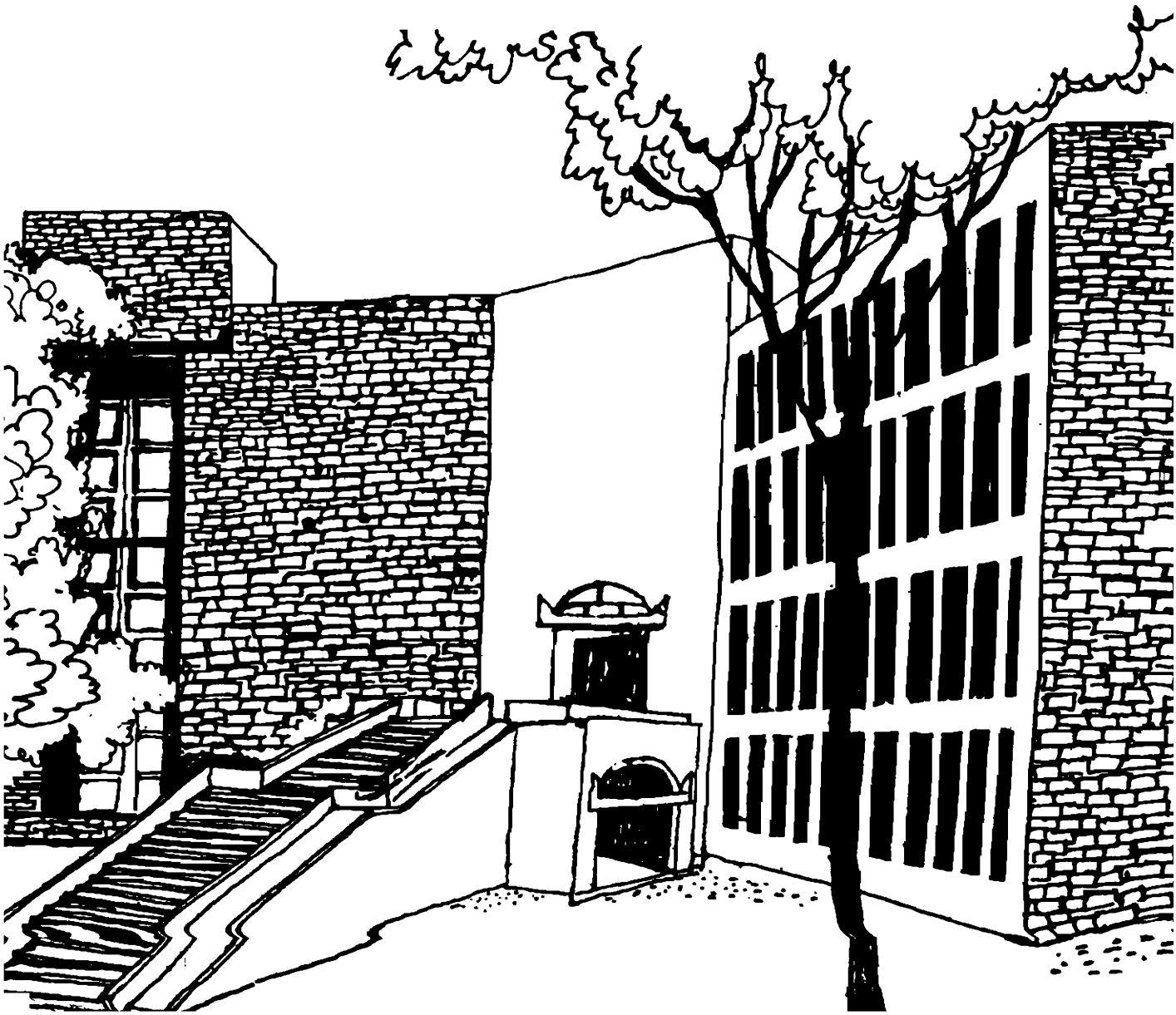
विद्याविनियोगाद्विकासः

IIM AHMEDABAD

Working Paper



MANAGING BATCH PROCESSORS TO REDUCE LEAD TIME IN A SEMICONDUCTOR PACKAGING LINE

By

Pankaj Chandra
Sudheer Gupta

W P No.1302
April 1996

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD - 380 015
INDIA

Managing Batch Processors to Reduce Lead Time in a Semiconductor Packaging Line

Pankaj Chandra †

Sudheer Gupta ‡

† Indian Institute of Management

Vastrapur, Ahmedabad - 380015, India

Tel: 001-91-79-407241 / Fax: 001-91-79-6427896

email: chandra@iimahd.ernet.in

‡Faculty of Management, McGill University

Montreal, PQ, H3A 1G5, Canada

Tel: (514) 398-4000 / Fax: (514) 398-3876

email: gupta@management.mcgill.ca

Revised November 1995

This research was supported by NSERC-Canada Grant OGP00425 and the Management Sciences Research Center, McGill University. The authors are grateful to Gora Ganguli and Louise Legendre of IBM, Bromont for suggesting the problem, providing data & information and for extensive discussions on the project. The authors also wish to thank Rafaela Del Pilar and the anonymous referees for their helpful comments and suggestions.

Managing Batch Processors to Reduce Lead Time in a Semiconductor Packaging Line

Abstract

In this paper we study a semiconductor packaging line at IBM Bromont. At the line, modules are assembled and then tested in a Burn-In oven. The Burn-In oven is a batch processing station. We outline a procedure to determine order release schedule and lot sizes for the various work stations in the line, such that total manufacturing lead time is minimized. The internal parameters of the procedure are set by simulation experiments and by heuristics. Sensitivity analysis is carried out to determine the robustness of the procedure with respect to various external parameter settings.

(LEAD-TIMES, LOTSIZES, ORDER-RELEASE SCHEDULE, SIMULATION, HEURISTICS)

1 Introduction

Shop floor synchronization is emerging as a key factor in the quest to reduce lead time in manufacturing operations. But the relationships between lead time and other parameters relevant to the operations are not very well defined, especially for multi-stage, multi-product manufacturing systems. Moreover, the complexity of the modeling process and computational difficulty have prevented the development of generic algorithms that give consistently good schedules. The presence of situation specific peculiarities force one to look into tailor-made approaches for solving shop floor planning problems.

This paper addresses a problem that is encountered in several industries where a batch processor is a part of a manufacturing line. A batch processor can process more than one unit of a product at a time (unlike a discrete processor which processes one unit at a time). Once the batch process begins, it cannot be interrupted until the entire batch is processed. Examples of these processors include burn-in ovens in a semiconductor packaging line or a printed circuit board manufacturing line, diffusion and oxidation stages in wafer fabrication, and heat treatment facilities found in steel and ceramic industries. Usually, the processing time on a batch processor is the longest amongst all stations in such manufacturing lines. Consequently, any effort to synchronize production will have to consider the impact of the batch processor scheduling on the scheduling decisions for the rest of the manufacturing line and vice-versa. However, the issues involved in integrating batch processors in an overall scheduling framework, in a multi-stage multi-product manufacturing facility, are not very well understood as yet. More case studies are needed to flesh out the issues that will have to be addressed in any generic algorithm for such manufacturing lines. We model the related decision making process in the context of a 36mm semiconductor packaging line at the IBM plant in Bromont. In the following paragraphs, we first describe the structure of this complex packaging line and then highlight the issues under consideration.

The IBM plant in Bromont produces 700 different end items called "modules". Production of a module involves mounting a set of chips on a ceramic substrate. The entire process is known as "semiconductor packaging". The manufacturing line consists of 12 different processes (Figure 1). The production process can be divided into 2 sections. In the first section, called the **Assembly Section**, a chip is de-greased, cleaned, mounted on a substrate, encapsulated, epoxy-cured in an

oven, wave soldered and then inspected. There are setups at most workcenters. Some workcenters have multiple parallel machines and parts reenter a particular station (i.e., de-greasing) more than once. An important station in this section is the Epoxy Cure (EC) oven. The EC ovens are batch processors of a fixed capacity and constitute the longest processing time in the whole Assembly Section. Different module types can be put in these ovens in any combination. At the end of this section, the modules are racked and shipped to the second section of the packaging line.

In the second section, called the **Burn-In Section**, modules are tested in specialized Burn-In ovens. There are multiple ovens in this section whose purpose is to test modules on eventual failure by subjecting them to different temperature and voltage stimuli. The information on these stimuli is stored on "program cards" that are unique for each module type. Each oven has several distinct chambers of equal capacity. A module is placed in a special board called the "burn-in board" (bib). A program card is attached to each bib before the bib can be placed in a chamber of an oven. A bib can hold a fixed number modules of the same type. There is a limit on the number of bibs that can go in each chamber. There are incompatibility constraints to the effect that only one type of module can be placed in a chamber of an oven. In addition, only module types belonging to the same family can be tested together in the same oven. The availability of bibs and program cards is limited as they are quite expensive. This restricts the quantity of each module type that can be loaded at any time in an oven for a test run. The processing time at the Burn-In ovens is the longest, amongst all stations, in this manufacturing line (Table 1). The processing time is fixed, irrespective of the number of modules that are put in the oven. Needless to say, it is quite expensive to run these ovens. Since the ovens are quite expensive, buying additional ovens was not a feasible option for the current study. Upon completion of processing at the ovens, all modules are tested. The rejected modules are scrapped while the acceptable ones are stored for shipment.

