WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

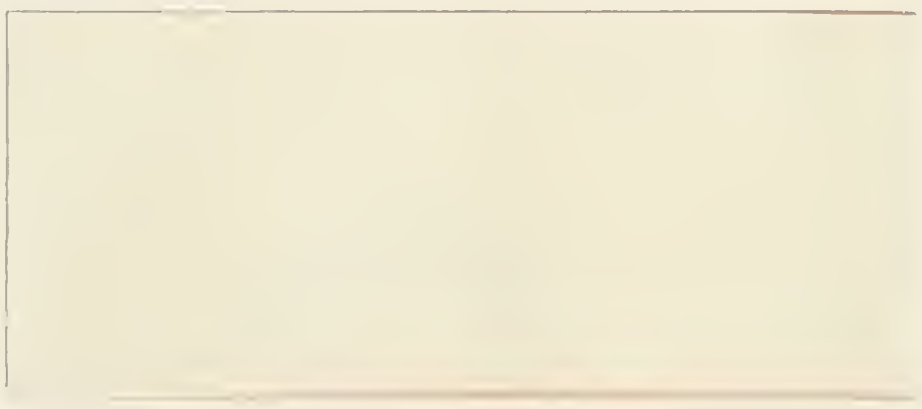
PLANNING AND SCHEDULING FOR EPITAXIAL WAFER PRODUCTION FACILITIES

by
Gabriel R. Bitran*
and
Devanath Tirupati*

#1744-86

February 1986
Revised April 1986

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139



PLANNING AND SCHEDULING FOR EPITAXIAL WAFER PRODUCTION FACILITIES

by

Gabriel R. Bitran*
and
Devanath Tirupati*

WP #1744-86

February 1986
Revised April 1986

* Massachusetts Institute of Technology

PLANNING AND SCHEDULING FOR EPITAXIAL WAFER PRODUCTION FACILITIES

Gabriel R. Bitran and Devanath Tirupati

Massachusetts Institute of Technology, Cambridge, Mass.

ABSTRACT

In this paper we describe production planning and scheduling models for a semiconductor company manufacturing specialty wafers. The production of specialty wafers involves processing a large number of low volume customer specific orders which can be classified into a number of product groups.

Typically, the literature in scheduling deals with one product and a single objective. In contrast, we propose and test several heuristics for scheduling jobs in a multi-product parallel reactor (machine) shop with different criteria. We introduce a set of indices to measure the degree of homogeneity in the product set. Based on the results of the computational experiments, we recommend that the choice of the heuristic should be guided by the homogeneity of the product set and the chosen objective criterion.

The planning problem relates to the assignment of reactors to the product groups to obtain homogeneous product sets and can be viewed as an effort to define smaller, independent shops. The significance of the planning exercise is due to the fact that a homogeneous product set enables the use of a simpler heuristic and reduces the complexity of the scheduling system. The planning model is a nonlinear integer program and difficult to solve. We propose to obtain an approximate solution by solving a related quadratic program. We provide bounds on the performance of this approximate solution and demonstrate that it is asymptotically optimal.

1.0 INTRODUCTION

In this paper we describe a scheduling system developed for a semiconductor wafer manufacturing company. The facility produces specialty wafers which are a major input in the manufacture of a variety of semiconductor devices. The manufacture of specialty wafers is characterized by relatively small volumes, small lot sizes, and a wide range of products sharing common equipment. The production of epitaxial wafers involves a number of steps which can be classified into three stages--crystal growth, substrate fabrication and epitaxial growth. In the first stage (crystal growth) ingots are produced from electronic grade silicon. The ingots are converted into substrates in the second stage by a series of machining and chemical operations. Epitaxial growth is a batch operation in which a thin silicon dopant layer is grown on the wafer surface. This is a critical operation and invariably it is necessary to process each order (job) separately. The epitaxial growth is achieved by vapor deposition process in a reactor under specified conditions.

A detailed analysis of the capacity at various stages revealed that the epitaxial growth in the reactors was a bottleneck operation. This finding was consistent with the management perception and observations on the shop floor. This paper describes the system developed for scheduling jobs at this stage. It should be noted that the facility is equipped with a number of reactors with varying capability. This results in restrictions on the type of jobs that can be processed on each reactor. In addition, since the number of jobs is large, scheduling at this stage is a complex problem.

We model the scheduling problem by a single-stage parallel-machines facility. Traditional approaches in scheduling typically consider a homogeneous job stream and a single objective. The variation in resource requirements of different products suggests that this may not be quite appropriate in the present application. We consider a facility with different types of products and propose several

heuristics for two criteria: make-span and tardiness. Computational experiments were performed with realistic data to test the heuristics. In addition, we examined the performance of these algorithms with respect to the number of tardy jobs. The results suggest that the two phase hierarchical procedure (referred to as H5 in the paper) proposed performs reasonably well under a variety of conditions for all the three objectives. An exception to this is the case when the job due dates are dependent on the product groups and are tight. In this instance we recommend the use of two heuristics depending on the objective. Further, we classify the prod-uct mix as "homogeneous" or "heterogeneous" and suggest measures to do so. The experimental results suggest that while a simpler heuristic is adequate for homo-geneous products, a more elaborate algorithm is required for the heterogeneous case.

We also consider the planning problem of obtaining a homogeneous set of products by appropriately assigning reactors to products. The motivation for this effort is the fact that a homogeneous product set would enable use of a simpler heuristic and reduce the complexity of the scheduling system. We formulate a non-linear integer programming model, which can be interpreted as an attempt to obtain smaller independent shops with homogeneous product sets. Since this is difficult to solve, we propose a simpler problem to obtain an approximate solution. We provide bounds on the performance of this approximation and demonstrate that it is asymptotically optimal with the number of products. We also show that the approximate solution yields assignments that are asymptotically optimal for the make-span.

The plan of the paper is as follows. In the next section we describe the scheduling model and the heuristics. The results of the computational experiments are discussed in Section 3. In Section 4, we describe the planning models for assigning reactors to products. The last section provides a brief summary of the conclusions of this study.

2.0 THE SCHEDULING PROBLEM

We model the scheduling problem at the wafer manufacturing facility in the following terms.

1) Each job requires one operation.

2) Jobs are independent--i.e. there are no precedence constraints.

3) Process times are deterministic and sequence-independent.

4) All the jobs in the facility are available initially for processing.

5) The facility consists of multiple machines operating in parallel.

6) The machines are neither identical nor uniform. (The machines are said to be uniform when they can process all jobs with possibly different speeds). Each job can be processed on a subset of machines depending upon the product characteristics. The process time, however, is identical on the reactors which are capable of processing the job. Thus, we can classify jobs into product groups. With each group j we can associate a subset of reactors $I(j)$ which is capable of processing the jobs in that group.

7) The availability of the machines is deterministic but not identical for all the reactors. This follows from the differences in the operating schedules of the machines.

The brief review of the literature which follows suggests that the above characteristics lead to a difficult problem for which no efficient algorithms are available to obtain optimal solutions. As we describe subsequently, the model we consider does incorporate some features that are important from a practical perspective.

2.1 REVIEW OF THE LITERATURE

The single machine scheduling problem is perhaps the simplest of the sequencing problems and is discussed in detail in Baker (1974) and Conway et al. (1967). It is shown that for the static case with all jobs available initially, sequencing

according to the shortest processing time minimizes the average flow time. Likewise, sequencing in order of due dates minimizes the maximum lateness. While simple rules are available for some objectives, no efficient algorithm exists for minimizing the average tardiness. In fact this is still an open problem since it has not been shown to be NP-complete. Optimal solutions can be obtained by using dynamic programming or branch and bound techniques and a number of heuristics have been proposed for generating "good" schedules.

The complexity of sequencing with parallel machines depends on the machine characteristics and whether preemption of jobs is permitted. The problem of minimizing total tardiness with multiple machines and no preemption is NP-hard (Lawler et al., 1982) and hence there is little hope of developing an efficient algorithm for this criterion. Minimizing the make-span and average flow time are the criteria that have been considered in detail and reported in the literature for the multiple machine problem. Make-span refers to the time required to complete all the jobs and is of interest because it is often related to the work-in-process. It has been shown that a simple ordering by processing times minimizes the average flow time if the machines are identical. The make-span problem has been shown to be NP-complete with two or more machines and no preemptions. A number of heuristics have been proposed--for example list processing techniques such as largest processing time, the differencing method by Karmarkar and Karp (1982), and a duality-based method by Hochbaum and Shmoys (1984). All these algorithms use intuitively simple notions and have been shown to be asymptotically optimal with the number of jobs.

