AN ABSTRACT OF THE THESIS OF

Cynthia P. Juwono for the degree of Master of Science in Industrial and Manufacturing Engineering presented on March 16, 1992.

Title : A Heuristic Algorithm for Multistage Scheduling in Food Processing Industry.

Redacted for Privacy

Abstract Approved : _______________

Sabah Randhawa

A multistage production system consists of a number of production stages that are interrelated, that is the output from one stage forms input to the next stage. There are constraints associated with each stage as well as constraints imposed by the overall system. Besides, there are multiple objectives that need to be satisfied, and in numerous cases, these objectives conflict with each other. What is required is an efficient technique to allocate and schedule resources so as to provide a balance between the conflicting objectives within the system constraints.

This study is concerned with the problem of scheduling multistage production systems in food processing industry. The system and products have complex structure and relationships. This makes the system difficult to be solved analytically. Therefore, the problem is solved by developing a heuristic algorithm that considers most of the constraints. The output generated by the algorithm includes a production schedule

which specifies the starting and completion times of the products in each stage and the machines where the products are to be processed. In addition, a summary of system performances including throughput times, resources' utilizations, and tardy products is reported.

A HEURISTIC ALGORITHM FOR MULTISTAGE SCHEDULING IN FOOD PROCESSING INDUSTRY

by

Cynthia P. Juwono

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed March 16, 1992

Commencement June 1992

APPROVED :

Redacted for Privacy

Professor of Industrial and Manufacturing Engineering in charge of Major

Redacted for Privacy

Head of department of Industrial and Manufacturing Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented March 16, 1992

Typed by Budiman Saleh for Cynthia P. Juwono

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my major professor Dr. Sabah Randhawa for his continuous support, advice, and encouragement throughout this research.

Also, I would like to thank Dr. Sheikh Burhanuddin for his helpful discussion during the research.

I would like to thank Steve Mather from Oregon Freeze Dry for providing data and information.

I would like to thank to many individuals from PT IPTN, Indonesia, for their assistance during my study at Oregon State University.

I would especially like to thank my parents, sisters, brother, and Oom & Tante Bosman whose support, love, and encouragement have made this possible.

Finally, I must thank Budi for his help, love, and understanding.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

A HEURISTIC ALGORITHM FOR MULTISTAGE SCHEDULING IN FOOD PROCESSING INDUSTRY

CHAPTER 1. INTRODUCTION

1.1 Importance and Complexity of Multistage Problem

Most manufacturing processes, both in discrete (metal working) and in continuous (food processing, paper manufacturing) operations, involve a number of production stages. These stages are interrelated, that is, the output from one stage forms input to the next stage. Each stage has specific constraints that determine the output at that stage. These include machine capacities, resource requirements, setup requirements, and space limitations. There are also constraints imposed by the overall system. Example of such constraints include order size, product mix, due dates, and the resource availability.

The system usually has several objectives that conflict with each other such as maximizing machine utilizations, maximizing throughput, meeting due dates, and minimizing work-in-process inventories. In order to meet these objectives, control and coordination of the total system is required. This includes the allocation and scheduling the raw materials, machines, and other resources at the right time and in the right amount so as to obtain a balance of the conflicting objectives.

The problem of controlling multistage production systems involves two phases :

1. Developing an initial production schedule that satisfies the stage-level and system-level constraints and meets the production objectives.
2. Modification of the initial schedule during operation, if necessary. For example, if mechanical failures, labor inadequacies, and raw material delays occur during operation, then the initial schedule may have to be modified.

1.2 Objective and Scope of Study

This study is concerned with scheduling a multistage production system in food processing industry. The raw materials are processed through several operations in series. The complete set of operations is called freeze-drying process.

Each stage of the operations has specific constraints related to machine capacities and resource requirements. In addition, due to the nature of the products and the freeze-drying process itself, work-in-process inventories between stages are not allowed. The product mix to be processed is determined every week and may vary from one week to the next week.

The objective of this study was to develop an initial production schedule for a specified planning horizon. The schedule specifies the starting and completion times of products in each stage and on particular machines on which the products are to be processed. The study focuses only on the first phase of scheduling, that is, development of an initial production schedule.

1.3 Solution Approach

The complexity of the system makes it difficult to model the problem using analytical techniques. A heuristic algorithm was developed to provide a good, practical production schedule. The development of the algorithm was based on the following steps :

1. Identifying the structure and characteristic of the system.
2. Determining parameters that may affect the system performance.
3. Identifying necessary assumptions.
4. Developing the conceptual framework for the algorithm.
5. Developing the detailed structure of the algorithm.
6. Implementing a computer model to implement the algorithm.
7. Verifying the computer model and validating the algorithm performance.

1.4 Organization of Thesis

The organization of this thesis is as follows :

- Chapter 2 provides the background of scheduling problems, in particular the multistage scheduling problem, and some of the techniques developed to analyze the problem; a brief description of the freeze-drying process; and the description of the system under study.

- Chapter 3 describes the development of the algorithm.
- Chapter 4 provides the implementation details of the algorithm, its application using actual production data, and discussion of the results.
- Chapter 5 summarizes the research work and concludes with suggestions for extensions of this study.

CHAPTER 2. BACKGROUND

2.1 Scheduling Problem

Scheduling problems can be classified into several categories (Figure 2.1). Based on the nature of the arrival of jobs, the classification of Figure 2.1 divides scheduling problems into two categories : static and dynamic. In the static case, a set of jobs is available at the beginning of the planning period; no new jobs arrive at the system after the initial set. On the other hand, in the dynamic case jobs arrive at the system continuously over the planning period.