The semiconductor packaging industry competes on the basis of high volumes and low lead times. The plant is evaluated on the percentage of delivery commitments met. Monthly demand estimates are provided and updated each week. Thus the demand is deterministic over the weekly planning horizon and production is expected to meet the weekly delivery dates. We assume that the weekly demand can be met, if production is planned properly. The plant operates on a single shift of eight hours duration. In case of very high demand, excess capacity in terms of extra shifts and additional lines could be used.

The problem can, then, be stated as follows: we want to determine the loading plan for the Burn-In ovens (i.e., quantity and type of module that should be loaded in every chamber of each run of the ovens) and the order release schedule (i.e., release dates, sequence, and the quantity released) for different module types in the manufacturing line such that lead time is minimized. We develop a planning methodology which incorporates the characteristics of the batch processor in designing order release schedules for the packaging line. In addition, we address the question that is often asked on the shop floor - how full should an oven be before it can be run?

The remainder of the paper is organized as follows: in the next section, we briefly review the related literature and outline the contributions of this paper. In section 3 we describe the assumptions, data, definitions, and the key objective for this case study. In section 4 we outline an iterative procedure which combines optimization and simulation models for determining order release schedule and the loading plan for the Burn-In ovens in order to minimize the lead time. An example is provided to illustrate the solution procedure. The various issues that arise while designing an order release policy that matches the batch processor requirements are discussed in section 5. We conclude, in section 6, with a discussion of the computational results and future research directions.

2 Literature Review

There is quite an extensive treatment of deterministic flow shop scheduling problems in literature. However, as pointed out by Dobson et al. [1987], the key assumption in most of these efforts is that processing time of a job on a machine is known and fixed. In reality, the effective processing time of a batch (i.e., the sum of setup, processing, and waiting times) varies with the lotsize even when the processing times are deterministic. The effects of lot size on the extent of queueing delays and lead time related performance measures, in case of a single machine, are well documented (see Dobson et al. [1987] and Karmarkar [1987a 1987b]). In general, multi-stage manufacturing problems have remained analytically intractable. As a consequence, simulation has been commonly used to analyse such systems. McDowell and Randhawa [1989] and Lin et al. [1987] have developed simulation based decision support systems for analyzing printed circuit board fabrication lines. Karmarkar et al. [1985] developed a simulation/analytical model for examining the characteristics of

a manufacturing cell and for devising appropriate lot sizing policies at Eastman Kodak's Apparatus Division. None of the above papers, however, address the additional complexities imposed by batch processors.

Most papers on batch processing operations (Ikura and Gimple [1986], Glassey and Weng [1991], Fowler et al. [1992], Dobson and Nambimadom [1992], and Lee et al. [1992]) have considered such a machine in isolation of a larger manufacturing line (i.e., as a single stage problem). They have determined the best loading plan for the batch processor under different assumptions. Lee et al. [1992] have highlighted the need to integrate these models and the related algorithms in a larger modeling system which is capable of linking the batch processing decisions with scheduling decisions on other machines on the shop floor. It is not obvious that the single-stage batch processor algorithms will remain valid in a multi-stage environment as well, due to interaction effects with other machines and operations. Ahmadi et al. [1992] present the only paper where a system of two machines is modeled, at least one of which is a batch processor. For a number of different configurations of the batch and discrete processors, they provide polynomial time algorithms for the objectives of minimizing makespan and the sum of completion time. However, they do not consider setups at the discrete processor. At the same time, the batch processor operates without any constraint except on capacity.

To the best of our knowledge, this is the first attempt in successfully solving a large scale problem with batch processors. Recently, Bhatnagar et. al. (1993) have used the general approach which is presented in this paper to develop a methodology for managing batch processors in an entirely different industrial context. The contributions of this paper can be summarized as follows:

- we have addressed an important and a complex problem that is found in several industries but has not been addressed adequately in the literature;
- we have designed an intuitive procedure for solving the lead time minimization problem – the approach is quite general and is useful in managing other types of lines with batch processors;
- we have modeled and solved a new variant of the bin packing problem – the heuristic gave optimal solutions for all the test cases that we solved;
- we have established the effects of lot sizing and release decisions on lead time and utilization related performance measures for lines with batch processors;

- we have established the efficacy of this approach by illustating its usefulness in a real manufacturing environment.

3 Assumptions, Data, Definitions & Objective

We will describe the key assumptions that govern the planning process, some definitions as well as the objective of the problem. For this study, data was provided by IBM's Bromont plant. We have disguised some of this data for purposes of confidentiality. We use a snapshot picture of the weekly production requirement for our analysis. The details are given below:

Assumptions

- the planning horizon is four weeks and the demand forecast is updated each week;
- production is scheduled on a rolling horizon of one week and the schedule is frozen for this duration (i.e., the lot sizes & order releases are planned for a period of one week);
- weekly demand estimates incorporate the impact of variability due to failures at the ovens.

Data

- the setup and processing times on each machine are given in Table 1;
- number of EC ovens available = 5;
- number of Burn-In ovens available = 4;
- capacity of each bib = 10 modules;
- capacity of each chamber in a Burn-In oven = 10 bibs;
- number of chambers in each Burn-In oven = 6.

Definitions

The following definitions are used in the paper:

- Input Lot Size: the lot size of each module type which is released in the Assembly Section. The input lot sizes, the sequence in which they are released, and the release dates define an Order Release Schedule. We study two strategies for determine the Order Release Schedule:

(a) 'Lot for Lot Strategy': Input lot size for each module type is set equal to its weekly demand and lots are released in the descending order of their size. This reflects the current scheduling practice at the plant.

(b) 'Batching Strategy': Input lot size for each module type per shift is determined by the Synchronized Pull procedure (which is described in the next section).

- Burn-In Oven Lot Size: the quantity of each module type to be loaded in each chamber of each oven run, for the entire week.

- Minimum Oven Load (MOL): the minimum quantity of modules required to start a run of an oven. The assumption here is that an oven run cannot be initiated unless enough modules (of different or same type) have been produced to equal the specified Minimum Oven Load. The maximum capacity of a Burn-In oven is 600 modules (i.e., $1 \leq$ MOL for the Burn-In oven ≤ 600), while the maximum capacity of an EC oven is 1400 modules (i.e., $1 \leq$ MOL for EC oven ≤ 1400).

Objective

The objective of this study is to minimize the manufacturing lead time while meeting the demand requirement of the different module types over a week (i.e., the makespan should not exceed seven shifts). We define lead time as the average time that a module spends in the system, or, mathematically

$$\left\{ \frac{\sum_{all\ modules} (\text{time that a module spends in the system})}{(\text{total number of modules})} \right\}.$$

4 A Synchronized Pull Procedure

In this section we describe the methodology which we have developed in order to solve the lead time minimization problem. The approach adopted here is to let the bottleneck station, i.e., the Burn-In Section, pull its requirements from the Assembly Section (Figure 2). We have designed a **Synchronized Pull Procedure** which implements this control mechanism. It determines the order release policy in accordance with the requirement of the batch processors.

The Synchronized Pull procedure (which is described in detail in section 4.1), first, determines

the minimum number of oven runs required to meet the demand, and the quantity and type of a module that should be loaded in each chamber of every oven run. This is determined by solving the Oven Loading Problem which is formulated as a variant of the bin packing problem (described in detail in section 4.2). This establishes the requirements of the Burn-In Section in an ideal situation (i.e., if all the modules were available for testing at the beginning of the week).

Second, we determine the Order Release Schedule using the 'batching strategy'. The input lots are *derived from* from the solution of the Oven Loading Problem. We consolidate, across all ovens in each shift, the quantity for each module type that is to be loaded in each chamber of an oven (i.e., from the solution of the Oven Loading Problem). This constitutes the Input Lot Size for that module in that shift (see section 5 for a discussion on consolidation of lots). The resulting input lots are sorted in a descending order of size, for each shift. This constitutes the Order Release Schedule.

Third, the Order Release Schedule becomes an input to a simulation model of the entire manufacturing line which is used to study the performance of the schedule. Modules are processed in the Assembly Section and are then loaded into the ovens as they (and other resources like bibs and program cards) become available. The oven is run once a Minimum Oven Load (which is specified externally) has accumulated. In other words, rather than loading the ovens as per the ideal plan that was obtained from the solution of the Oven Loading Problem, we modify the oven loading by using the MOL criterion. The resulting loads in the ovens define the Burn-In Oven Loading Plan.

Fourth, once all the modules have been tested in the ovens, performance statistics are obtained on makespan, average lead time, etc. The procedure is repeated for different specifications of the MOL for Burn-In and EC ovens. The value of MOL for which minimum lead time is obtained is selected. If the makespan, corresponding to the selected MOL, does not violate the due date for the entire weekly demand then we have a feasible solution. If the due date is violated, we modify the Order Release Schedule. We change the input lots by trading off the reduction in setups in the Assembly Section against an increase in wait at the Burn-In ovens. The modified lots are once again released into the simulation model and performance is evaluated for different values MOL. This iterative process is carried on until either feasibility is achieved or the capacity is found inadequate for meeting the demand (in which case extra shifts are recommended).