The problem complexity is considerably reduced by permitting preemptions. Polynomial-time algorithms have been proposed for the make-span problem with uniform machines. Federgruen and Groenevelt [8] developed an algorithm using ordinary network flow techniques. Labetoulle et al. (1984) consider the problem with arbitrary release dates and provide two algorithms--one for minimizing the make-

span with an arbitrary number of machines and the other for minimizing maximum lateness with two uniform machines.

Most of the reported literature deals with the scheduling problems with either identical or uniform machines and very few results are available for the general case. For the non-uniform machine case Horn (1973) formulated an assignment problem to minimize the make-span with no preemptions and general processing times. More recently Davis and Jaffe (1981) proposed a heuristic based on list processing techniques.

2.2 THE SCHEDULING PROBLEM AT THE EPITAXIAL GROWTH SECTION

The above discussion makes it clear that scheduling jobs with parallel machines is a difficult problem with either the make-span or the tardiness objective. The problem outlined at the beginning of the section corresponds to a quite general case of parallel processing and hence it is difficult to solve. Some practical issues not explicitly considered in the initial description of the problem are briefly discussed below.

i) The model considers a static job shop, while in practice the shop is dynamic and jobs are released as and when orders are received. We address this problem by periodically solving static problems comprising the jobs on hand. Since new jobs are typically assigned later due dates we expect the procedure to yield schedules which are stable in the short run. Our experience with the scheduling system that has been implemented at the wafer manufacturing facility confirmed this conjecture.

ii) Deterministic approximation of stochastic process times is quite reasonable in the present application. The process time includes a component for test run which is stochastic. The time required for the production run which follows the test run is nearly deterministic. In addition, the jobs are relatively large in the sense that test runs constitute a minor part of the total process time.

Operational data from the shop floor supports this reasoning and the variability in process times is not significant.

While the research in scheduling is typically concerned with one objective, in practice, managers are often faced with multiple and at times conflicting objectives. Minimizing the make-span is an important criterion, since this is often related to the work-in-process. In addition, jobs have assigned due dates and minimizing the total tardiness and/or the number of late jobs is also a critical objective. It is also well known that these objectives are, in general, not equivalent and an optimal solution for one is not necessarily optimal for other objectives. Thus, we have two different scheduling problems with conflicting objectives--one with the make-span as an objective and the second with total tardiness as the criterion. We now describe some heuristics to address these problems. In the following section we report on the computational experiments to test the performance of these heuristics.

2.3 PROBLEM 1 - TARDINESS CRITERION

We first consider the case where the objective is to minimize the total tardiness. The earlier discussion on the literature should make it clear that we are unlikely to find efficient algorithms to obtain optimal solutions and the best we can hope for are good heuristics. We propose to examine four different heuristics. The algorithms are similar and schedules are derived in two phases. In the first phase jobs are considered at an aggregate level and priorities are assigned to the product groups. In the second phase, based on these priorities, reactors are assigned to the individual jobs. For jobs within a product group the priority is given by its due date. The assignment of jobs to reactors is obtained by a single-pass heuristic. The four heuristics essentially differ in the manner in which the priorities are defined. Finally jobs assigned to a reactor are scheduled according to the due dates. The priority rule for reactor assignment for the

four heuristics is as follows (note that a lower index represents a higher priority):

i) Each product group is assigned a priority based on the number of reactors on which jobs in that group can be processed, i.e. the cardinality of the set I(j). The priority for jobs within each group is determined by due date and in case of ties by the process times.

ii) The priority for product groups is determined by the ratio of total work load for jobs in the group to the number of reactors on which these jobs can be processed. The priority of jobs within each group is similar to that in case i).

iii) The priority is defined by the quotient of two factors. The first factor relates to the tightness of job due dates and is defined by the average slack for jobs in the group. The second factor is the average load per reactor. The priority is defined as follows.

$P(j)$ = priority for product group j = $Sl(j) \cdot |I(j)| \; / \; d_j$

where: $Sl(j)$ = average slack per job = $\sum_{k=1}^{g_j} (dd(k) - p(k))/g_j$

$|I(j)|$ = number of reactors available for group j

g_j = number of jobs in group j

d_j = total work load for jobs in group j

$dd(k)$ = due date of job k

$p(k)$ = process time for job k.

Again, for jobs within each group the priority is similar to the previous cases.

iv) The priority is determined by the due date of the job and in case of ties by the process time. In this heuristic we do not classify the jobs into product groups.

Heuristic iv) is similar to the one proposed by Bernardo and Lin (1984) adapted to the present situation and is used for comparison. The authors have tested their heuristic on small (up to 14 jobs and 6 machines) and large problems (up to

80 jobs and 20 machines) and reported encouraging results. In the present application we have up to 300 jobs. The heuristics are described in detail below.

Notation:

m = number of reactors/machines

n = number of jobs

g = number of product groups

I(j) = set of reactors capable of processing jobs in group j, j = 1, ..., g

g(j) = number of jobs in group j

dd(k) = due date of job k

p(k) = process time for job k

P(k) = priority measure for job k

GJ(k) = product group of job k

S(k) = start time of job k

F(k) = completion time of job k

n(i) = number of jobs assigned to reactor i

a(k,i) = job that occupies position k on reactor i

Z = measure of tardiness (objective).

Without loss of generality we can assume that product groups are ordered according to the priority, i.e., group 1 has the highest priority and group g the lowest. Note that the priority of a product group depends on the rules (i - iv) used and may be different for the four heuristics.

ALGORITHM

Step 1 Initialization

$Z \leftarrow 0$.

Set n(i) = 0, i = 1 to m.

Compute product group priority. Then within each group compute job priority. For jobs within a product group the priority is given by the due date and in case of ties by the process time. Any remaining ties are resolved randomly. We assume that the jobs are arranged in accordance with this priority.

Step 2 Reactor Assignment

For $k = 1$ to n:

a) compute $\Delta(k,i)$ = increase in the total tardiness if job k is assigned to reactor i and scheduled according to the earliest due date (e.d.d.) rule, $i \in I(GJ(k))$.

Note that the e.d.d. rule implies that jobs assigned to a reactor are processed according to the due date. Hence $\Delta(k,i)$ is computed by first determining the position in which the new job k would be inserted on reactor i by following the e.d.d. rule. The increase in total tardiness is then computed by examining the changes in the completion times of the succeeding jobs on that reactor.

b) $i^* = \arg\min_{i \in I(GJ(k))} \Delta(k,i)$, $\Delta = \Delta(k,i^*)$.

c) Assign job k to reactor i^*.

Update the queue at reactor i^*; $n(i^*) \leftarrow n(i^*) + 1$.

Compute start and finish times for job k and the jobs on i^* which are processed after k.

$Z \leftarrow Z + \Delta$.

Step 3 Adjusting the Schedule

The schedule is adjusted to delay the start of jobs as much as possible without increasing the tardiness. The sequence of jobs on a reactor remains unchanged.

```
For  i = 1  to  m:
   For  k = n(i)  to  1  step -1
      If  F(a(k,i)) < dd(a(k,i))
      then  {F(a(k,i)) ← dd(a(k,i))
             S(a(k,i)) ← F(a(k,i)) - p(a(k,i))}
      else next i
   Next  k
Next  i
```

Note that while the algorithm described above schedules all the jobs, it can also be used to add new jobs. In this case the initialization step is modified. The queues are not empty but reflect the current schedule. The priority index and related computations in Step 2 are required for new jobs only. This is a very useful feature since the plans are updated on a rolling horizon basis.

Step 3 of the procedure is relevant if some of the jobs assigned to a reactor can be completed within the due date. In such cases the schedule is adjusted so that the jobs are completed as late as possible without making any job overdue. This feature has been incorporated because early completion imposes a small penalty. This is considered to be a secondary objective.

2.4 PROBLEM 2 - MAKE-SPAN CRITERION

We propose a two phase hierarchical approach to develop a schedule to minimize the make-span. In the first phase we consider jobs in aggregate at the group level and determine the allocation of load among the various reactors. The criterion at this stage is to minimize the make-span. In the second phase jobs are assigned to reactors and a schedule is developed consistent with the loading pattern derived in the first phase. The objective in the second phase is total tardiness. In fact, in the first phase we consider the additional objective of achieving a balanced load between the reactors. This is subsequently described in this section. We derive a schedule which considers three objectives, albeit in a hierarchical fashion. The problem of allocating the load among the reactors is formulated as a convex program described below. Observe that at this level we are ignoring the due dates of individual jobs. However, we expect that this procedure will give reasonable results for tardiness-related criteria as well, so long as due date assignment is not based on the product group. The results of computational experiments described in the next section confirm this conjecture.