The second level of Figure 2.1 breaks the problems based on number of machines in the system. Based on the number of machines involved, there are single stage problems with one machine in the system and multistage problems with several machines in the system. Multistage problems can in turn be classified into three cases. These are parallel machines, machines in series, and combination of parallel and series stages or hybrid system. In parallel machine case, there are more than one identical machines and the jobs have to be processed exactly once on one of the machines. In the series case, the system consists of several machines performing different operations, and the jobs are required to be processed on more than one machine in sequence. If there are several identical machines for processing one operation, then the system is called a hybrid system.

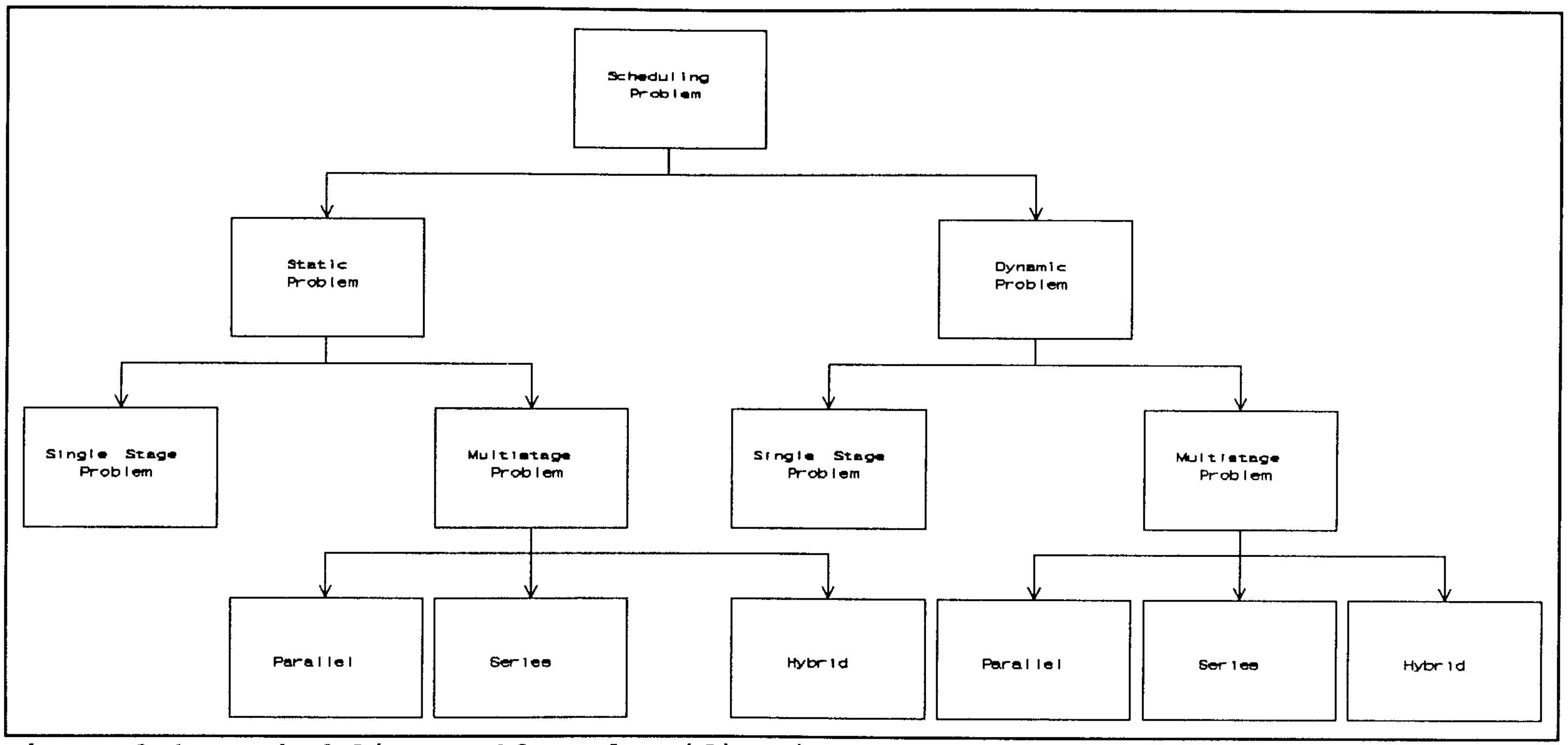

Figure 2.1 : Scheduling problem classification (Day and Hottenstein [1970]).

Scheduling or priority rules have been developed as an attempt to deal with the complexity of the scheduling problem that result when an attempt is made to solve the problem analytically. These rules are used to select the next job to be processed at a machine when that machine becomes available and more than one job is waiting to be processed. Example of commonly used scheduling rules are : shortest processing time (SPT), earliest due date (EDD), first-in-first-out (FIFO), and slack-based rules. Combination of simple rules has been widely used to analyze complex scheduling problems. It has been shown that for both static and dynamic problems, SPT rule gives better performance than most other rules (Bedworth and Bailey [1987], Conway [1972] and Day and Hottenstein [1970]).

There has been extensive research in the scheduling area, particularly for discrete systems. Most of prior studies have focused on static scheduling problem, primarily because of the simplicity associated with these problems.

Single stage problems are easier to analyze than multistage problems. Optimum schedules for single stage static problems can be obtained by applying single priority rules such as SPT and EDD (see Baker [1974], Bedworth and Bailey [1987], and Day and Hottenstein [1970]). These types of problems can also be analyzed successfully using mathematical and operation research models. Simple dynamic problems have been analyzed using queueing theory (Day and Hottenstein [1970]).

The complexity and the approaches to analyze multistage problems are briefly explained in the following section.