The simulation model of the packaging line was developed using GPSS/H. The input to the

simulation model consists of demands for each module type, number of bibs and program cards available for each module type, setup and processing times at each workstation, capacity of the ovens, Order Release Schedule, MOL for Burn-In and EC ovens etc. The system is first primed with a week's demand. Then the schedule is released into the line and oven runs are simulated.

4.1 Details of the Synchronized Pull Procedure

A step-by-step description of the **Synchronized Pull Procedure** is given below:

STAGE 1: Solve the Oven Loading Problem and determine the Order Release Schedule.

Step 1: Determine the maximum number of units (L_i) of each module type i that can be packed in the ovens in each shift, i.e., $L_i = min(d_i, b'_i, p'_i)$.

Step 2: Split these quantities, L_i, into chamber loads and pack them into ovens.

Step 3: Determine the Order Release Schedule.

(Please see section 4.2 for details of Steps 1 - 3.)

STAGE II: Determine the makespan for the Order Release Schedule determined in Stage I, and the corresponding lead time (using the simulation model).

Step 4: Run simulation with different specifications of the of the Minimum Oven Loads for the EC and Burn-In ovens. Choose the Minimum Oven Loads corresponding to the minimum lead time. For the chosen Minimum Oven Loads, measure the makespan (T_c).

Step 5: Determine feasibility of the schedule.

Let T_b = due date for the entire demand. IF $T_c \leq T_b$ THEN stop, ELSE, go to Step 6.

STAGE III: Modify the Order Release Schedule to remove any infeasibility.

Step 6: Calculate the time violation, $\delta = T_c - T_b$.

Starting from the last shift, i.e., with $t = T$

Step 7: Estimate Q, the number of units whose production time is equivalent to violation time δ (i.e., the load to be shifted from shift t to earlier shifts).

$Q = \frac{\delta * \Omega}{\Gamma}$, where

Ω = total number of modules in shift t = $\sum_i \omega(i,t)$

Γ = estimate of time spent by shift t's load in the system

= (total setup time + total processing time + estimate of cure oven run time) for shift t's load

= $\sum_{all\ machines}$(no. of setups per machine)*(setup time per machine) +

$\sum_{all\ machines} \sum_i [\omega(i,t)$*(processing time per module per machine)] +

$\lceil \frac{\Omega}{MOL\ for\ EC\ Ovens} \rceil$*(EC oven run time).

Step 8: Determine the product mix of Q that is to be removed from shift t, and the location of an earlier shift(s) where Q should be scheduled.

a) First, an attempt is made to remove the modules for which bibs and program cards are available in an earlier shift *and* if setup reduction is possible.

FOR index α = 1,t-1

FOR module i = 1,n

IF $(0 < \omega(i,t) \leq Q)$ THEN

IF $(\omega(i,t-\alpha) \neq 0)$ THEN

IF $\omega(i,t-\alpha) + \omega(i,t) \leq \min(b_i, p_i)$ THEN

$\omega(i,t-\alpha) = \omega(i,t-\alpha) + \omega(i,t)$

$Q = Q - \omega(i,t)$

$\omega(i,t) = 0$

ELSE go to next i.

ELSE go to next i.

ELSE go to next i.

IF $(Q \neq 0)$ THEN go to next i ELSE go to Step 9.

IF $(Q \neq 0)$ THEN go to next α ELSE go to Step 9.

(b) IF (Q $\neq$ 0) THEN $t = t - 1$

IF (t $\neq$ 1) THEN go to Step 8(a) ELSE go to Step 8(c).

ELSE go to Step 9.

(c) In this step, starting with the last shift, units are removed to reduce setups in the Assembly Section, ignoring the availability of bibs/program cards. That is, starting with $t = T$, the procedure in Steps 8(a) and 8(b) are repeated but without the condition that $\omega(i,t-\alpha) + \omega(i,t) \leq \min(b_i, p_i)$.

Step 9: Go to **STAGE II**.

4.2 The Oven Loading Problem

We describe here the Oven Loading Problem and the two-pass heuristic that has been developed to solve this problem. Assuming that all modules are available in the Burn-In Section, we want to determine the number of oven runs needed each shift and the loading plan for each oven run (i.e., the quantity and type of module to be loaded in each chamber of each oven run). A related mathematical problem is the bin packing problem. The bin packing problem has been shown to be NP-complete (Karp[1972]). Heuristic procedures for variants of this problem have been suggested by Hall et. al. [1988], Leung [1989], Raman [1989] and Johnson [1973]. We formulate the Oven Loading Problem as a variant of the bin packing problem. Owing to the presence of additional constraints on the availability of bibs and program cards, families of module types, capacity restrictions on chambers and ovens, etc., the model formulated in this paper differs from those suggested in the above efforts.

We now define the Oven Loading Problem. Let,

i = index for module type = $1, 2, \ldots, n$

j = index for ovens = $1, 2, \ldots, m$

k = index for chambers in each oven = $1, 2, \ldots, s$

t = index for shifts = $1, 2, \ldots T$

g = index for family of module types = $1, 2, \ldots v$

Also, let R be a set $\{ = R_1, R_2, \ldots, R_v\}$ where each element is a unique family of module types. Only module types within a family can be tested together in the same oven. No two families are compatible (i.e., $R_v \cap R_u = \emptyset$ for $v \neq u$).

Parameters

d_i = demand of module type i per week

q = capacity of each bib

b_{R_v} = number of bibs available for module type i

p_i = number of program cards available for module type i

C = capacity of each chamber

w_{jt} = positive number such that $w_{11} < w_{21}, w_{21} < w_{12}$, etc.

Then, $b_i' = b_i * q$, and $p_i' = p_i * q$.

Variables

x_{ijkt} = quantity of module type i loaded in the k^{th} chamber of of the j^{th} oven in shift t

$$y_{ijkt} = \begin{cases} 1 & \text{if module type } i \text{ is loaded in the } k^{th} \text{ chamber of the } j^{th} \text{ oven in shift } t \\ 0 & \text{otherwise} \end{cases}$$

$$z_{R_v jt} = \begin{cases} 1 & \text{if a module type belonging to family } R_v \text{ is loaded in the } j^{th} \text{ oven in shift } t \\ 0 & \text{otherwise} \end{cases}$$

The **Oven Loading Problem** can be formulated as

$$\text{Min} \quad \sum_i \sum_j \sum_k \sum_t w_{jt} . y_{ijkt}$$

subject to

$$\sum_j \sum_k \sum_t x_{ijkt} \geq d_i \qquad \forall i \qquad (1)$$

$$\sum_k \sum_j x_{ijkt} \leq p_i' \qquad \forall i,t \qquad (2)$$

$$\sum_k \sum_j x_{ijkt} \leq b_i' \qquad \forall i,t \qquad (3)$$

$$\sum_i y_{ijkt} \leq 1 \qquad \forall j,k,t \qquad (4)$$

$$x_{ijkt} \leq C y_{ijkt} \qquad \forall i,j,k,t \qquad (5)$$

$$z_{R_v jt} \geq \sum_{i \in R_v} \sum_k (y_{ijkt})/s \qquad \forall R_v, j, t \qquad (6)$$

$$\sum_{R_v} z_{R_v jt} \leq 1 \qquad \forall j,t \qquad (7)$$

$$x_{ijkt} \geq 0 \ \& \ \text{integer}, y_{ijkt}, z_{R_v jt} \in \{0,1\} \ \forall i,j,k,t,R_v \qquad (8)$$

The objective function ensures that chambers in lower-numbered ovens in earlier shifts are given preference while loading modules in the ovens. Constraints (1) ensures that the demand for each module type is met for the planning horizon. Constraints (2) and (3) are the restrictions imposed by the availability of program cards and bibs respectively in each shift. Constraint (4) ensures that a chamber never gets loaded with more than one type of module, while constraint (5) ensures that

the capacity of any chamber is not exceeded. Constraints (6) and (7) prevent modules from two incompatible groups to be packed together into the same oven.

We now describe a **two-pass heuristic** for solving the Oven Loading Problem:

a) First Pass:

FOR shift t = 1,T

FOR family g = 1,v

FOR all module types $i \in g$,

(i) Determine the maximum number of modules (L_i) of each module type i that can be packed in the ovens: $L_i = min(d_i, b'_i, p'_i)$.

(ii) Split each (L_i) into ($\lfloor L_i/C \rfloor$) full chamber loads of size C, and a partial chamber load of size ($L_i - \lfloor L_i/C \rfloor * C$).

(iii) Sort these chamber loads in descending order.

(iv) Start from the top of the sorted list, and assign each chamber load to first oven-chamber-shift combination available (ensuring that constraints (4), (6) and (7) are not violated). Assigned loads are deleted from the list.

(v) Continue till the list becomes empty OR no more oven-chambers are available in this shift.

(vi) Determine the remaining demand for each module type, i.e., the remaining number of units to be packed in subsequent shifts: $d_i = d_i - \sum_{j,k} x_{ijkt}$.

b) Second Pass:

FOR shift t = 1,T-1

FOR each module type i^* with a partial chamber load in shift t, i.e., ($0 < x_{i^*j^*kt} < C$),

FOR shift (t^*) = t+1,T

(i) Scan for another chamber with part i^* and a partial chamber load, i.e., ($0 < x_{i^*j^*k^*t^*} < C$).