We define the following additional notation:

d_j = total load of jobs in product group j (load refers to the machine or process time required to complete the jobs in the group)

$av(i)$ = availability of reactor i (for example hrs/week)

$J(i)$ = set of products that can be processed on reactor i.

Variables:

$x(i,j)$: load of product group j assigned to reactor i

MS : make-span

r_i : total load assigned to reactor i.

$$[AP] \qquad Z_{AP} = \min \sum_{i=1}^{m} (MS - r_i/av(i))^2$$

$$\text{s.t.} \qquad r_i = \sum_{j \in J(i)} x(i,j), \qquad i = 1, 2, \ldots, m;$$

$$r_i/av(i) \leq MS, \qquad i = 1, 2, \ldots, m;$$

$$\sum_{i \in I(j)} x(i,j) = d_j, \qquad j = 1, 2, \ldots, g;$$

$$x(i,j) \geq 0, \qquad i \in I(j),\ j = 1, 2, \ldots, g.$$

In this formulation we are assuming that the jobs are completely divisible and can be processed in parallel. This is not an unduly restrictive assumption in the present instance. Long jobs can and sometimes are processed in parallel on different reactors. However, since it is more common to find a large number of small jobs, the indivisibility of jobs has a marginal impact on the total make-span. This follows from the fact that if MS* is the optimal MS in problem AP and p_{max} is the process time of the longest job, the optimal make-span without divisibilities in jobs is less than $MS^* + p_{max}/\min_i (av(i))$.

The choice of objective function needs some elaboration. The problem of minimizing the make-span can be formulated with a linear objective function (Min MS) as a modified assignment problem. This is unlikely to be very useful because of

the large number of alternative optimal solutions. We are in fact interested in achieving minimum make-span and also maintaining a balanced load on various reactors. Each term in the objective function can be interpreted as the idle time on each reactor. Replacing the quadratic term by a linear one would result in minimzing the idle time and the makespan but would not discriminate between the loads on different reactors. It may be noted that the processing requirements of the various product groups could lead to imbalance in reactor loads. The minimum make-span is determined by the load on the critical reactor. A wide range of feasible loading patterns on the other reactors gives rise to alternate optima with respect to the make-span objective. The use of the quadratic objective function ensures that the optimal solution will favor a uniform loading of the reactors. The results provided in Appendix 1 demonstrate that the optimal solution to the problem will indeed result in a minimum make-span solution.

The convex program is relatively small since, in practice, the number of product groups and reactors tends to be of the order of 25 and 10, respectively. The problem can be solved using standard techniques (Bradley et al. (1977), Schrage (1984), Shapiro (1979)). In the computational experiments we used the LINDO package to solve the quadratic program.

The detailed scheduling consisting of job assignments to reactors can be done using the heuristic (iv) described earlier, subject to the aggregate allocations given by AP. Although such allocation may require job splitting, in implementing the heuristic we have maintained integrality of jobs in a reactor. We have also modified Step 2 of the algorithm as follows: The set of admissible reactors at any stage excludes those which have already been assigned jobs up to the limit specified in the primary allocation.

3.0 COMPUTATIONAL EXPERIMENTS

In this section we describe the computational experiments performed to examine

the behavior of the heuristics discussed earlier. The data base for generating the test problems was developed from operational data obtained from a specialty wafer manufacturing corporation and is representative of realistic situations in that industry. The corporation has two shops which operate independently and produce a range of semiconductor wafers. The two shops provided data bases for the two sets of experiments reported in this study. All the test problems in the experiments consisted of 300 jobs and the following criteria were used to compare the results:

i) Make-span or the minimum time required to complete all the jobs;

ii) Total tardiness;

iii) Number of jobs tardy.

The maximum number of jobs in each shop at any point in time is of the order of 300. We observe that these problems are significantly larger (in terms of number of jobs) than those reported in the literature in testing related scheduling algorithms. For example Baker and Martin (1974) considered test problems with a maximum of 15 jobs while Bernardo and Lin (1984) reported results with 80 jobs and 20 machine problems. It should be mentioned that Baker and Martin solved the problems to optimality for comparison purposes and this is nearly impossible in large problems. However, we do obtain a lower bound for the make-span criterion and the bound is reasonably close.

3.1 DATA BASES FOR THE EXPERIMENTS

1) Number of machines: 9 and 12, respectively, for the two data sets.

2) Product Groups: The jobs processed by the facility were classified into homogeneous groups from a processing perspective. The numbers of groups are 15 and 18 for the two data sets. For each product group a subset of reactors, capable of processing jobs in that group, is assigned. The distribution of jobs into the product groups was determined empirically from a sample of orders processed by

the company. While the test problems were run with the total number of jobs fixed at 300, the product groups were assigned in a random manner consistent with the observed distribution.

3) Process (Operation) Time: Computational experiments reported in the literature have typically used the normal distribution to generate process time data. For example see Bernardo and Lin (1984) and Baker and Martin (1974). The data from our sample suggested that this distribution is inappropriate in the present instance. Operations in the wafer manufacturing facility were characterized by a large proportion of small jobs with few orders requiring large process times. This feature is characteristic of both shops and is analogous to the A-B-C classification in inventory systems. It was also observed that the distribution of process times was very similar for jobs in different groups. Therefore, we decided to use the same form of distribution for all the groups in both the data sets. The parameters defining the distribution were different for each group and were estimated from the sample data. We were unable to fit any standard distribution to this data and hence used the empirical distribution obtained from the sample data itself. The ratio of process time (p) to its mean ($\bar{p}$) defines a non-dimensional measure of process time ($\hat{p}$) which was used to describe the distribution. The relation between process time and the number of jobs is described by the following:

$$\log(\hat{p}) = a_1 + b_1 \log(x), \qquad 0 \leq x \leq x_0$$

$$= a_2 + b_2 \log(x), \qquad x_0 < x \leq 1$$

where $\hat{p} = p/\bar{p}$

$\bar{p}$ = mean process time

x = cumulative proportion of number of jobs

and x_0, a_1, b_1, a_2, b_2 are parameters determined from the sample data.

Since the distribution of cumulative number of jobs is uniform in the interval [0,1], the process times for test problems were generated using a uniformly

tion. The corresponding process time is obtained as $p = \hat{p} \cdot \bar{p}$ where $\bar{p}$ is the mean process time for the group.

4) Due Dates: To generate due dates for the jobs in the test problems we adopted a procedure similar to that followed in the literature (Baker (1974), Baker and Martin (1974), Bernardo and Lin (1984), and Conway et al. (1967)). In assigning due dates, three factors were considered: tardiness factor (t), due date range (r), and dependence/independence between process times and due dates. The due date for job k was selected randomly from the interval $\bar{\bar{dd}}_k \pm w/2$, where $\bar{\bar{dd}}_k$ is the mean due date for job k and w is the due date range. In case of dependent due dates, the mean due date for each job is proportional to its process time while with independent due dates the mean due date is the same for all jobs. In both cases the mean is a function of the tardiness factor t. Likewise, the range of the due date distribution is the same for all jobs and is defined by the range factor r. The determination of these parameters is described in detail below.

a) Tardiness factor (t) is a rough measure of due date tightness which together with the dependence factor determines the mean due date of a job.

$$\bar{\bar{dd}}_k = n(1 - t) / mp_k \qquad (1)$$

$$\bar{\bar{dd}}_k = n(1 - t) / m\bar{p} \qquad (2)$$

where $\bar{\bar{dd}}_k$ = mean due date for job k

m = number of machines (reactors)

n = number of jobs

t = tardiness factor, $0 \leq t \leq 1$

p_k = process time for job k

$\bar{p}$ = mean process time for the job set = $(\sum_{k=1}^{n} p_k)/n$.

b) Dependence factor: "Dependent due dates" refer to the case in which due dates are dependent on process times. We use a procedure similar to that adopted

by Baker and Martin (1974) and the mean due date is determined by (1). When the due dates are independent of the process times, the mean due date is given by (2).

c) The width of the due date distribution is defined by the range factor r by the formula:

$$w = r \sum_{k=1}^{n} p_k / m$$

where w is the width of the due date distribution.

3.2 EXPERIMENTAL RESULTS

In designing the computational experiments, we considered two levels for each of the three factors. This is consistent with the practice followed in the literature (Baker and Martin (1974), Bernardo and Lin (1984)). The factor levels are such that the due dates are generated with varying degrees of tightness. The experiments were performed for the following factor levels.

Tardiness (tightness) factor t = 0.6 and 0.2

Due date range r = 0.2 and 0.95

Dependent and independent due dates.