2.2 <u>Multistage Scheduling Problem</u>

It is difficult to solve most multistage scheduling problems using mathematical or operation research (OR) models due to the complexity of the resulting model. In some cases, solving a system that consists of a large number of machines and jobs analytically requires enormous computational resources to make the problem computationally infeasible. In other cases, due to the inherent assumptions made in the formulation of OR models, the models cannot be used to analyze multistage systems with complex structures and characteristics that do not fit the relatively rigid structure of OR models. Example includes dynamic systems with complex distributions of arrival and service rates.

In addition to the efforts made to solve the multistage problems analytically, the use of heuristic algorithms have been investigated to obtain solutions for these problems. Heuristic algorithms do not guarantee optimality but the solutions to large-scale problems can be obtained with limited computational resources and in a relatively short amount of time.

2.2.1 Static case.

Several algorithms have been developed to analyze parallel machine systems. Elmaghraby and Park [1973] developed a modified branch and bound algorithm. An optimal schedule with minimum makespan can be obtained using McNaughton's algorithm for systems where job preemption is permitted (Baker [1974]). A schedule with minimum makespan for parallel machine system can also be obtained by applying LPT (Longest Processing Time) rule but this schedule may not be optimal (Baker [1974] and Bedworth and Bailey [1987]).

Many algorithms have been suggested for flow shop and job shop scheduling problems. A flow shop system consists of multiple machines in series with the sequence of operations being the same for all jobs. Johnson [1954] developed an algorithm for two-stage flow shop system that yields optimum solution. This algorithm can be extended to three-stage flow shop system under some constraints.

The basic branch and bound approach for flow shops was developed by Ignall and Schrage (see Baker [1974]) for systems which have more than three machines. An algorithm developed by Dudek and Teuton, Jr. [1964] yields optimum sequence with minimum makespan for special structure of flow shops.

Some heuristic algorithms have also been developed to analyze flow shop systems, as for example, algorithms suggested by Campbell, Dudek, and Smith (1970, in Baker [1974]), Palmer (1965, in Baker [1974]), and by Gelders and

Sambandam [1978].

In job shop systems, the sequence of operations of each job may not be the same. Job shop can be formulated as an integer programming problem to obtain an optimal schedule but the computational effort may often be large (Baker [1974]). The branch and bound technique for the job shop problem was developed by Brook and White (see Baker [1974]). Modification of branch and bound techniques was suggested by Pritsker, Miller, and Zinkl [1971] and Barker and McMahon [1985]. Conner [1972] developed a heuristic algorithm called sequencing programming to obtain schedule that minimizes the total sum of squares of tardiness.

Most research in job shop problems uses priority rules and/or combination of priority rules to analyze the system as in Dar-El and Wysk [1982], Day and Hottenstein [1970], Gere, Jr. [1966], and Pritsker, Miller, and Zinkl [1971].

2.2.2 Dynamic case.

Heuristic rules have also been investigated for use in dynamic systems (Day and Hottenstein [1970] and Gere, Jr. [1966]). Jackson [1961] analyzed m parallel machines with dynamic-priority queueing system. Queueing model for two-stage dynamic problem was developed by Buzacott and Shanthikumar [1985].

The algorithms for a multistage system in series (static and dynamic cases) have been developed under the following assumptions :

1. There are no groups of similar machines.
2. Each machine can process only one job at a time.
3. Work-in-process inventory between machines is allowed.
4. Machines never breakdown.
5. Production factors, such as labor, tools, energy, and other resources, are not limited.
6. Move time between machines is negligible.

2.3 **Freeze-Drying Process**

Basically, freeze-drying process consists of several operations that are performed in interrelated stages. This section briefly describes characteristics of the freeze-drying process.

2.3.1 Objectives and advantages of freeze-drying process.

Freeze-drying is a method of preserving biological materials by combining the processes of freezing and desiccation to obtain dried materials with high quality. The contained water in the materials to be preserved is frozen and the ice is removed by sublimation.

Freeze-drying method, as described by Mellor [1978], can be applied to various biological materials as follows :

1. Non-living materials, such as blood plasma, serum, and foodstuff.
2. Living cells intended to remain active for a long time,

such as bacteria, yeasts, and viruses.

3. Surgical transplants that are made non-active, such as bone and skin.

There are several advantages of the freeze-drying process compared to other methods of preservation. These include:

1. Shrinkage in freeze-drying process is less than that in ordinary drying process.
2. Freeze-dried products can be kept for longer periods at room temperature and are easily rehydrated to their original shapes.
3. Freeze-dried products are easy to package, store, and transport.
4. For some types of foods, freeze-drying process is carried out more easily than the dehydration process.
5. The application of this method to foodstuffs will result in freeze-dried foods with good flavor, appearance, and high preservation of nutrients.
6. The loss of volatile components (for example, aroma) is minimized.

2.3.2 Basic steps of the freeze-drying process.

Generally, freeze-drying process involves several successive operations as shown in Figure 2.2.

1. Raw materials.

Biological materials serve as raw materials. Some materials need to be pretreated before being frozen. For example, some foods are cut, ground, or precooked while other materials are pretreated by using chemical agents.