(ii) IF such a chamber (j^*, k^*) is found THEN

IF ($x_{i^*jkt} + x_{i^*j^*k^*t^*} \leq$ C) AND ($min(b'_{i^*}, p'_{i^*}) \geq \sum_{j,k} x_{i^*jkt^*} + x_{i^*jkt}$) THEN

shift quantity x_{i^*jkt} of module type i^* from shift t to chamber k^* of oven j^* in shift t^*.

Note that the first pass of the heuristic can lead to the use of more than optimal number of chambers, if there is any module i with a demand of the form, $d_i = a.C + X$, and $min(b'_i, p'_i) = b.C + Y$, (where, C = capacity of a chamber; a and b are positive integers such that $b \leq a$; and X and Y are

positive integers < C). Let's call the above conditions as **condition (A)**. If we denote the first-pass heuristic solution by Z_{heur}, and the optimal number of chambers required to pack in all modules by Z_{opt}, then it is observed that: $Z_{heur} \leq Z_{opt} + N(a - b + 1)$, where N is the number of different module types satisfying **condition (A)**. The worst case occurs when $b < a$; $Y < X$; $Y \neq 0$, and $X \neq 0$.

The second pass of the heuristic is designed to overcome this deficiency. It examines all the partially filled chambers and tries to combine them, subject to the constraints (1) - (7) in the Oven Loading Problem. Table 2 compares the performance of the two-pass heuristic with the optimal solution to the Oven Loading Problem by using IBM's OSL [OSL, 1991]. The heuristic finds the optimal solution in all cases that were studied.

Determining the Order Release Schedule:

FOR shift t = 1,T

FOR module type i = 1,n

calculate the lotsize of module i to be released in the Assembly Section (Input Lotsize) as:

$\omega_{it} = \sum_{j,k} x_{ijkt}$.

Sort ω_{it} in descending order.

4.3 An Example

We illustrate below, with the help of a small example, the details of the Synchronized Pull procedure. The data used in the example are as follows:

Number of ovens (m) = 1;	Number of chambers per oven (s) = 6;
Capacity of each chamber (C) = 100 modules;	Capacity of each bib (q) = 10 modules ;
Number of shifts per day = 2 (8 hours each);	Due date for the demand = 2 days (4 shifts).

The setup and processing times are as shown in Table 1. All module types belong to the same

family, i.e., they can be tested together in the same oven.

Module Type (i)	Family Type (g)	Demand (d_i)	No. of bibs (b_i)	No.of Program Cards (p_i)
1	1	50	10	10
2	1	230	10	10
3	1	600	20	20

The SYNCHRONIZED PULL procedure is applied as follows:

STAGE I : Solve the Oven Loading Problem and determine the Order Release Schedule (using the procedure described in section 4.2).

Step 1: Determine the maximum number of units of each module type that can be packed into the oven in the first shift:

Module Type	Quantity
1	min (50, 100, 100) = 50
2	min(230, 100, 100) = 100
3	min(600, 200, 200) = 200

where, $b'_1 = 100$, $b'_2 = 100$, $b'_3 = 200$; $p'_1 = 100$, $p'_2 = 100$, $p'_3 = 200$.

Step 2: Split these quantities into chamber loads and pack them into the oven as follows:

Module Type	Oven No.	Chamber No.	Quantity loaded
3	1	1	100
3	1	2	100
2	1	3	100
1	1	4	50

Calculate the remaining demand that is yet to be released:

Module Type	Remaining Demand
1	50 - 50 = 0
2	230 - 100 = 130
3	600 - 200 = 400

With this remaining demand for each module type, repeat Steps 1 - 2 to determine the loading plan for the remaining shifts. The final loading plan (i.e., the solution to the Oven Loading Problem)

for all shifts is shown in Figure 3.

Step 3: The Order Release Schedule for all the shifts is determined as follows:

a) Calculate the Input Lotsizes for each shift, t, as $\omega_{it} = \sum_{j=1}^{m} \sum_{k=1}^{s} x_{ijkt}$, $\forall i$.

b) Sort these lotsizes ω_{it} in descending order.

The final release schedule, then, is given as:

Module Type	Shift No. 1	Shift No.2	Shift No. 3
3	200	200	200
2	100	100	30
1	50	-	-

The shift numbers correspond to different oven runs.

STAGE II: Use the simulation model to determine the makespan and the corresponding lead time for the Order Release Schedule determined in Stage I.

Step 4: Run simulation with different specifications of the of the MOL for the Burn-In and EC ovens. Here, the best MOL is found to be 300 for the EC ovens, and 360 for the Burn-In ovens. Measure the makespan. T_c = 1954 minutes.

Step 5: Determine the feasibility of the schedule.

T_b = 4 shifts * 8 (hours per shift) * 60 (minutes per hour)= 1920 minutes

The solution is not feasible since $T_c > T_b$.

STAGE III: Modify the Order Release Schedule to remove any infeasibility.

Step 6: Determine the time violation (δ) of the due date.

$\delta = T_c - T_b$ = 1954 - 1920 = 34 minutes.

Step 7: Estimate the load (Q) to be shifted from the last shift to earlier shift(s).

* Total number of modules in the last shift = 230
* Total setup time for the last shift's load = 180 minutes
* Total processing time for the last shift's load = 98.4 minutes
* Estimate of EC oven run time for the last shift's load = $\lceil \frac{230}{300} \rceil$ * 420 = 420 minutes
* Estimate of the total time taken by the last shift's load in the Assembly Section

$= 180 + 98.4 + 420 = 698.4$ minutes.

* Estimate of the number of units, Q, whose production time is equivalent to the violation time $= \frac{230*34}{698.4} \approx 11$ units.

Step 8: Determine the product mix of Q, i.e., module type (and its quantity) to be shifted, and the location of an earlier shift(s) where this load should be scheduled.

In this example it is clear, by inspection, that module type 2 is the best candidate for shifting, as its lotsize in the last shift is closest to Q, and, a reduction in setups is possible by consolidating the lots for module type 2 in Shifts 2 and 3. The modified Order Release Schedule becomes:

Module Type	Shift 1	Shift 2	Shift 3
3	200	200	200
2	100	130	-
1	50	-	-

Step 9: Go To **STAGE II.**

* Measure the makespan for the modified order releasesequence. $T_c = 1919$ minutes.
* Is the solution feasible?

  The solution is feasible since $T_c < T_b$.

The modified Order Release Schedule results in a reduction of 12 setups in the Assembly Section (across various machines). This reduction in setups, however, is counter-balanced to some extent by an increased wait at all machines (because of larger lot for module type 2), and by an increased wait at the Burn-In oven (because of the non-availability of bibs and program cards). However, in this particular example, the impact of reduction in setups dominates. This example also illustrates that even with adequate capacity (at the Burn-In ovens), the constraints on bibs and program cards can result in waiting at the ovens or in other words the wait time is affected by the characteristic of the batch procesor.

5 Discussion of Issues

In this section we discuss the issues involved and the impact of various parameters set in the Synchronized Pull procedure.

I. Effect of Order Release Schedule: In an environment with known demand, setups and processing times, the following two components of the average lead time can be affected by an order release policy: waiting time at each workstation in the Assembly Section and waiting time in the Burn-In Section. The first component is dependent on the lot sizes of different module types at each workstation in the Assembly Section as well as the sequence in which these lots are processed. The second component of the lead time is determined by: (a) waiting due to non-availability of bibs and/or program cards; (b) waiting due to non-availability of the Burn-In ovens; and (c) waiting for enough units to be processed from the Assembly Section in order to satisfy the Minimum Oven Load specified for the Burn-In ovens.

Therefore, the input lot sizing decisions affect lead time in two ways. The first effect (corresponding to the first component of lead time) is due to the well-established relationship between lot sizes and queueing. As lot sizes are reduced, the queuing time at each workstation goes down. This results in a corresponding reduction in lead time upto a certain point, beyond which any further reduction in lot sizes results in higher lead time due to increased waiting implied by a large number of setups and the resulting loss in capacity (see Karmarkar, 1987a,b).

The second effect of the input lot sizing decision (corresponding to the second component of lead time) is due to the variation in the product mix and quantities of different types of modules produced by the Assembly Section, in a given amount of time. Due to the limited number of bibs and program cards available for each module type and the incompatibility restrictions on processing different modules together in an oven, some of the modules may have to wait for a long time before they finally get processed in the Burn-In Section. On the other hand, if enough modules are not produced by the Assembly Section to fill an oven by the time it is run, then more oven runs will be required to process all the modules. This may increase the lead time considerably. Besides, the cost of oven runs would increase.

It is observed that, for this problem structure, the second effect on the lead time is the dominant one because of the relatively large run time of the Burn-In oven as compared to the setup times at other workstations in the Assembly Section. The "lot-for-lot strategy" (as practiced by the plant) takes into account only the first component of the lead time. It ignores the requirements of the Burn-In Section (and consequently, the waiting at Burn-In) while determining the order release. In contrast, the "batching strategy" incorporates the requirements of Burn-In and, therefore, addresses

both the components of lead time. While the determination of Input Lots from the solution of the Oven Loading Problem eliminates the wait due to the non-availability of bibs/program cards, the Minimum Oven Load criterion trades off waiting due to the non-availability of Burn-In ovens and waiting for enough units to be processed in the Assembly Section before an oven can be run. At the same time, the consolidation of loads while determining Input Lot sizes as well as the process of modification of the Input lot Sizes (e.g., in case of infeasibility) take into consideration the impact of setups on the Assembly Section.