The heuristics were tested with eight sets of parameters defined by the three factors. For each set, ten replications were performed totalling 80 problems for each data set. As described in the previous section the five heuristics tested are

i) (H1) Priority of group j is determined by the cardinality of the set I(j). In case of ties, the group with higher load will have a higher priority.

ii) (H2) Priority is given by $|I(j)| / d_j$

iii) (H3) Priority is given by $Sl(j) \cdot |I(j)| / g(j)$

iv) (H4) No priority among product groups.

v) (H5) The two-phase heuristic for the make-span objective proposed in the

previous section. In the first phase of this method the work load of the product groups is allocated to the reactors by solving the quadratic programming problem described earlier.

We propose a reactor-based index for each data base to check for homogeneity in the products. This index measures the average load on each reactor and is defined as follows:

$$RG(i) = \sum_{j \in J(i)} N(j)\, \bar{P}(j) \;/\; |I(j)|$$

where $N(j)$ = number of jobs in group j

$\bar{P}(j)$ = mean process time for jobs in group j

and $I(j)$, $J(i)$ are as defined earlier.

Each term in the summation represents the average load per reactor due to a group, assuming that the total load is uniformly distributed among all the reactors that can process such jobs. We propose this measure because it reflects the thinking of the shop managers in the absence of any formal scheduling system. As the computational results show, this measure plays a useful role in selecting the algorithm for the scheduling system.

We define a "homogeneous" product mix as one in which the variation in the reactor indices is "small." In contrast, a "heterogeneous" product mix is characterized by a "large" variation in these indices. These definitions are vague and qualitative in nature and are intended to serve only as a diagnostic in choosing the appropriate heuristic. Products with variations in process requirements would give rise to wide spread in the reactor indices and would be classified as "heterogeneous." The variation in reactor indices is indicative of the imbalance in the reactor loads. These measures for the two data sets tested are displayed in Table 1. It can be observed that for data set 1 the RG index varies between 21.28 and 158.67, while for data set 2 the variation is from 20.65 to 44.22. This suggests that the job data for the second set is relatively more homogeneous than for the first. In the discussion which follows we refer to the products in data bases 1 and 2 as heterogeneous and homogeneous products, respectively.

Table 2 compares the performance of the heuristics for some typical cases. In the absence of an optimal solution, we chose, for each problem, the best performance measure among the five heuristics as the reference point and measured the deviations with respect to this base. Table 2 provides the average of these deviations for the ten problems in each case. The deviations are expressed as a percentage of the mean value of the reference point for the ten problems. The table provides these measures for the make-span and tardiness objectives. It can be observed that while heuristic H5 gives schedules with very good make-span, its performance with respect to tardiness-related criteria is quite poor when the due dates are dependent on process times. In contrast with independent due dates, H5 provides least make-span schedules that are typically within 10% of others for the tardiness criteria. This supports our conjecture that the hierarchical heuristic H5 should provide reasonable schedules for tardiness criteria as well, when the due dates are not product group dependent. The table also highlights the difference between homogeneous and heterogeneous product sets as contrasted by the results for the two data sets. With a heterogeneous product (data set 1), heuristic H4, which does not discriminate between product groups, performs poorly for both make-span and tardiness criteria. These results demonstrate that priority based on product groups significantly improves the quality of the schedules when the products are not homogeneous. We note that the data provided in Table 2 is representative of the results obtained with other problems. We now describe some key results.

1) The two phase hierarchical heuristic H5 has performed well. It provides dominant schedules for the make-span criterion. In fact this method provided the best results for this objective for all but 15 of the 160 problems in the experiment. The deviations from the minimum in these cases is quite small and due to the indivisibilities of the jobs. The make-spans are quite close (within 5%) to the lower bound provided by the result of the convex program. These schedules are

REACTOR	1	2	3	4	5	6	7	8	9	10	11	12
DATA SET 1 RG(•)	47.8	21.28	113.7	26.93	68.05	72.08	86.25	158.67	44.48			
DATA SET 2 RG(•)	38.03	44.22	41.42	34.62	32.85	27.4	33.37	35.62	36.29	24.87	20.65	28.79

TABLE 2: COMPARISON OF HEURISTICS' PEFORMANCE (for selected cases)

		DATA SET 1		DATA SET 2	
		DEPENDENT DUE DATE	INDEPENDENT DUE DATE	DEPENDENT DUE DATE	INDEPENDENT DUE DATE
	Tardiness Factor t	.6	.6	.6	.6
	Range Factor r	.2	.2	.95	.95
	H1	9.66	5.21	10.51	13.84
OBJECTIVE	H2	11.3	7.19	5.43	8.21
MAKE	H3	15.9	11.64	9.09	11.34
SPAN	H4	19.8	6.6	9.73	1.41
	H5	0.0	0.0	0.0	.06
	H1	1.9	0.67	1.89	2.85
OBJECTIVE	H2	23.5	3.47	6.27	2.99
TARDINESS	H3	73.6	4.33	2.69	4.71
	H4	66.7	4.89	0.72	5.37
	H5	173.0	6.44	30.05	12.68

quite good for the tardiness-related criteria. An exception is the case with dependent due dates which are tight (t = 0.6).

2) The trade-off between the make-span and tardiness criteria is most pronounced with data base 1 for the case in which the due dates are dependent on process times and are tight (t = 0.6). In this case H5 yields a make-span which is about 10-20% better than those obtained with the other heuristics. But this is at the expense of tardiness measures. Total tardiness and number of late jobs go up by more than 100%. This result is important in that it clearly highlights the relation between objectives to be attained and the appropriate method for scheduling jobs.

3) When the tardiness factor is set at 0.2, suggesting loose due dates, the tardiness-related criteria become less important. In this case the use of H5 gives schedules with least make-span and negligible penalty in tardiness measures.

4) The situation is similar when due dates are set independently of the process times. Here again, H5 gives least make-span with marginally higher tardiness measures. These results indicate that the heuristic H5 would perform well for the tardiness criterion when the due dates are independent of the product groups.

5) Regarding the heuristics proposed for the tardiness-related objective (H1, H2, H3, H4), the results suggest a strong correlation between heterogeneity in job characteristics and priority based on product groups. In the problems with relatively homogeneous product groups (data set 2), the four methods provide comparable results, typically within 6% of each other. On the other hand, in tests with data set 1, the schedules provided by H4 were inferior to those given by the heuristics H1 and H2. It is interesting to note that H3, in which slack is a factor in determining group priority, does not perform well. In most of the problems tested it gives schedules that are inferior to those given by H1 and H2. H1 performed slightly better than H2 in most cases and seems to offer a compromise between make-span and tardiness criteria. Typically this heuristic gave schedules

with deviations within 15% and 5% respectively for the make-span and tardiness objectives.

In summary, the computational results suggest that, with some exceptions, the two-phase hierarchical approach, i.e. H5, yields schedules with low make-spans that are marginally inferior to those given by the other heuristics in terms of tardiness criteria. An exception to this is the case when the due dates are tight and are process-time dependent. In these instances it is preferable to use two heuristics depending on the objective. While H5 is good for make-span, the choice for the tardiness objective should be guided by the product characteristics. H1 is appropriate when the product set is heterogeneous. For homogeneous products we recommend H4, since it is simpler to implement and provides results comparable to H1, H2, and H3. The experiments also suggest that the use of priority based on product groups significantly affects the quality of the schedules in the presence of heterogeneous products.

4.0 PLANNING MODELS FOR WAFER PROCESSING FACILITIES

The computational experiments described in the previous section strongly suggest that homogeneity of the product mix is an important factor in the choice of an appropriate heuristic for scheduling. These results demonstrate that while the heuristic H4 gives good results with a "homogeneous" product, the quality of the schedules degenerates with increase in heterogeneity. In these instances it is necessary to use heuristics with more elaborate priority rules (such as H1, H2, H5). However, the use of H4 is attractive from a computational and implementation perspective for the following reasons.

1) The heuristics are computed very often in scheduling a dynamic shop.

2) The heuristic H5 requires solution of a quadratic program. While this problem is easy to solve, it does introduce additional complexity.

3) The complexity of computation of $\Delta(k,i)$ in step 2(a) of the algorithm

described in Section 2 is significantly larger for heuristics H1, H2, and H3 than for H4. In H4 this computation is trivial, since job k will always be last and this will not affect the jobs already assigned to reactor i. In contrast, this is not true in the case of the other heuristics. It is necessary to determine the position of job k and compute the increase in tardiness of the jobs which follow it. Moreover, with H4 it is not necessary to define a priority for product groups.