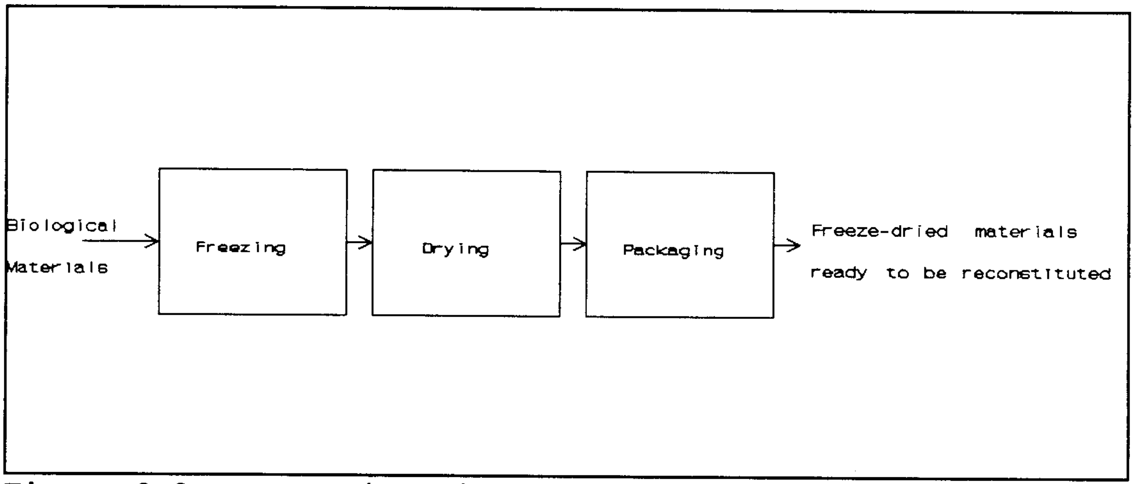

Figure 2.2 : Operations involved in freeze-drying process.

2. Freezing operation.

The objective of the freezing operation is to transform the liquid in the material into a solid state before the next operation (drying) is performed. It is important to carry out the freezing operation in a proper way to protect the main characteristics of the substances and to achieve the correct structure of the solidified material.

3. Drying operation.

The objective of the drying operation is to remove the ice from the frozen material. This operation is performed under reduced pressure in a closed chamber. By supplying the latent heat, the ice in the frozen material slowly sublimes, that is, transforms to vapor. The water vapor is then condensed on a refrigerated coil to prevent it from returning to the product.

The operating conditions (i.e. temperature, pressure, and time) of freezing and drying operations are important and must be controlled carefully. It should also be pointed out that different types of materials require different operating conditions.

4. Packaging and storing.

The freeze-dried products can be kept for long periods. However, since products have ability to absorb moisture, they should be packaged in such a way that there is no contact with water vapor and oxygen.

5. End products.

The end products of freeze-drying process are dry, light, and porous. They are easily reconstituted to their original conditions (structure and appearance) by the addition of liquid, which in most cases can be done with water.

2.4 **Problem Description**

The system studied in this research is a multi-stage production system used for the freeze-drying operations at Oregon Freeze Dry (OFD). OFD, located in Albany, Oregon, produces a wide variety of freeze-dried foods. The products are processed through several successive stages or operations; the processing sequence for all products is the same (preparation, freezing, drying, and packaging).

2.4.1 Description of the operations.

The system under consideration consists of four stages designated as wet-side operation, freezing operation, chamber operation, and dry-side operation (see Figure 2.3). The terminologies used here are specific to OFD.

The preparation of raw materials, such as cooking, slicing, or dicing, is done in the first stage (wet-side operation). The prepared materials are loaded on carts. One cart is equivalent to one batch of product. The carts are then sent to cold room where the freezing operation is carried out. Following the freezing operation, drying operation is performed in closed chambers.

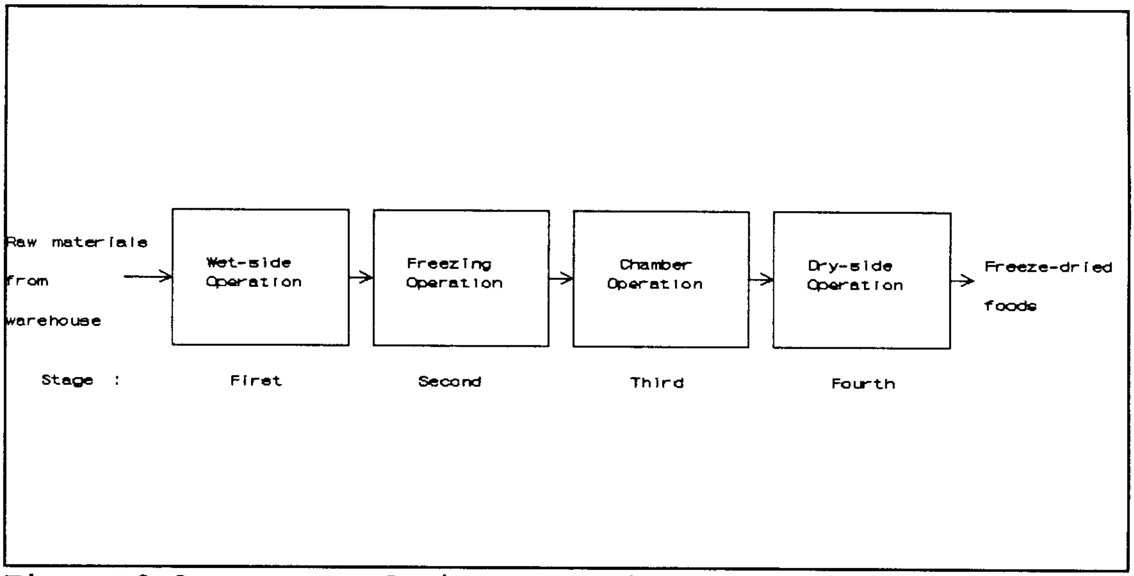

Figure 2.3 : Freeze-drying operations at OFD.

The final operation is called the dry-side operation where the freeze-dried products released from chambers are unloaded from the carts and packaged. This operation is done in rooms called dumping rooms.