We note that the consolidation of loads for each module type across all ovens, in every shift, leads to a reduction in waiting in the Assembly Section due to setups. If the loads of each oven (from the solution of the Oven Loading Problem) were to be released without consolidation, it could lead to larger number of setups and consequently longer lead times. This conjecture was indeed confirmed through simulation (using data for case 7 in Table 2). The minimum (total) lead time for the batching strategy (i.e., *with consolidation across ovens*) was determined as 2235.6 minutes. The lead time in the Assembly Section (i.e., the average time a module spends in the Assembly Section) was 1509.4 minutes. However, when the Order Release Schedule was determined *without consolidation*, the minimum (total) lead time was obtained as 2455.7 minutes (with a corresponding lead time in the Assembly Section of 1729.0 minutes). It should also be appreciated that if loads across many shifts were consolidated together, then, waiting due to large batch sizes may start to dominate.

II. Effect of Minimum Oven Load for Burn-In ovens: Recall that in the Synchronized Pull procedure, the Order Release Schedule was determined from the solution of the Oven Loading Problem which assumed that all modules were available for testing at the beginning of the week. However, in reality, the arrival times of the modules at the ovens will vary. Consequently, waiting for the whole batch to accumulate before initiating an oven run can lead to higher waiting times. On the other hand, if an oven is run at too small a load, then more oven runs will be required to test all the modules. This may increase the lead time and the cost of oven runs considerably. Thus choosing the best MOL for Burn-In ovens corresponds to the trade off between components (b) and (c) of the waiting time in the Burn-In Section, as discussed above. The advantage of using a MOL is borne out by our results which are discussed in the next section. Moreover, it simplifies the implementation of the loading process at the ovens.

III. Effect of Minimum Oven Load for EC Ovens : A similar discussion applies to the choice of MOL for EC ovens. However, the EC ovens differ from the Burn-In ovens in that there are no separate chambers in the EC ovens, all module types are compatible, and no bibs or program cards are required. The only restriction here is on the capacity of the oven. Therefore, no separate loading plan is required for them; the only relevant decision is the determination of MOL, which is chosen through simulation of different specifications of MOL for the EC ovens.

6 Computational Results and Conclusions

We describe below various results and show the extent of improvement in the management of this pacakging line over the practice that is currently followed by the plant.

The lead time plots for the two release policies (i.e., lot for lot and batching strategies) are shown in Figure 4. It shows the relationship between average lead time and the Minimum Oven Load at the Burn-In ovens for both the order release policies. The convex curve establishes the "best" Minimum Oven Load which will minimize the average lead time. It also shows that the batching strategy dominates the lot for lot strategy thereby improving upon the current practice at the plant.

Figure 5 emphasizes the choice of an appropriate MOL in order to reduce the waiting time in the Burn-In Section. It shows the two components of the total wait time in the Burn-In section: waiting for an oven to become available (or waiting outside) and waiting for enough modules to be produced by the Assembly Section so as to complete the specified load of the ovens (or waiting inside). As the MOL is increased, the wait outside continues to fall. This means that with larger MOL the availability of ovens increases as fewer oven runs are required to process a given demand. However, beyond a certain point, the wait inside starts to increase as the oven has to wait longer to accumulate enough modules in order to complete the specified load size, thus raising the total wait time. The minimum of total wait time at a MOL of 400 also corresponds to the minimum lead time as shown in Figure 4. This establishes the dominance of waiting time, at the Burn-In ovens, in determining the overall lead time.

Similar observations hold for the EC ovens. Figure 6 shows the plot of average lead time versus the MOL and establishes the best MOL of 800 modules for this demand profile. We also found

that the minimum total wait time at the EC ovens was also obtained at a MOL of 800.

We performed sensitivity analysis on the demand as well as the product mix to establish the robustness of the Synchronized Pull procedure. Our approach consistently performs better than the current practice at the plant for a range of demand scenarios. It can be seen from Table 2 that the extent of lead time reduction, obtained with the use of batching strategy over the lot-for-lot strategy, ranges from 0.7% to 40.9%, depending on the demand structure. (Note that the number of module types, demand, and bibs & program cards for each module type were generated from uniform distribution.) The choice of the best Minimum Oven Load depends not only on the total demand but also on the product mix. Figure 7 confirms this observation. In each case that is shown in Figure 7, the total demand was kept constant at 11,236 modules, whereas the number of different module types, demand for each type, as well as the number of bibs and program cards available for each module type were allowed to vary. The best MOL (as well as the lead time) clearly dpends on the product mix.

The discussion in the last section points towards the importance of considering various types of lot sizes when determining lead time, a key performance measure of shop floor effectiveness. However, in many industrial situations, managers become pre-occupied with performance measures such as capacity utilization and, often, neglect these factors. Often, the shop floor decision to increase the utilization of bottleneck workstations or costly operations could lead to poor synchronization in the shop floor and consequently long lead times. Figure 8 shows two kinds of utilization levels at the Burn-In ovens for various Minimum Oven Loads. **Time utilization** represents the percent of time ovens are busy processing ; **space utilization** represents the percent of total capacity of the ovens that is utilized (i.e., space utilization of ovens = (total number of parts processed/(number of oven runs * oven capacity)). As the MOL is increased, space utilization goes up but the percent of time the ovens are busy goes down as they have to wait longer for this load to be processed at the upstream stations. This is consistent with our earlier observations where the wait outside decreased as the MOL is increased (Figure 5). Figure 8 reveals some important implications:

a) A managerial decision to utilize as much capacity as possible will result in larger Minimum Oven Load than the optimal, and consequently a longer lead time.

b) A managerial decision to keep the ovens busy as much as possible will result in a smaller

Minimum Oven Load being used, than the optimal, thereby leading to a longer lead time.

c) The point where time and space utilizations are equal does not represent the optimal load size of the ovens thereby reflecting the inadequacy of utilization based measures in absence of their impact on lead times.

The above discussion also applies to the choice of the Minimum Oven Load for EC ovens.

In conclusion, we have presented in this paper an analysis of an actual semiconductor packaging line with an objective of reducing manufacturing lead time. The problem has a unique structure where the last workstation is a batch processor. We have developed a new variant of the bin packing problem and have shown the efficacy of the solution procedure. We have described the production process in detail, solved the decision problem of determining "good" lot sizes, and discussed the implications of various decisions. We have also addressed an important issue related to the frequency of oven runs and the minimum load for each oven run. Our solution procedure highlights the role of information on the state of the shop floor in making better decisions in complex manufacturing lines as well as the need for matching the requirements of the batch and discrete processors in designing release policies. This has largely been ignored by recent papers on burn-in operations. The procedure that we have developed synchronizes the Burn-In requirements with order release in the Assembly Section. Our analysis leads to decisions which are superior to current practices at the plant under study. The insights that we present here could be used to further improve the analytical models by focusing on critical parameters that affect performance.

It may be useful to delineate the contribution of batch processor as a bottleneck station (and otherwise) on lead time. While our study does not isolate this effect, one thing is certain that the constraints on the burn-in oven would lead to WIP even if it were not a bottleneck station; hence the need to manage them effectively. In addition, long processing times at the Burn-In necessitate that the upstream stations design their batches keeping the batch processor utilization in mind. At the same time, the fact that it is not best advised to have a very high capacity utilization at the Burn-In ovens underscores the fact that the uniqe characteristics of the batch processors have a dominant effect on lead times. However, a better understanding is required to isolate this phenomenon.

We also think that developing approximations on queueing times both in the assembly and

the burn-in sections, that are based on the solutions to the oven loading problem, could be a key direction towards developing analytical models for the overall lead time minimization problem. Since the the solution to the the oven loading problem yields optimal results for a wide range of demand & module types (as shown in Table 2), the effect of any error arising from the estimation of these parameters will be minimal. This also strengthens the use of the bin packing model as the core of any analytical model for the overall lead time minimization problem.

Another idea that we wish to explore further is the decomposition of a manufacturing line at bottlenecks. Most complex production lines exhibit multiple bottlenecks. By decomposing such a line into several sub-lines where each sub-line represents the case under study. Then the issue would be to design buffers at the beginning of each sub-line from which the bottleneck station can pull its requirements which can then be pushed through the stations before it. Perhaps CONWIP ideas of Spearman and Zazanis [1992] could be used to synchronize across the sub-lines.

References

Ahmadi, J. H., R. Z. Ahmadi, S. Dasu, and C. S. Tang, (1992) "Batching and Scheduling Jobs on Batch and Discrete Processors", *Operations Research*, 40, pp. 750-763.

Bhatnagar, R., P. Chandra, R. Loulou, J. Qiu, (1993) "Order Release and Coordination of Batches in a Complex Printed Circuit Board Manufacturing Environment", Faculty of Management, McGill University, Montreal, Canada.