In this section, we consider the problem of "homogenizing" the products. The objective is to obtain a homogeneous set of products that would enable use of the simpler algorithm H4. A preliminary step in this process is the identification of independent shops within the facility. This would decompose the planning and scheduling problems in the facilty and considerably reduce the computational requirements at each stage. Identification of "independent" shops within a facility is relatively straightforward and in many instances can be obtained by inspection. In large facilities, such shops can be determined by constructing a bipartite graph showing the product-reactor dependence. The nodes of the graph represent products and reactors. Every feasible product-reactor assignment is represented by an arc between the respective nodes. Each isolated subgraph in the resulting network corresponds to an independent shop of the facility.

The "homogenization" of the products can be interpreted as a planning problem in which reactors are assigned to the product groups. This has to be done considering the technical constraints that restrict each product group j to a feasible set of reactors I(j). We propose to use the squared coefficient of variation of the reactor indices as a measure of the variation, and the objective is to minimize this measure. We consider a one-year horizon and assume that the demand is deterministic. This is reasonable because at the aggregate (product group) level the variance in annual demand is quite small. This is consistent with other approaches in aggregate planning, e.g., Bitran and Hax (1977), Hax (1978). We also

observe that the demand for specialty wafers is not characterized by any seasonalities. The planning model is described in detail below.

Notation:

M = set of reactors.

m = number of reactors.

G = set of product groups.

g = number of product groups.

$J(i)$ = set of product groups that can be processed on reactor i.

$I(j)$ = set of reactors that can process product group j.

r_i = load assigned to reactor i.

p_{ij} = proportion of load of product j assigned to reactor i.

x_{ij} = indicator variable for assignment of reactor i to product j.

m_j = number of reactors assigned to product j.

d_j = total load due to product j (annual demand of product j).

$\bar{r}$ = average load per reactor = $\frac{1}{m} \sum_{i=1}^{m} r_i$.

MS = make-span = $\max_{1 \leq i \leq m} r_i$.

$$[M1] \qquad Z_1 = \min \sum_{i=1}^{m} (r_i - \bar{r})^2 / (m\bar{r}^2)$$

$$\text{s.t.} \quad r_i = \sum_{j \in J(i)} (d_j/m_j) x_{ij} \qquad i = 1, 2, \ldots, m \qquad (1)$$

$$m_j = \sum_{i \in I(j)} x_{ij} \qquad j = 1, 2, \ldots, g \qquad (2)$$

$$x_{ij} \in \{0,1\}, \quad i \in I(j), \qquad j = 1, 2, \ldots, g \qquad (3).$$

The constraints (1) in the model define the reactor loads and are based on the assumption that the demand of a product group is distributed equally among the reactors that are assigned to the product. This is consistent with the practice observed in the wafer manufacturing facility which generated the data base and corresponds to the reactor indices defined in the previous section. Observe that

the average reactor load $\bar{r} = \sum_{i=1}^{m} r_i/m = \sum_{j=1}^{g} d_j/m$ and is independent of the reactor assignments. We make two additional remarks about problem M1.

i) The model is a non-linear integer program and is in general difficult to solve.

ii) In the absence of any restrictions on the sets $I(j)$, the optimal solution is obtained trivially by assigning all reactors to all the products, i.e. $X_{ij} = 1$ for all i, j, and $Z_1 = 0$.

Since M1 is difficult to solve, we formulate an alternative model, M2, and propose to solve it instead of M1. The model is described below.

$$[M2] \qquad Z_2 = \min \sum_{i=1}^{m} (r_i - \bar{r})^2 / (m\bar{r}^2)$$

$$\text{s.t.} \quad r_i = \sum_{j \in J(i)} d_j p_{ij} \qquad i = 1, 2, \ldots, m \qquad (4)$$

$$\sum_{i \in I(j)} p_{ij} = 1 \qquad j = 1, 2, \ldots, g \qquad (5)$$

$$p_{ij} \geq 0 \qquad i \in I(j),\ j = 1, 2, \ldots, g \qquad (6).$$

The model M2 not only determines the optimal assignments of reactors to product groups but also provides the optimal loading pattern (allocation of load from a product group to different reactors). Thus M2 provides a more "homogeneous" product mix than M1. It may be observed that M2 is a convex program and hence easy to solve. Some properties of this problem are described below. The proofs of some of these propositions can be found in Appendix 2.

Proposition P1: Let LM1 denote problem M1 with the integrality constraints relaxed, i.e.

$$[LM1] \qquad \min \sum_{i=1}^{m} (r_i - \bar{r})^2 / (m\bar{r}^2)$$

s.t. conditions (1), (2), above, are satisfied, together with:

$$0 \leq x_{ij} \leq 1, \qquad i \in I(j),\ j = 1, 2, \ldots, g \qquad (3').$$

Then, M2 and LM1 are equivalent. It follows that $Z_1 \geq Z_2$.

Proposition P2: Any optimal solution to M2 will also minimize the make-span for the problem.

Proposition P3: There exists an optimal solution to M2 such that at most m products are assigned to more than one reactor.

The above proposition suggests that there exists an optimal solution to M2 which is nearly feasible in M1. The following propositions provide a method to obtain a feasible solution to M1.

Define $R(j)$ = set of reactors assigned to product j

$$= \{ i \mid p_{ij} > 0 \}.$$

Proposition P4: In any optimal solution $Y^* = (p^*, r^*)$ to M2, for any $j = 1, 2, \ldots, g$, if $i, k \in R(j)$ then $r_i^* = r_k^*$.

Given a feasible solution $Y = (p,r)$ for M2, let $M_1, M_2, \ldots, M_k$ be a partition of M such that reactors within each partition are equally loaded, i.e.,

$$M = \bigcup_{i=1}^{K} M_i; \qquad M_i \cap M_j = \emptyset \text{ if } i \neq j;$$

if $\ell, s \in M_i$ then $r_\ell = r_s$, for $i = 1, 2, \ldots, K$;

if $\ell \in M_i$, $s \in M_j$, $i \neq j$, then $r_\ell \neq r_s$.

Corollary 1: There exists an optimal solution $Y^* = (p^*, r^*)$ with the following property.

Let A be any subset of products that share at least one reactor in the assignments given by Y^*. Then $\bigcup_{j \in A} R(j) \subseteq M_i$ for some i.

Corollary 2: Let n_i denote the number of products with fractional p_{ij} and assigned to reactors in M_i in the solution Y^*. Then $n_i \leq |M_i|$.

Corollary 3: There exists an optimal solution $Y^* = (p^*, r^*)$ with the following property. Let A' be any subset of products with fractional p_{ij}. Then $|A'| \leq |\bigcup_{j \in A'} R(j)|$.

Algorithm for obtaining a feasible solution to M1:

We assume that a $Y^* = (p^*, r^*)$, optimal for M2 and satisfying Proposition P3 and Corollaries 1, 2, and 3, is the initial solution for obtaining a feasible assignment for M1. Without any loss of generality assume that the product groups are ordered such that $d_1 \geq d_2 \geq \ldots \geq d_g$.

Step 1: Construct a bipartite graph (V,E) as follows.

a) one node for each reactor (sink node)

b) one node for each product with fractional p^*_{ij} (source node)

c) $(i,j) \in E$ if and only if $0 < p^*_{ij} < 1$.

Step 2: Obtain a maximum cardinality matching for the network in step 1. From Corollary 3, and Theorem 8.7 of Hu (1969), it follows that the cardinality of such a matching is equal to the number of products or source nodes.

Step 3: A feasible solution to M1 is obtained as follows.

$\hat{x}_{ij} = 1$ if $0 < p^*_{ij} < 1$ and nodes i, j are matched in Step 2;

$= 0$ if $0 < p^*_{ij} < 1$ and nodes i, j are not matched in Step 2;

$\hat{x}_{ij} = p^*_{ij}$ for the pairs (i,j) not in the graph;

$\hat{r}_i = r^*_i + \sum_{j \in J'} (x_{ij} - p^*_{ij}) d_j$, where the summation is over products with fractional allocation in the optimal solution to M2, i.e.

$J' = \{ j \mid 0 < p^*_{ij} < 1 \text{ for some } i \}$.

Proposition P5: The increase in the reactor load r_i by the application of the above algorithm is less than d_1, the product with the highest demand.

Corollary 4: The make-span of the solution given by the algorithm is within d_1 of the optimal, i.e., $MS^*_2 \leq MS^*_1 \leq \hat{MS}_1 \leq MS^*_2 + d_1 \leq MS^*_1 + d_1$

where $\hat{MS}_1$ is the make-span of the solution from the algorithm and

MS^*_1 and MS^*_2 are the minimum make-spans for problems M1 and M2.