All equipment used in these operations is operated 24 hours a day and seven days a week. The product and system characteristics as well as the capacity and resource limitations in each stage are explained in the section below.

2.4.2 Product and system characteristics.

1. Products.

   a. A critical consideration in the chamber operation is the vacuum required for drying a specific product. Based on the vacuum requirements for products processed at OFD, products were divided into three groups, low, medium, and high, referred to as vacuum groups 1, 2, and 3, respectively. Each group represents a range of vacuums.
   b. Each product consists of several batches. The processing time per batch for a specific operation is deterministic, and varies for different types of products. This processing time includes time to load and unload the product to and from the machines.
   c. Because of the characteristics of the freeze-drying process and the nature of the products, there is no work-in-process inventory allowed between some of the operations. Thus, the unavailability of machines (rooms) in the subsequent stage can cause the machines (rooms) in the current stage to be blocked. For example, a batch of

product may complete its processing in a chamber, and should be sent to a dumping room. However, if there is no dumping room available, the batch will not be released from the chamber and remains in the chamber until a dumping room becomes available. During the blockage, the chamber cannot process a new batch of product.

2. Equipment for raw material preparation and cold room.

The machines used to prepare the raw material, and the space capacity of the cold room to hold carts are considered unlimited, based on OFD operations.

3. Chambers.

a. There are nine pairs of tandem chambers, numbered from 1 to 18. Each pair of chambers uses a vacuum pump which is used to evacuate the air in the chamber and sustain the pressure at a certain level based on the product requirement.

b. One chamber can only process one batch of product at a time. Batches which are being processed in the tandem chambers are from the same type of product, so that tandem chambers process products with compatible vacuum requirements.

c. There may be restrictions on specific products only to be processed in specific chambers. These restrictions may be the result of relatively small, but significant variations in vacuum. For the operations being analyzed, products in vacuum groups 2 and 3 can be processed in any chamber but products in vacuum group 1 can only be processed in the first three pairs of chambers (chambers numbered 1 through 6).

d. There may be additional constraints due to resources availability. For example, when processing some types of products, operation of tandem chambers which shares one vacuum pump have to be staggered. This is caused by the vacuum requirements of products that dictate that the vacuum pump may not be used to evacuate air from both chambers at the same time. Similarly, loading and unloading of chambers may have to be staggered if the number of cranes (or material handling devices) used for this purpose are limited.

4. Dumping rooms.

a. There are five dumping rooms that are used in the dry-side operation, numbered from 1 to 5. Each dumping room can only hold one batch of product at a time.

b. Dumping room assignment may depend on product type due to the equipment and structure of dumping rooms. At OFD, dumping rooms 1 and 2 are dedicated to specific products, while dumping rooms 3, 4, and 5 are used for all products other than the few products processed in dumping rooms 1 and 2.

c. Change over of the products in dumping rooms requires cleaning and setup. Cleaning and setup times are product dependent.

2.4.3 Problem complexity.

The chambers are the most capital intensive component of the operation. Therefore, it is desired to maximize the utilization of chambers. However, if the capacity of the dumping rooms is smaller than the capacity of the chambers, increasing the batches processed simultaneously in the chambers increases the possibility of the chambers being blocked because of limited capacity of the subsequent operation. Moreover, the frequent change over of products in dumping rooms will increase cleaning and setup times that may additionally cause delay and blockage in the chambers.

On the average, there are 15 to 25 different types of products produced every week at OFD, and the product mix varies from week to week. The set of products to be processed in a week is specified at the beginning of that week. Actually, there are no specified due dates assigned to the products but all products are expected to be completed by the end of the week. What is required is a weekly production schedule to process the weekly requirements and accommodate the conditions imposed on the system at individual stages and due to interaction among stages. The schedule should give information on the starting and completion times of each batch in the product as well as the particular machine (within each stage) in which the batch is to be processed.

CHAPTER 3. METHODOLOGY

3.1 General Approach

The system under consideration involves a large number of machines and products with their specific characteristics, and has several objectives that may conflict with each other. Solving the problem analytically to obtain optimal solution is difficult (since some characteristics of the problem do not fit to the structure of analytic models) and uneconomical. In this research, a heuristic scheduling algorithm is designed and implemented for developing production schedules for the system. The heuristic algorithm may not yield an optimal solution. However, it yields a good, practical solution to the system because it accounts for most of the constraints.

To develop the algorithm, some assumptions had to be made.

3.1.1 Assumptions.

There are several underlying assumptions that are used in developing the heuristic algorithm. These are :

1. All the machines and equipment in the dry-side operation, cold room, chambers, and dumping rooms are idle at time zero (at the beginning of the week) and ready to process the products.
2. The transportation time between stages is negligible.
3. The breakdown of the machines is ignored

4. Raw materials are always available when needed.

3.1.2 Basic concepts of the algorithm.

This section describes the underlying concepts of the algorithm.

1. Since the chamber operation is the most critical stage in the system, the production schedule is constructed with emphasis on the chamber operation and the subsequent operation, i.e. dry-side operation. The schedule is then extended to the operations that precede the chamber operation (wet-side and freezing operations).
2. All products to be processed are divided into four groups based on their vacuum requirement and the type of dumping room needed (Table 3.1).

Table 3.1 : Product classification.