Dobson, G., U. S. Karmarkar, and J. L. Rummel (1987) "Batching to Minimize Flow Times on One Machine", *Management Science*, 33, pp. 784-789.

Dobson, G. and R. S. Nambimadom (1992) "The Batch Loading and Scheduling Problem", *Working Paper No. QM 92-03*, Simon School of Business Administration, University of Rochester, Rochester, N.Y.

Fowler, J. W., G. L. Hogg, and D. T. Phillips (1992) "Control of Multiproduct Bulk Service Diffusion/Oxidation Processes", *IIE Transactions*, 24, pp. 84-96.

Glassey, C. R. and W. W. Weng (1991) "Dynamic Batching Heuristic for Simultaneous Processing", *IEEE Transactions on Semiconductor Manufacturing*, 4, pp. 77-82.

Hall, N., S. Ghosh, R.D. Kankey, S. Narsimhan, and W.T. Rhee (1988) "Bin Packing Problems in One Dimension: Heuristic Solutions and Confidence Intervals", *Computers & Operations Research*, 15, pp. 171-177.

Ikura, Y. and M. Gimple (1986) "Scheduling Algorithms for a Single Batch Processing Machine", *Operations Research Letters*, 5, pp. 61-65.

Johnson, D. S. (1973) "Near-Optimal Bin Packing Algorithms", Technical Report MAC TR109, Project MAC, Massachusetts Institute of Technology, Cambridge, Mass.

Johnson, D. S. (1974) "Fast Algorithms for Bin-Packing", *Journal of Computing System Science*, 8, pp. 272-314.

Karmarkar, U. S., S. Kekre, S. Kekre, and S.Freeman (1985) "Lot Sizing and Lead Time Performance in a Manufacturing Cell", *Interfaces*, 15, pp. 1-9.

Karmarkar, U. S. (1987a) "Lot Sizes, Lead Times, and In-process Inventories", *Management Science*, 33, pp. 409-418.

Karmarkar, U. S. (1987b) "Lot Sizing and Sequencing Delays", *Management Science*, 33, pp. 419-423.

Karp, R. M. (1972) " Reducibility Among Combinatorial Problems", in *Complexity of Computer Computations* (eds. R.E Miller and J.W. Thatcher), Plenum Press, New York, pp. 85-104.

Leung, J. Y. T. (1989) "Bin Packing with Restricted Piece Sizes", *Information Processing Letters*, 31, pp. 145-149.

Lee, C-Y., R. Uzsoy, and L. A. Martin-Vega, (1992)" Efficient Algorithms for Scheduling Semiconductor Burn-In Operations", *Operations Research*, 40, pp. 764-775.

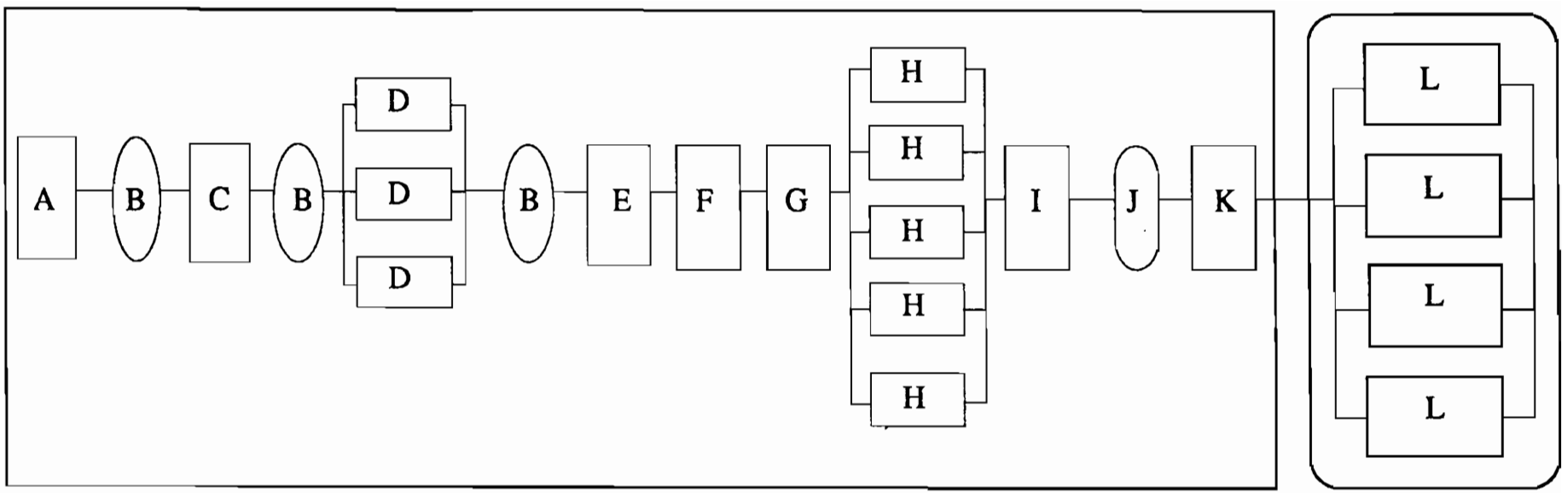
Lin, L. and J.K. Cochran (1987) "Optimization of a Complex Flow Line for Printed Circuit Board Fabrication by Computer Simulation", *Journal of Manufacturing Systems*, 6, pp. 47-57.

McDowell E. D. and S.U. Randhawa (1989) "A Simulation-based Production Planning Support System for Printed Circuit Board Fabrication", *Journal of Manufacturing Systems*, 8, pp. 225-234.

Optimization Subroutine Library (OSL), Guide and Reference, Release 2, (1991), SC23-0519-02, 3rd. ed., IBM Corporation, Kingston, N.Y.

Raman, P. (1989) "Average-Case Analysis of the Smart Next Fit Algorithm", *Information Processing Letters*, 31, pp. 221-225.

Spearman, M.L. and M. A. Zazanis (1992), "Push and Pull Production Systems: Issues and Comparisons", *Operations Research*, 40, pp. 521-532.