Proposition P6: The relative worst-case error bound on the make-span is given by

$$\frac{\hat{MS}_1 - MS^*_1}{MS^*_1} \leq \frac{m\, d_1}{\sum_{j=1}^{g} d_j} .$$

The following proposition provides a bound on the deviation of the objective from the optimal value and demonstrates that in some sense it is asymptotically optimal.

Proposition P7: $0 \leq \hat{Z}_1 - Z_1 \leq \hat{Z}_1 - Z_2 \leq \min\{2md_1 / \sum_{j=1}^{g} d_j , \ (m-1)(md_1 / \sum_{j=1}^{g} d_j)^2\}$

where $\hat{Z}_1$ is the objective value for the approximate solution obtained from the algorithm.

The Models M2 and AP:

We briefly comment about problems M2 and AP described in Section 2. Both are convex programs with essentially the same structure and are designed to determine the allocation of reactors to product groups. In fact, as the following proposition makes it clear, if the availability factor av(i) is the same for all reactors, the set of optimal solutions is the same for both problems.

Proposition P8: Consider problems M2 and AP. Assume that the availability factor av(i) is same for all reactors (say 1.0). Then any optimal solution to AP is optimal in M2 and vice versa, i.e., $Y^* = (x^*, r^*)$ is optimal in AP if and only if $Y^* = (p^*, r^*)$ is optimal in M2 (where $x^*(i,j) = d_j p^*_{ij}$ defines the relation between $x^*(i,j)$ and p^*_{ij}).

The essential difference between the two models is in the emphasis and applica-tion. Observe that while the structure of the two problems is similar, the data used is quite different. In M2 we use aggregate annual demand data while the data for AP is dynamic and reflects the shop load at that time. M2 is essentially a planning problem and the focus is on reactor product assignments. The allocations p_{ij} determined by this problem are not very meaningful at the operational level due to the dynamic character of the shop.

Problem AP, on the other hand, is part of the scheduling heuristic and is solved periodically on a rolling horizon basis. The product-reactor load allocation is meaningful in the short run and should be consistent in the long run with

the assignments determined at the planning stage.

Finally it should be remembered that the objective of the planning models M1 and M2 is the "homogenization" of the product. If this is successful (indicated by a small value for the objective function), the scheduling of jobs can be done by heuristic H4. If the heterogeneity still persists, we recommend the use of H5 for the make-span objective, and if necessary the use of H1 or H2 for the tardiness criterion.

5.0 CONCLUSIONS:

In this paper we have focused on two aspects that are important for the successful design and implementation of a class of scheduling systems, and we have described an application in the semiconductor industry. Typically the scheduling literature on parallel machines considers a single objective and a single product. In contrast, more realistic situations are characterized by multiple objectives and a range of products with varying resource requirements. We analyzed a scheduling application with two different objectives and proposed five heuristics. The results of the computational experiments suggest that the two-phase hierarchical approach (referred to as heuristic H5) gives reasonable results under a variety of circumstances. In this method we first solve a quadratic program to determine the allocation of work among machines at an aggregate level so as to minimize make-span. In the second phase individual jobs are assigned to the reactors. This heuristic has, in general, performed well for both make-span and tardiness related criteria. Based on the computational results, we have identified some conditions when this method does not give good results for the tardiness objective. In these cases we recommend an alternative algorithm.

The computational results also suggest that priority based on product groups leads to significantly better schedules when the products are heterogeneous. We propose a reactor index to detect the presence of heterogeneity in the product

mix. A wide variation in these measures would suggest that the products are not homogeneous, and in these cases algorithms which discriminate between products (e.g. heuristics H1, H2) perform better than the traditional approach (heuristic H4).

We also described a planning model to assign reactors to product groups to achieve a homogeneous set of products. This can be interpreted as an attempt to reduce the complexity of the problem by creating smaller, independent shops with homogeneous products. A successful effort at the planning stage would enable the use of a simpler algorithm and reduce the complexity of the scheduling system. In this context, we showed that an approximate solution obtained by solving the easier problem (M2) instead of the nonlinear program M1 can be expected to give reasonably good results in many cases of practical interest. The performance of the bounds show that the assignments given by the approximation are such that the worst-case error for the objective function of M1 (proposition P7) and the relative error for the make-span (proposition P6) are small. Also the solution is asymptotically optimal.

ACKNOWLEDGEMENTS:

The authors are grateful to Professor Hirofumi Matsuo for his helpful comments on an earlier version of this paper.

REFERENCES

1) Baker, K.R. Introduction to Sequencing and Scheduling, Wiley (1974).

2) Baker, K.R. and J.B. Martin. "An Experimental Comparison of Solution Algorithm for the Single Machine Tardiness Problem," NRLQ 21(1) 1974, 187-200

3) Bernardo J.J and K.S. Lin. "Scheduling Independent Jobs on Non-Uniform, Unequal Processors," J. of Op. Mgt., Vol. 4, Aug. '84

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

5) Bradley, S.P., A.C. Hax, and T.M. Magnanti. Applied Mathematical Programming, Addison Wesley, 1977

6) Conway, R.W., W.L. Maxwell, and L.W. Miller. Theory of Scheduling, Addison Wesley, 1967

7) Davis, E. and J.M. Jaffe. "Algorithm for Scheduling Tasks on Unrelated Processors," J. of the A.C.M., Vol. 28, 1981

8) Federgruen, A. and H. Groenevelt. Preemptive Scheduling of Uniform Machines by Ordinary Network Flow Techniques", to appear in OR.

9) Frank, J.B.G. and A.H.G. Rinnoy Kan. "The Asymptotic Optimality of LPT Rule." Report 8418/0 Erasmus University, Rotterdam, The Netherlands, (1984).

10) Hax, A.C. "Aggregate Production Planning," in Handbook of Operations Research. J. Moder and S.E. Elmaghraby, eds. Van Nostrand Reinhold, (1978).

11) Horn, W.A., "Minimizing Average Flow Time with Parallel Machines," OR (21), 846-847 (1973).

12) Hochbaum, D.S. and D.B. Shmoys. "A Packing Problem You can Almost Solve by Sitting on Your Suit Case," University of California, Berkeley, June 1984.

13) Hu, T.C. Integer Programming and Network Flows, Addison Wesley (1969).

14) Karmarkar, N. and R.M. Karp. "The Differencing Method of Set Partitioning." Working Paper, Univ. of California, Berkeley, UCB/CSD 82/113, Dec. 1982.

15) Labetoulle, J., E.L. Lawler, and J.K. Lenstra. "Preemptive Scheduling of Uniform Machines Subject to Release Dates," Progress in Combinatorial Optimization (1984).

16) Lawler, E.L., J.K. Lenstra, and A.H.G. Rinnooy Kan. "Recent Developments in Deterministic Sequencing and Scheduling: A Survey," in Deterministic and Stochastic Scheduling, Dempster, M.A.H., J.K. Lenstra, and A.H.G. Rinnoy Kan, eds. Reide Dordrecht (1982).

17) Schrage, L. "Linear, Integer, and Quadratic Programming with LINDO." The Scientific Press, Calif. (1984).

18) Shapiro, J.F. Mathematical Programming: Structures and Algorithms. John Wiley and Sons (1979).

APPENDIX 1

Proposition A1: Any optimal solution to AP will also minimize the make-span of the problem.

Outline of proof. Let $Y^* = (x^*, r^*, MS^*)$ be an optimal solution to AP. Then there exist multipliers λ^*, α^* which together with Y^* satisfy the following Kuhn-Tucker (KT) conditions (note that AP is a convex program with linear constraints):

a) Y^* solves the problem DAP with $\lambda = \lambda^*$ and $\alpha = \alpha^*$.

[DAP]:

$$L(\lambda,\alpha) = \min \sum_{i=1}^{m} (MS - r_i/av(i))^2 + \sum_{i=1}^{m} \lambda_i [r_i - \sum_{j \in J(i)} x(i,j)] + \sum_{i=1}^{m} \alpha_i [(r_i/av(i)) - MS]$$

$$\text{s.t.} \quad \sum_{i \in I(j)} x(i,j) = d_j, \qquad j = 1, 2, \ldots, g;$$

$$x(i,j) \geq 0, \qquad i \in I(j), \quad j = 1, 2, \ldots, g.$$

b) $\alpha_i^* (r_i^*/av(i) - MS^*) = 0, \qquad i = 1, 2, \ldots, m.$

c) $\alpha_i^* \geq 0, \qquad i = 1, 2, \ldots, m.$

Let AP' be the problem AP with minimizing MS as the objective function. A feasible solution $\bar{Y} = (\bar{x}, \bar{r}, \bar{MS})$ is optimal in AP' if and only if there exist multipliers $\bar{\lambda}$, $\bar{\alpha}$ satisfying the following KT conditions.

d) $\bar{Y}$ solves the problem DAP' with $\lambda = \bar{\lambda}$ and $\alpha = \bar{\alpha}$.