D-room required	Vacuum Group	
	1	2 and 3
Dedicated d-room	Group 1 (L_1)	Group 3 (L_3)
General d-room	Group 2 (L_2)	Group 4 (L_4)

These groups are as follows :

a. Group 1 (L_1) consists of products that are in vacuum group 1 and need dedicated dumping rooms. Products in this group can only be processed in chambers 1 to 6 and in dumping room 1 or 2.

b. Group 2 (L_2) consists of products that are in vacuum group 1 and need general dumping rooms. Products in this group can only be processed in chambers 1 to 6 and in dumping rooms 3 to 5.

c. Group 3 (L_3) consists of product that are in vacuum group 2 or 3 and need dedicated dumping rooms. Products in this group can be processed in chamber 1 to 18 and in dumping room 1 or 2.

d. Group 4 (L_4) consists of products that are in vacuum group 2 or 3 and need general dumping rooms. Products in this group can be processed in chamber 1 to 18 and in dumping rooms 3 to 5.

A scalar value representing priority is assigned to each product. Then products in each group are ordered based on the low value of this priority value.

3. Chamber allocation.

A specific product generally consists of multiple batches and as such requires processing on more than one chamber. The purpose of chamber allocation is to determine how many chambers should be assigned to process a product. It should be pointed out that the number of chambers assigned to a product has to be an even number because different products cannot be assigned to tandem chambers. The procedure for allocating chambers is as follows :

(a) Chamber allocation for products in groups 1 and 3 requiring dedicated dumping rooms :

For each product in these two groups, calculate the number of chambers required for continuous processing of all batches of the product. The total number of chambers allocated at any one time to each group is the sum of number of chambers required by each product in that group. For group 1, this value cannot exceed six chambers because only six chambers can be used to process these products. Chambers allocated to group 1 are from chambers 1 through 6, whereas chambers allocated to group 3 are from chambers 7 through 18.

(b) Chamber allocation for products in groups 2 and 4 : The rest of chambers are allocated to group 2 and 4. It should be noted that products in these groups need general dumping rooms. The critical consideration here is to determine the number of types of products from each group to be processed simultaneously in chambers. Let n be the number of types of products from group 2 and m be the number of types of products from group 4. If the total batches for (m + n) products is small, the chambers might be underutilized. However, if the number of batches for (m + n) products is large, the change over time in dumping rooms will increase and may cause blockage in the chambers. Thus, the values of n and m have to be carefully selected.

The allocation of chambers is dynamic, that is, it may change over time. For example, if all products in a particular group are processed, the chambers that were previously allocated to that group can be used or allocated to another group of products.

4. General dumping room allocation.

   There is no problem in allocating dumping rooms 1 and 2 because these are dedicated to certain products. However, general dumping rooms (dumping rooms 3 through 5) have to be shared by (m + n) products as explained earlier. The allocation of dumping rooms to each product depends on the number of chambers used or allocated to that product. This allocation of dumping room is also dynamic in nature. Dumping room allocation to a product may change if the number of chambers used by that product changes. For example, a product may initially have 100 batches to be processed that may require allocating eight chambers and all three general dumping rooms. However, as batches are processed, both chamber and dumping room requirements decrease, and the units originally used for this product may be allocated to different products.

3.1.3 Performance measures.

The performance measures used in this study are as follows :

(a) Makespan.

Makespan is the time required for all products to complete processing on all operations.

(b) Chamber utilization and dumping room utilization.

Utilization represents the percentage of time that a chamber or dumping room is busy processing the products during certain period of time (makespan). In this study the utilizations of individual chamber and dumping room are treated separately. Utilizations from individual chambers are then averaged to obtain an aggregate chamber utilization measure. Similarly, individual dumping room utilizations are combined to obtain an overall dumping room utilization measure.

(c) Average blockage of the chambers.

Blockage represents the percentage of time a chamber is blocked during the production period. Similar to utilization, blockage percentages from individual chambers are averaged to obtain an overall measure.

(d) Average cleaning and setup percentage of the dumping rooms.

Cleaning and setup represents the percentage of time a dumping room is being cleaned and setup during the production period. Cleaning and setup percentages from individual dumping rooms are averaged to obtain an overall measure.

(e) Average time in system.

Time in the system is the time spent to complete all operations for individual products in the system.

(f) Number of tardy products and the average of tardiness. Actually there are no due dates assigned to the products but since the production schedule is developed on a weekly basis it is desired that all products are completed by the end of the week. In this study, a product is considered a tardy product if by the end of the week all operations for the product are not yet finished. The tardiness of the products are recorded and then averaged.

3.1.4 Parameters.

This section explains two parameters in the scheduling algorithm that could affect the system performance, that is, priority rules and number of types of products to be processed simultaneously.

1. Priority rules.

   These rules are used to assign priority to individual products in each group. There are two priority rules used in this study.

   (a) Shortest Chamber Processing Time (SCPT). The chamber processing time, p(c), for a product is given by

   $$p(c) = \text{(number of batches)} \times \text{(chamber processing time per batch)}$$

   Product with smallest chamber processing time is given processing priority.

(b) Shortest Chamber and Dry-side Processing Time (SCDPT). The chamber and dry-side processing time, p(cd), of a product is given by

p(cd) = (number of batches) x (sum of chamber and dry-side processing times per batch)

Product with smallest chamber and dry-side processing time is given processing priority.

2. Number of types of products to be processed simultaneously.

Several values of n (number of types of products from group 2) and m (types of products from group 4) are investigated.

3.2 Description of the Heuristic Scheduling Algorithm

This section describes the scheduling algorithm.