ASSEMBLY SECTION

BURN-IN SECTION

A: RACKING/PRECLEANING

B: DEGREASING

C: CLEANING

D: CHIP PLACEMENT

E: REFLUX/REFIRING

F: LASER CAPPING

G: EPOXY DISPENSING

H: EPOXY CURING

I: WAVE SOLDERING

J: BUBBLE LEAK TESTING

K: PIN STRAIGHTENER

L: BURN-IN OVENS

IGURE 1: SEMICONDUCTOR PACKAGING LINE

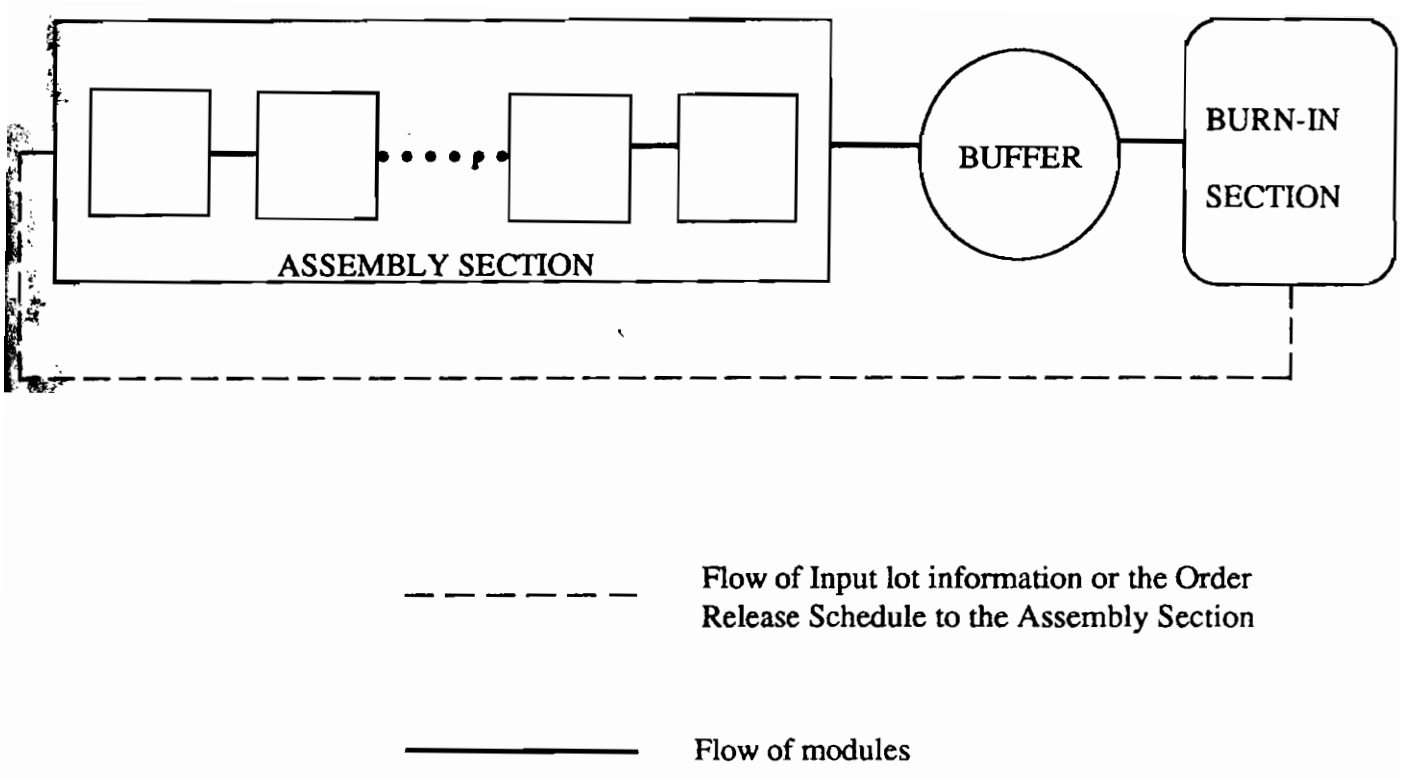


FIGURE 2: BURN-IN PULLS ITS REQUIREMENTS FROM ASSEMBLY

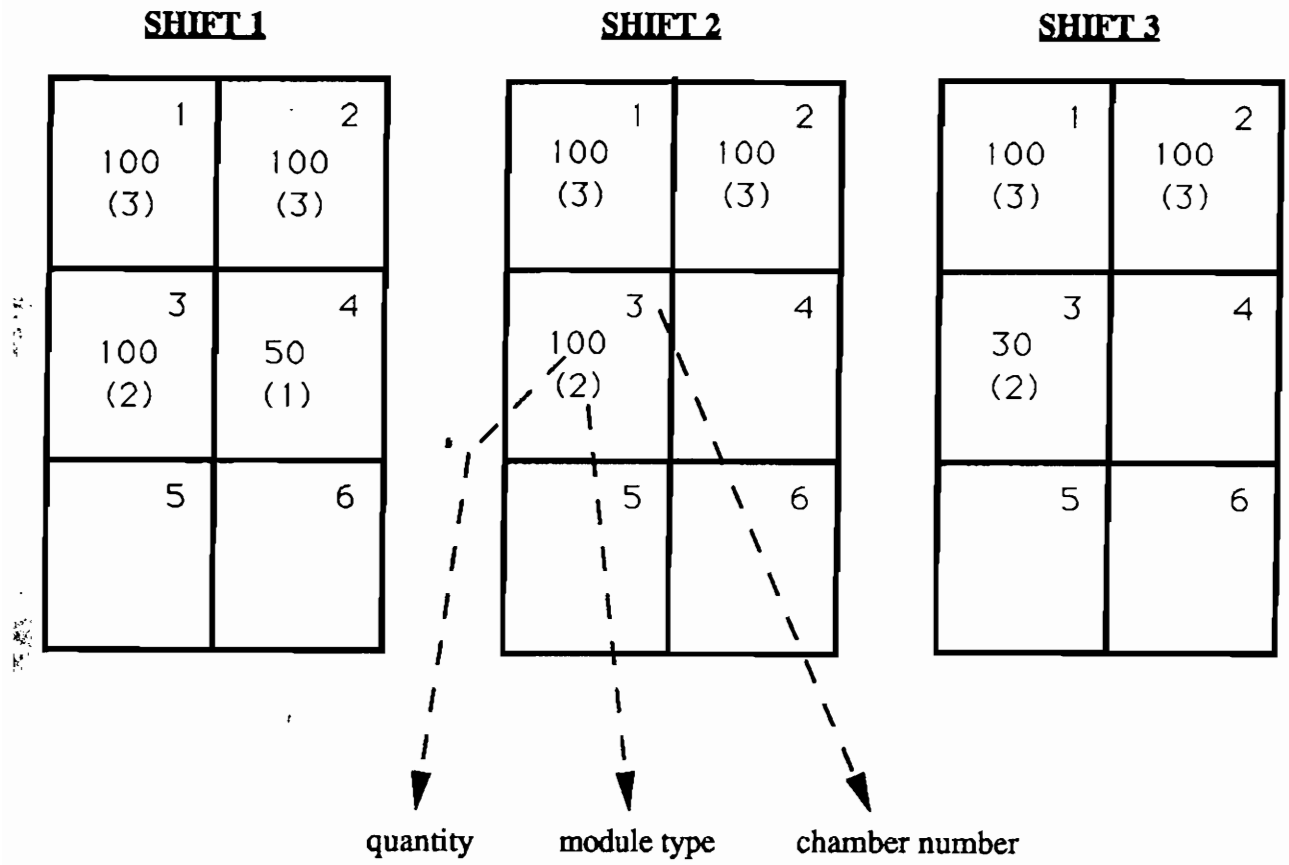