[DAP']:

$$L'(\lambda,\alpha) = \min \; MS + \sum_{i=1}^{m} \lambda_i [r_i - \sum_{j \in J(i)} x(i,j)] + \sum_{i=1}^{m} \alpha_i [(r_i/av(i)) - MS]$$

$$\text{s.t.} \quad \sum_{i \in I(j)} x(i,j) = d_j, \qquad j = 1, 2, \ldots, g;$$

$$x(i,j) \geq 0, \qquad i \in I(j), \quad j = 1, 2, \ldots, g.$$

e) $\bar{\alpha}_i (\bar{r}_i/av(i) - \bar{MS}) = 0, \qquad i = 1, 2, \ldots, m.$

f) $\bar{\alpha}_i \geq 0, \qquad i = 1, 2, \ldots, m.$

Let $\bar{Y} = Y^*$, $\bar{\alpha} = \alpha^*$, $\bar{\lambda} = \lambda^*$. Note that this solution satisfies (e) and (f). The second and third terms in the objective function of DAP' are equal to zero by feasibility and complementary slackness, respectively. Also note that $L'(\lambda^*,\alpha^*) = MS^* = \max_{1 \leq i \leq m} r_i^*$, since Y^* is feasible in AP'. It follows by weak duality that Y^* solves AP'.

APPENDIX 2

Define X_1 = set of feasible solutions to M1

X_2 = set of feasible solutions to M2

X_1^1 = set of feasible solutions to LM1.

Proposition P1: M2 and LM1 are equivalent.

Outline of Proof: (a) Let $Y_1 = (x, m, r) \in X_1^1$.

Consider the solution $Y_2 = (p, \hat{r})$ constructed as follows:

$p_{ij} = x_{ij}/m_j$, $\quad i \in I(j),\ j = 1, 2, \ldots, g;$

$\hat{r}_i = r_i$, $\quad i = 1, 2, \ldots, m.$

It can be shown that $Y_2 \in X_2$.

Therefore for every $Y_1 \in X_1^1$ there exists a $Y_2 \in X_2$.

(b) Similarly let $Y_2 = (p, \hat{r}) \in X_2$

and consider the solution $Y_1 = (x, m, r)$ defined as follows:

$x_{ij} = p_{ij}$, $\quad i \in I(j),\ j = 1, 2, \ldots, g$

$m_j = 1$, $\quad j = 1, 2, \ldots, g$

and $r_i = \hat{r}_i$, $\quad i = 1, 2, \ldots, m.$

Clearly $Y_1 \in X_1^1$. Hence for every $Y_2 \in X_2$ there exists $Y_1 \in X_1^1$.

Since the value of the objective function is identical for corresponding solutions Y_1 and Y_2, the equivalence between M2 and LM1 follows.

LEMMA L1: Let $Y = (p, r)$ be a feasible solution to M2. Define a partition of the reactor set M as follows.

$M' = \{i \mid r_i = MS\}$, and $\overline{M}' = \{i \mid r_i < MS\}$ where

MS = make-span of the solution $Y = \max_i r_i$

Let $J' = \{j \mid p_{ij} > 0 \text{ for some } i \in M'\}$.

Then if $MS > MS_2^*$ there exists at least one product $j \in J'$ such that $I(j) \bigcap \overline{M}' \neq \phi$, where MS_2^* is minimum make-span for M2.

Proof: Assume that there exists a solution $Y = (p,r)$ such that $MS > MS_2^*$ and $I(j) \bigcap \overline{M}' = \phi$ for all $j \in J'$ where J', M', $\overline{M}'$ are defined as above. Then in any feasible solution $\hat{Y} = (\hat{p}, \hat{r})$ we have, $\sum_{i \in M'} \hat{r}_i \geq \sum_{j \in J'} d_j = |M'|\ MS$

since $|M'|\ \hat{MS} \geq \sum_{i \in M'} \hat{r}_i$ it follows that $\hat{MS} \geq MS$. In particular this is true for a solution which minimizes make span and hence $MS_2^* \geq MS$. This contradicts the assumption that $MS_2^* < MS$ and the Lemma follows.

PROPOSITION P2: Any optimal solution to M2 will also minimize the make-span of the problem.

Kuhn-Tucker conditions can be used to prove this proposition. The proof would parallel that of proposition A1. We provide an alternative proof based on incremental analysis.

Proof: Suppose the proposition is false. Then there exists an optimal solution $\hat{Y} = (\hat{p}, \hat{r})$ for M2 with a make-span MS larger than MS_2^*. By Lemma L1, there exists a product $j \in J'$ and reactors i, k such that $i,k \in I(j)$, $i \in M'$, $k \in \overline{M}'$ and $\hat{p}_{ij} > 0$ where J', M' and $\overline{M}'$ are as defined in Lemma L1 for the solution $\hat{Y}$. Consider a solution Y = (p, r) defined as follows

$$\begin{aligned} p_{\ell s} &= \hat{p}_{\ell s} + \Delta \quad , && \text{if } \ell = k,\ s = j \\ &= \hat{p}_{\ell s} - \Delta \quad , && \text{if } \ell = i,\ s = j \\ &= \hat{p}_{\ell s} && \text{otherwise} \\ r_{\ell} &= \hat{r}_{\ell} + \Delta\, d_j && \text{if } \ell = k \\ &= \hat{r}_{\ell} - \Delta\, d_j && \text{if } \ell = i \\ &= \hat{r}_{\ell} && \text{otherwise} \end{aligned}$$

where Δ is some suitably chosen small positive fraction such that

$$0 < \Delta < \min\ (\hat{p}_{ij},\ (\hat{r}_i - \hat{r}_k)/2d_j)$$

By construction Y is feasible in M2 and attains a lower objective value than $\hat{Y}$ since

$$\sum_{\ell=1}^{m} (r_\ell - \bar{r})^2 - \sum_{\ell=1}^{m} (\hat{r}_\ell - \bar{r})^2 = (r_i - \bar{r})^2 + (r_k - \bar{r})^2 - (\hat{r}_i - \bar{r})^2 - (\hat{r}_k - \bar{r})^2$$

$$= -\Delta\, d_j\, (r_i + \hat{r}_i - r_k - \hat{r}_k) < 0$$

This contradicts the assumption that $\hat{Y}$ is optimal and the proposition follows.

REMARK: Note that M2 is a convex program. The objective function is convex and the constraints are linear. Therefore, Kuhn-Tucker (KT) conditions are necessary and sufficient for optimality. Let (Y*) and (λ*) be respectively a primal and a dual pair of solutions which satisfy the KT conditions.

PROPOSITION P3: There exists an optimal solution to M2 with at most m products with fractional p_{ij}.

Outline of Proof: Consider the partial dual D1 of M2.

$$\text{D1: } L(\lambda) = \min \sum_{i=1}^{m} (r_i - \bar{r})^2 / m\bar{r}^2 + \sum_{i=1}^{m} \lambda_i \Big(r_i - \sum_{j \in J(i)} d_j p_{ij}\Big)$$

$$\text{s.t. (5) and (6)}$$

$$= \min \sum_{i=1}^{m} [\ (r_i - \bar{r})^2 / m\bar{r}^2 + \lambda_i r_i\ ] - \sum_{i=1}^{m} \sum_{j \in J(i)} \lambda_i d_j p_{ij}$$

$$\text{s.t. (5) and (6)}$$

Let $r_i(\lambda)$, and $p_{ij}(\lambda)$ be optimal in D1 for a given λ. We make the following observations regarding problem D1.

(1) For a given λ, $r_i(\lambda) = \bar{r} - \lambda_i\, m\bar{r}^2/2$

(2) For a given λ, $p_{ij}(\lambda)$ solves the following linear program

$$\min \sum_{i=1}^{m} \sum_{j \in J(i)} (-\lambda_i d_j) p_{ij}$$

s.t. (5) and (6)

From (1) and (2) it follows that there is a pair $Y^* = (p^*, r^*)$ and λ^* of primal and dual solutions such that p_{ij}^* solves the linear program LP1.

[LP1] $$\min \sum_{i=1}^{m} \sum_{j \in J(i)} (-\lambda_i^* d_j) p_{ij}$$

s.t. (5), (6) and

$$\sum_{j \in J(i)} d_j p_{ij} = \bar{r} - \lambda_i^* \bar{mr}^2/2, \quad i = 1, 2, \ldots m$$

From LP theory we know that there exists an extreme point solution to LP1. At most m+g p_{ij}'s are positive at an extreme point. Since for each product at least one $p_{ij} > 0$, we have at most m products with fractional p_{ij} at an extreme point. Hence there exists an optimal solution with at most m products with fractional p_{ij}.