3.2.1 Notation.

The notation used in the algorithm is as follows :

- nb_i = number of batches of product i
- p_{i1} = wet-side processing time per batch for product i [hours]
- p_{i2} = freezing processing time per batch for product i [hours]
- p_{i3} = chamber processing time per batch for product i [hours]
- p_{i4} = dry-side processing time per batch for product i

[hours]

- $p(c)_i$ = chamber processing time for product i [hours]

  $p(c)_i = nb_i \times p_{i3}$

- $p(cd)_i$ = chamber and dry-side processing time for product i [hours]

  $p(cd)_i = nb_i \times (p_{i3} + p_{i4})$

- s_i = setup time in dry-side operation for product i [hours]

- cl_i = cleaning time in dry-side operation for product i [hours]

- t = number of hours in a week (168 hours)

- L_1 = list of products in group 1

- L_2 = list of products in group 2

- L_3 = list of products in group 3

- L_4 = list of products in group 4

- c_i = number of chambers assigned to product i

- cu_i = number of chambers being used by product i

- C(1) = number of chambers allocated to L_1

- C(2) = number of chambers allocated to L_2

- C(3) = number of chambers allocated to L_3

- C(4) = number of chambers allocated to L_4

3.2.2 Algorithm.

The scheduling algorithm consists of four phases with several steps in each phase.

Phase 1 Divide the set of products in four groups based on their vacuum requirement and type of dumping room required as explained in the previous section.

Phase 2 Initial Chamber Scheduling.

Phase 2.1 Chamber allocation and scheduling (Figure 3.1).

1. Chambers from set C(1), chambers 1 to 6, are allocated to products in L_1 for continuous processing. The number of chambers required by each product is given by :

   c_i = [int (nb_i x p_{i3})]/t

   Number of chambers allocated to L_1 is given by :

   $$C(1) = \sum_{i \in L_1} c_i$$

   It should be noted that C(1) must be less than or equal to six. The rest of chambers (6- C(1)) are available for processing other products. Then, schedule c_i batches of product i to c_i chambers.
2. Step 1 is repeated for products in L_3. Chambers from chambers 7 to 18 are allocated to products in L_3 for continuous processing. The number of chambers allocated to L_3 equals C(3), where C(3) must be equal to or less than 12. The rest of chambers (12 - C(3)) are available for processing other products. Then, c_i batches of product i are scheduled to c_i chambers.

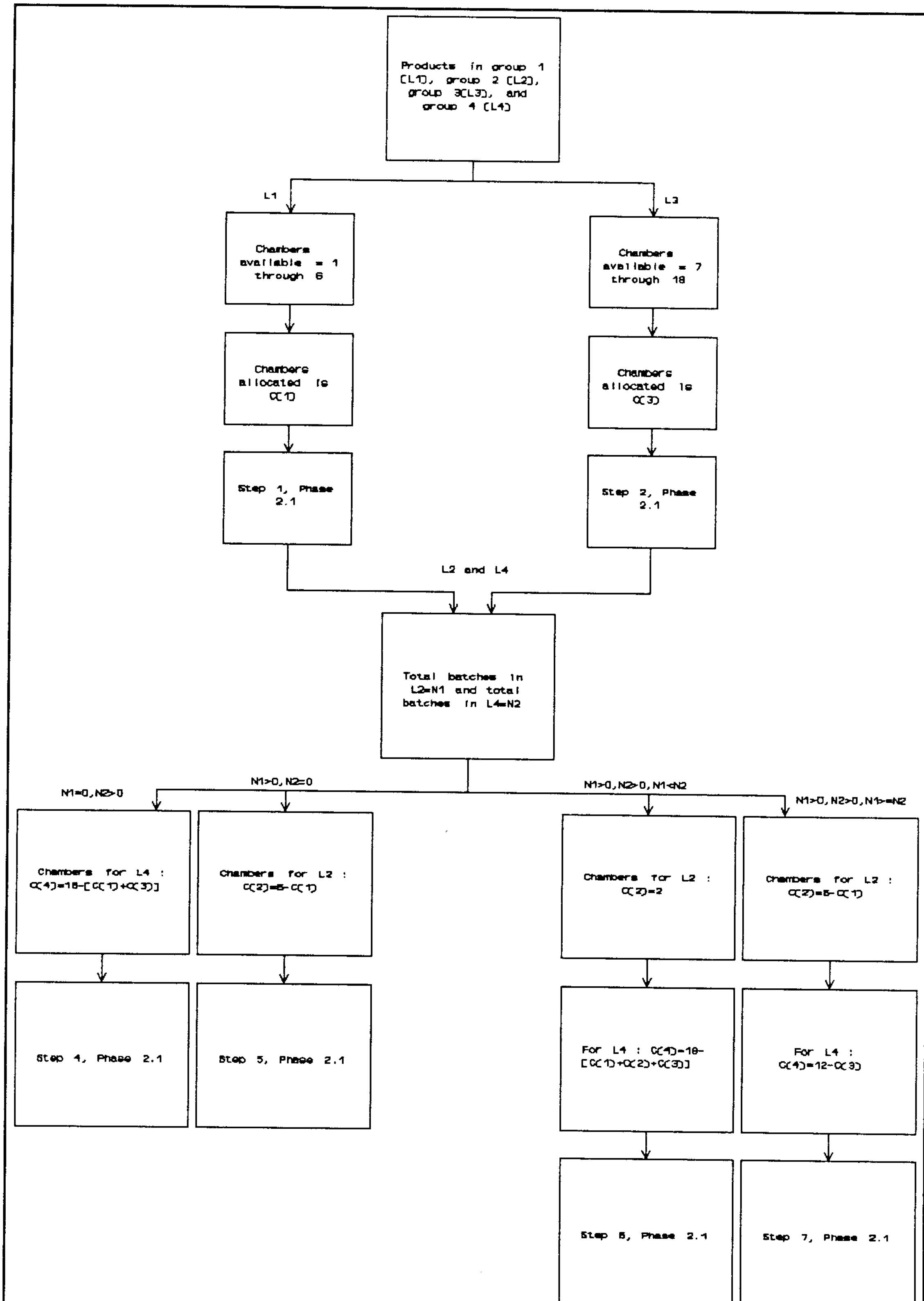

Figure 3.1 : Chamber allocation.