FIGURE 3: SOLUTION TO THE OVEN LOADING PROBLEM FOR THE EXAMPLE

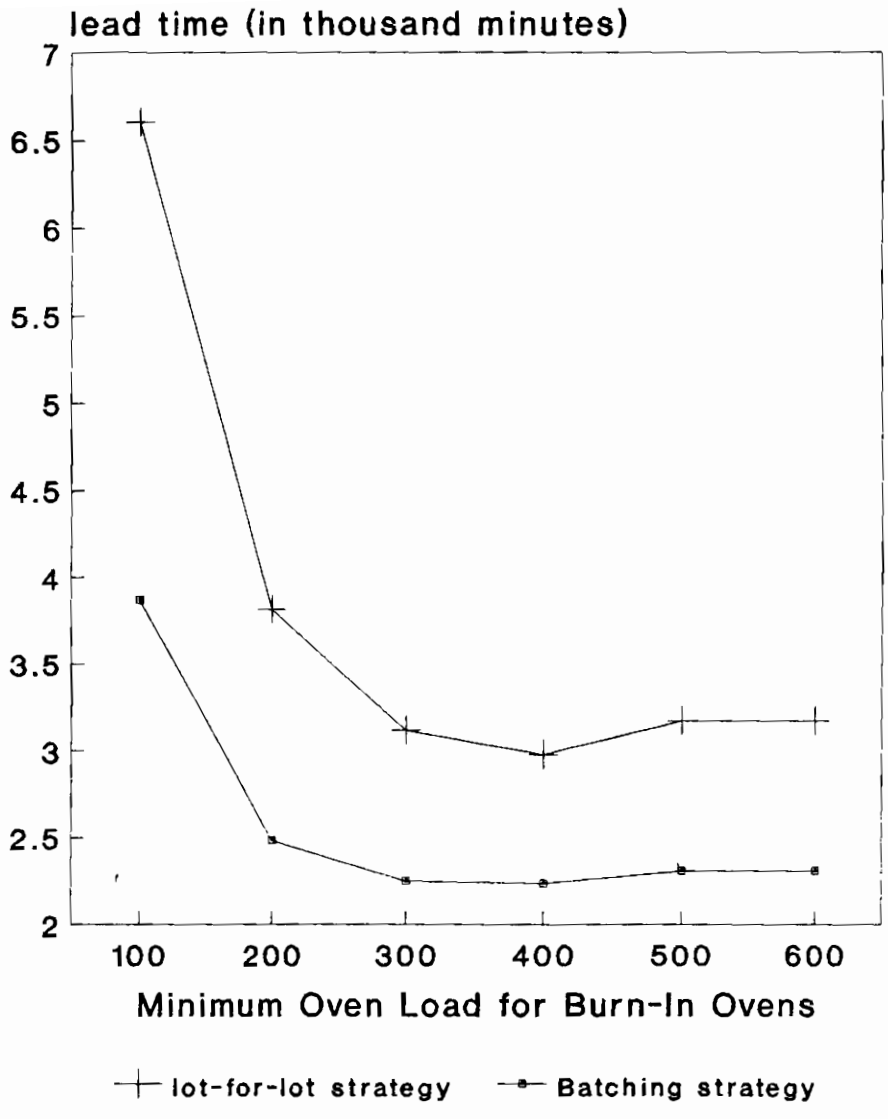


FIGURE 4: LEAD TIME PLOT FOR DIFFERENT ORDER RELEASE STRATEGIES

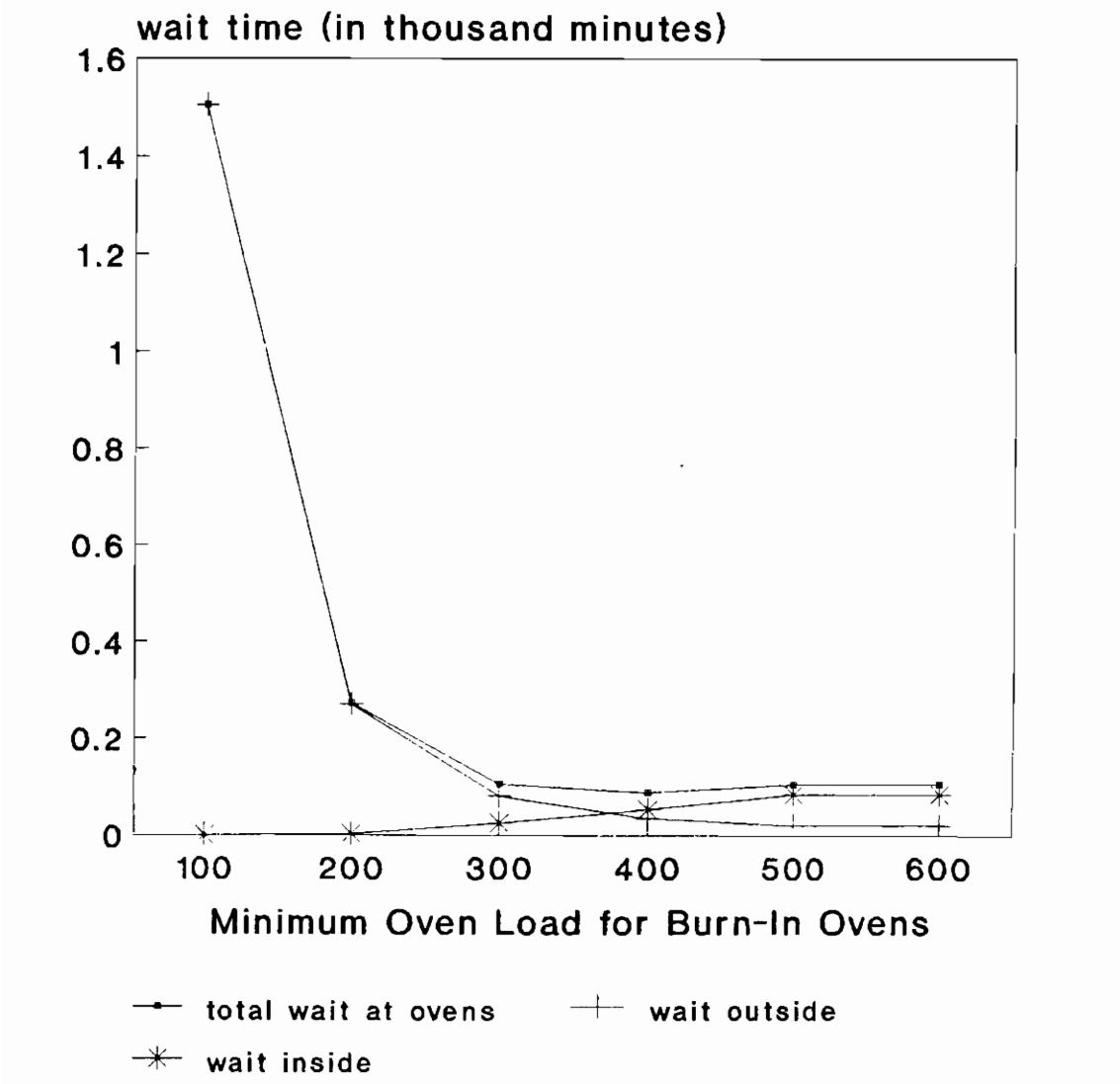


FIGURE 5: WAIT TIME PROFILE AT THE BURN-IN OVENS FOR THE BATCHING STRATEGY

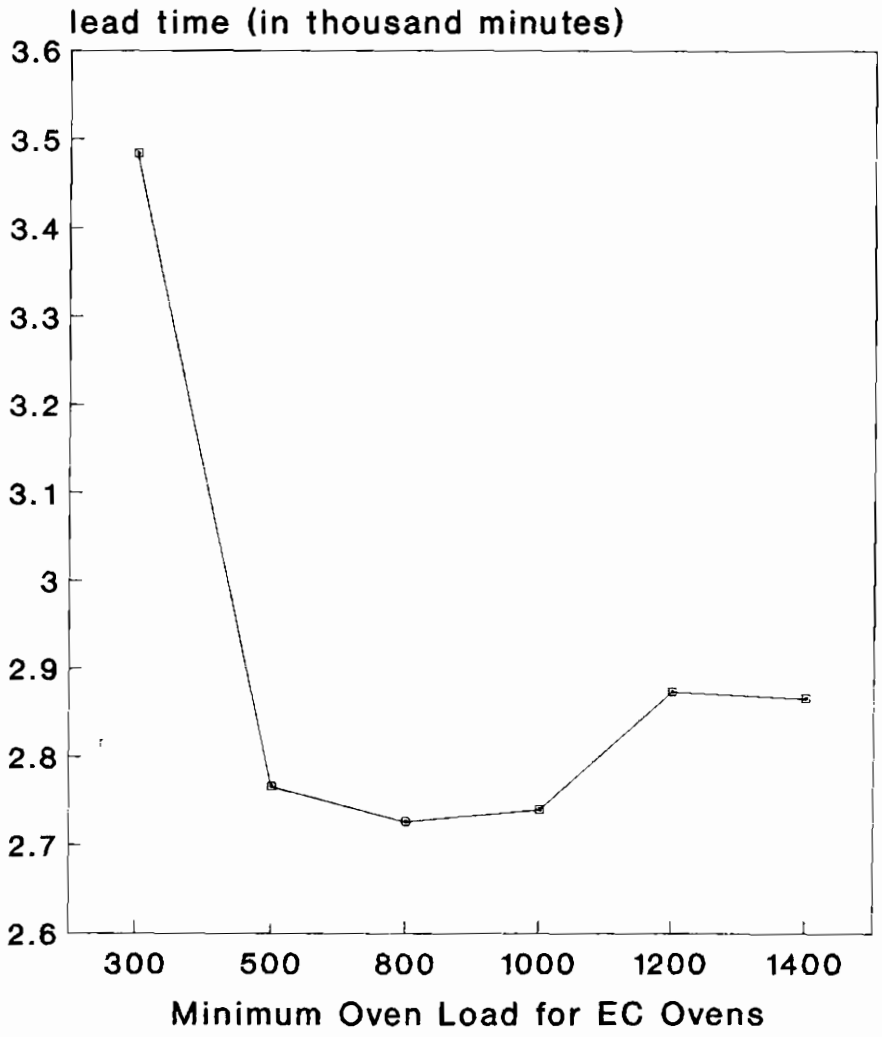


FIGURE 6: LEAD TIME PLOT FOR DIFFERENT MINIMUM OVEN LOADS FOR EC OVENS

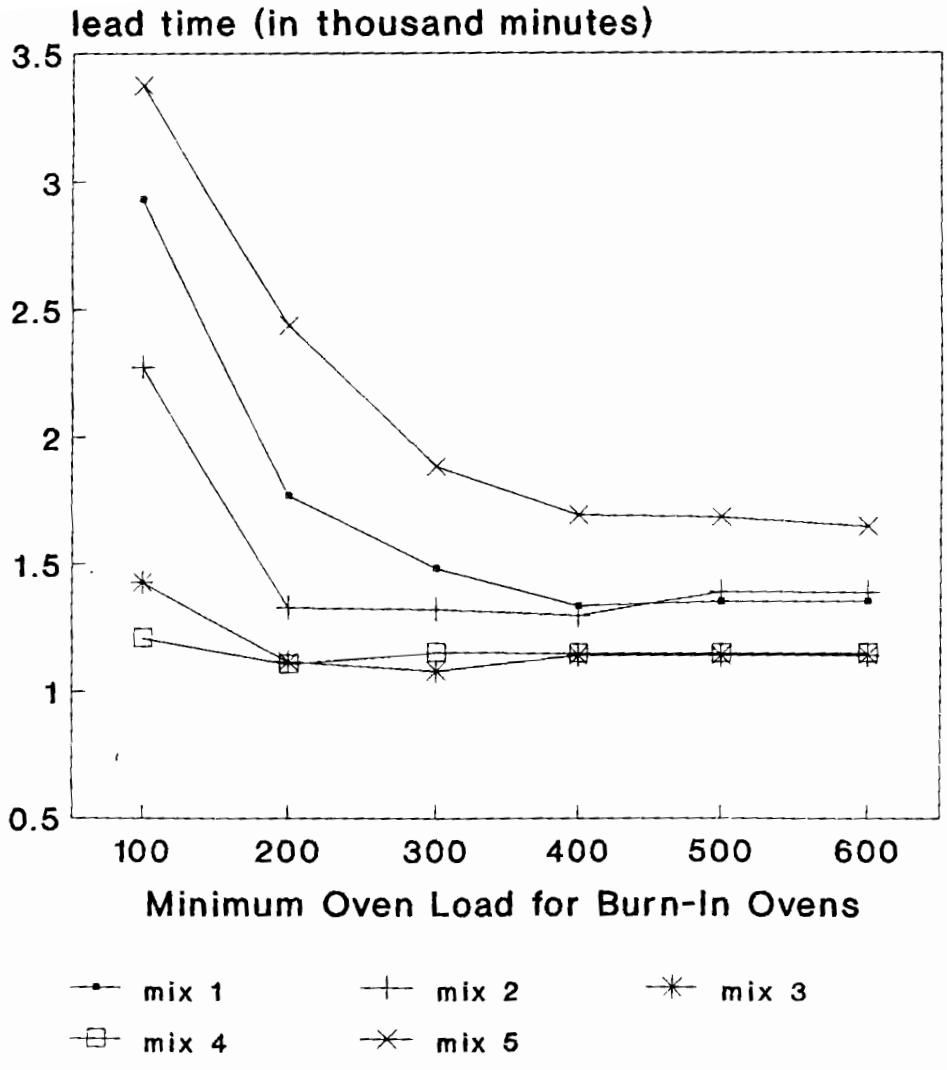


FIGURE 7: LEAD TIME PLOT FOR VARYING PRODUCT MIX