PROPOSITION P4: Let $Y^* = (p^*, r^*)$ be an optimal solution to M2 and let R(j), j = 1, 2, ... g define the corresponding reactor assignments.

If $i,k \in R(j)$, then $r_i^* = r_k^*$, j = 1, 2, ... g.

Outline of Proof: Let Y^* be an optimal solution to M2 and let λ^* be the corresponding multipliers such that Y^*, λ^* satisfy the KT conditions. Then by arguments similar to those in the proof of proposition P3, we can show that p^* solves the following linear program

$$\min \sum_{i=1}^{m} \sum_{j \in J(i)} (-\lambda_i^* d_j) p_{ij}$$

s.t. (5) and (6)

This implies that if $p_{ij}^* > 0$, $p_{kj}^* > 0$ i.e., $i, k \in R(j)$ then $\lambda_i^* = \lambda_k^*$ and $r_i^* = r_k^*$.

COROLLARY 1 There exists an optimal solution $Y^* = (p^*, r^*)$ to M2 with the following property.

Let R(j), j = 1, 2, ... g be the corresponding reactor assignments and let M_i, i = 1, 2, ... K be a partition of the reactors so that for each i=1,2,...K, M_i is a set of reactors with the same load. Let A be any subset of products that share at least one reactor in the solution Y^*. Then $\bigcup_{j \in A} R(j) \subseteq M_i$ for some i.

Proof: follows from proposition P4 and definition of M_i.

COROLLARY 2 Let n_i denote the number of products with fractional p_{ij} and assigned to reactors in M_i in the solution Y*. Then $n_i \leq |M_i|$.

Outline of Proof: Note that p^*_{ij} decompose the problem LP1 into K subproblems and solve them optimally.

Subproblem ℓ: LP1(ℓ):

$$\min \sum_{i \in M_\ell} \sum_{j \in J'(\ell)} (-\lambda_i^* d_j) p_{ij}$$

$$\text{s.t.} \sum_{i \in M_\ell \cap I(j)} p_{ij} = 1 , \qquad j \in J'(\ell)$$

$$\sum_{j \in J'(\ell)} d_j p_{ij} = \bar{r} - \lambda_i^* m\bar{r}^2/2 , \quad i \in M_\ell$$

$$p_{ij} \geq 0 , \qquad i \in I(j) \cap M_\ell , \; j \in J'(\ell)$$

where $J'(\ell)$ is the set of products assigned to reactors in M_ℓ in the solution Y*.
By Corollary 1, $J'(\ell) \cap J'(s) = \phi$ if $\ell \neq s$
An extreme point solution to LP1(ℓ) will have at most $|M_\ell|$ products with fractional p_{ij} (by arguments similar to those in proposition P3). This proves the corollary.
The proof of Corollary 3 parallels that of Corollary 2.

PROPOSITION P5: The increase in reactor load r_i, by the application of the algorithm is less than d_1.
Outline of Proof: By step 2 of the algorithm, at most one x_{ij} is set at 1 for each i and $j \in J' = \{j \mid 0 < p^*_{ij} < 1 \text{ for some } i\}$

$$r_i = r_i^* + \sum_{j \in J'} (x_{ij} - p^*_{ij}) d_j < r_i^* + \sum_{j \in J'} x_{ij} d_j \leq r_i^* + d_1$$

PROPOSITION P6: The relative worst case error bound on the make-span is given by the following.

$$(\hat{MS}_1 - MS_1^*) / MS_1^* \leq md_1 / \sum_{j=1}^{g} d_j$$

where $\hat{MS}_1$ = make-span of the solution obtained by the algorithm
MS_1^* = minimum make-span for M1
Proof: From propositions P5 and P2, $\hat{MS}_1 \leq MS_2^* + d_1$
From proposition P1 $MS_1^* \geq MS_2^*$

Since $MS_2^* \geq \sum_{j=1}^{g} d_j/m$,

we have $(\hat{MS}_1 - MS_1^*) / MS_1^* \leq (\hat{MS}_1 - MS_2^*) / MS_2^* \leq md_1 / \sum_{j=1}^{g} d_j$

PROPOSITION P7: Let $Y^* = (p^*, r^*)$ be an optimal solution to M2 satisfying proposition P3 and let $\hat{Y} = (\hat{x}, \hat{r})$ be the corresponding approximate solution to M1 by using the algorithm.

Then $\hat{Z}_1 - Z_1 \leq \hat{Z}_1 - Z_2 \leq \min \{2\, md_1/d\ ,\ (m-1)\,(md_1/d)^2\}$

where $\hat{Z}_1$ is the value of the objective function for $\hat{Y}$ and $d = \sum_{j=1}^{g} d_j$.

Outline of Proof: $\sum_{i=1}^{m} (\hat{r}_i - \bar{r})^2 - \sum_{i=1}^{m} (r_i^* - \bar{r})^2$

$$= \sum_{i=1}^{m} (\hat{r}_i^2 - r_i^{*2})$$

$$= \sum_{i=1}^{m} (\hat{r}_i - r_i^*)\,(\hat{r}_i + r_i^*)$$

$$\leq d_1 \sum_{i=1}^{m} (\hat{r}_i + r_i^*) = 2\, d_1 d$$

Hence $\hat{Z}_1 - Z_2 \leq 2\, d_1 d \,/\, (m\bar{r}^2) = 2\, md_1/d$ (a)

Also $\sum_{i=1}^{m} (\hat{r}_i^2 - r_i^{*2}) = \sum_{i=1}^{m} (\hat{r}_i - r_i^*)^2 + \sum_{i=1}^{m} 2\, r_i^* (\hat{r}_i - r_i^*)$.

Since r^* is a constant for $i \in M_k$ and $\sum_{i \in M_k} (\hat{r}_i - r_i^*) = 0$

$$\sum_{i=1}^{m} r_i^* (\hat{r}_i - r_i^*) = \sum_{k=1}^{K} \sum_{i \in M_k} r_i^* (\hat{r}_i - r_i^*) = 0$$

and $\sum_{i=1}^{m} (\hat{r}_i^2 - r^{*2}_i) = \sum_{i=1}^{m} (\hat{r}_i - r_i^*)^2 \leq (m-1)\, d_1^2 + (m-1)^2\, d_1^2 = m(m-1)\, d_1^2$

The last inequality follows from the fact that at worst d_1 is added to m-1 reactors and $(m-1)d_1$ is subtracted from the m^{th} reactor.

Hence $\hat{Z}_1 - Z_2 \leq m(m-1)\, d_1^2 / (m\bar{r}^2) = (m-1)\,(md_1/d)^2$ (b)

From (a) and (b) and noting that $Z_1 \geq Z_2$, we have

$$0 \leq \hat{Z}_1 - Z_1 \leq \hat{Z}_1 - Z_2 \leq \min \{2\, md_1/d\ ,\ (m-1)\,(md_1/d)^2\}$$

PROPOSITION P8: Consider the problems M2 and AP. Assume that the availability factors av(i) are the same for all reactors, (say av(i) = 1). Define the corresponding solutions $\hat{Y} = (p, \hat{r})$, $Y = (x, r)$ for M2 and AP as follows.

$x(i,j) = d_j \, p_{ij}$, $\quad i \in I(j)$, $j = 1, 2, \ldots g$

$r_i = \hat{r}_i \qquad i = 1, 2, \ldots m$

Then Y is optimal in AP if and only if $\hat{Y}$ is optimal in M2.

Outline of Proof: Clearly Y is feasible in AP if and only if $\hat{Y}$ is feasible in M2.

$$\text{Note that } \sum_{i=1}^{m} (MS - r_i)^2$$

$$= \sum_{i=1}^{m} (r_i - \bar{r} + \bar{r} - MS)^2$$

$$= \sum_{i=1}^{m} (r_i - \bar{r})^2 + m(\bar{r} - MS)^2$$

Also observe that the reactor loads r_i, $\hat{r}_i$ are identical for Y and $\hat{Y}$ by definition. Hence the make-spans are identical for corresponding solutions. By propositions P2 and A1, the optimal solutions for M2 and AP also minimize the make-span MS. Therefore Z_{AP} and $m\bar{r}^2 \, Z_2$ differ by the constant $m(\bar{r} - MS^*)^2$. Consequently, Y is optimal in AP if and only if $\hat{Y}$ is optimal in M2.