a management scientist elucidates physical assumptions about the process, including the production functions and the domain constraints on applications, allocations and transfers. From these assumptions the analyst derives the model’s constraints. The justification and explanation of the proposed model are reduced to justification and explanation of the analyst's physical assumptions, thereby facilitating the evaluation and acceptance of the model by management and by other management scientists. Moreover, the careful and complete derivation of constraints through continuous time typically yields a more accurate model than otherwise, as demonstrated in §3. Admittedly, we are proposing that considerably more effort should be expended on formulation by the management scientist.

Once an acceptable abstract formulation is completed, we suggest that the actual numerical formulation should be automatically generated by software written by the analyst that is interfaced with commercial factory floor data collection systems. Such systems compile statistics on lead times, resource input requirements, resource availabilities, etc., as well as provide the current status of inventory and work-in-process. As statistics on the production process evolve over time, so can the numerical formulation via the software interface.

This approach to model formulation and implementation already has been demonstrated by a planning model now in use in the semiconductor industry. The reader is referred to Leachman (1986) for a description of the dynamic production functions modeling complex semiconductor fabrication sequences and to Leachman (1986) and
from interfaces with factory data collection systems. The framework also has been utilized to generate a linear programming-based model for high-level multi-project planning (Leachman and Boysen 1985 and Hackman and Leachman 1989).²

² This research was supported in part by the Office of Naval Research and the Puget Sound Naval Shipyard under Contract N00014-76-C-0134 with the University of California at Berkeley, and by a gift to the University of California at Berkeley from Intel Corporation. The authors are grateful to John Vande Vate, John Harrison, Renato Monteiro and the anonymous referees for insightful comments that improved the exposition from earlier drafts. Encouragement from our colleagues John Bartholdi, Loren Platzman, Craig Tovey, Betsy Greenberg and Roger Glassy also is gratefully appreciated.

References