3. Allocation of chambers for products in L_2 and L_4 is based on the total number of batches in each group. Let,

$$N_1 = \sum_{i \in L_2} nb_i$$

$$N_2 = \sum_{i \in L_4} nb_i$$

where :

N_1 = total batches of products in group 2
N_2 = total batches of products in group 4

- If $N_2 > 0$ and $N_1 = 0$, go to step 4 in phase 2.1.
- If $N_1 > 0$ and $N_2 = 0$, go to step 5 in phase 2.1.
- If $N_1 > 0$, $N_2 > 0$, and $N_1 < N_2$, go to step 6 in phase 2.1.
- If $N_1 > 0$, $N_2 > 0$, and $N_1 \geq N_2$, go to step 7 in phase 2.1.
- If $N_1 = 0$ and $N_2 = 0$, go to phase 3.

4. In this case, there is no product in L_2 and since products in L_4 can be processed on any chamber, the number of chambers allocated to this group is given by :

$$C(4) = 18 - [C(1) + C(3)]$$

(a) The first m products are selected from L_4.

(b) The number of chambers allocated for each product in m are calculated based on ratio of their p(c)'s using steps 4(c) through 4(h).

(c) The algorithm computes r(i)'s as the ratios of the p(c)'s for the m products,

$$r(i) = \frac{p(c)_i}{\sum_{i \in L_4} p(c)_i}$$

(d) The proportion of chambers allocated to each product is then calculated based on r(i)'s and C(2) as :

$pc_i = r(i) \times C(4)$

(e) The algorithm starts with product with minimum {r(i)}.

(f) The chambers allocated to product i is given by

$c_i = \text{int}\ [pc_i]$,

where c_i should be an even number and no more than the number of chambers available.

(g) c_i batches of product i are scheduled to c_i chambers.

(h) The number of chambers available are updated, and then the algorithm moves to the product with the next min {r(i)}. Go to step 4f and repeat for all m products.

(i) Go to phase 2.2.

5. In this case, there is no product in L_4. Products in L_2 can only be processed in chambers 1 to 6. Thus, the number of available chambers is given by

$C(2) = 6 - C(1)$

If $C(2) = 2$, then only one type of product in L_2 can be processed in chambers. If C(2) > 2, the first n products are selected from L_2 and chambers allocated to each product in n using steps 4c through 4h above. Then go to phase 2.2.

6(a) Products in L_2 : Only two chambers are allocated to this group (i.e., C(2) equals two). This means that only one type of product in L_2 can be processed (n = 1). Two batches of this product are scheduled in two chambers. Go to phase 2.2.

(b) Products in L_4 : The number of chambers allocated to this group is given by

$$C(4) = 18 - [C(1) + C(2) + C(3)]$$

The first m products from L_4 are selected and allocated to chambers to be used by each product in m using steps 4c through 4h above. Go to phase 2.2.

7(a) Products in L_2 :

The number of chambers allocated to this group is given by :

$$C(2) = 6 - C(1)$$

n products from L_2 are selected and chambers are allocated to each product in n using steps 4c through 4h above. Then go to phase 2.2.

(b) Products in L_4 : The number of chambers allocated to this group is given by

$$C(4) = 12 - C(3)$$

m products from L_4 are selected and chambers are allocated to each product in m using steps 4c through 4h. Then, go to phase 2.2.

Phase 2.2 Initial dumping room allocation.

This procedure is used only for products that need general dumping rooms. Three general dumping rooms have to be assigned to (m + n) products.

1. Total number of chambers being used is given by

$$totcu = \sum_{i=1}^{m+n} cu_i$$

2. The first step is to calculate the ratio of chambers being used by each product, rd(i), as follows :

$$rd(i) = cu_i / totcu$$

3. If the maximum value of all rd(i)'s is one, it means that there is only one type of product being processed in chambers. Therefore, all three dumping rooms are assigned to this product type.
4. If $[0.667 \le \max\{rd(i)\} < 1]$, then two dumping rooms are assigned to product i. The third dumping room is shared by other (n + m - 1) types of products on FCFS (First-Come-First-Serve) basis.
5. If $[0.333 \le \max\{rd(i)\} < 0.667]$, then one dumping room is assigned to product i. Furthermore, the next maximum value of rd(i) will be checked. If $[0.333 \le next\ \max\{rd(i)\} < 0.667]$, then one dumping room is assigned to the next product i and the other dumping room is shared by other (n + m - 2) product types on FCFS basis. However, if the next

maximum value of rd(i) is less than 0.333, then two dumping rooms are shared by (n + m - 1) product types on FCFS basis.

6. If [max {rd(i)} < 0.333], then the three dumping rooms are shared by all product types on FCFS basis.

Phase 3 When one or more chambers finish operation, the batches of products are then processed in dry-side operation performed in dumping rooms. The chambers becoming idle are then available for processing new batches of products. This phase is conducted every time one or more chambers become available.

Phase 3.1 Dry-side scheduling.

After completing processing in chambers, a batch is processed in a dumping room.

1. If only one dumping room is assigned to the product type and that dumping room is available, then the batch released from the chamber is scheduled in the dumping room. However, if the dumping room is not available, the batch cannot be released from chamber. Consequently, the chamber is blocked until the dumping room becomes available.
2. If two or more dumping rooms are assigned to the product type, then the algorithm tries to find a dumping room which is currently processing the same

product type, so as to avoid cleaning and setup times. If such a dumping room is available and idle, then the batch is released from the chamber and scheduled in the dumping room. Otherwise, a search is initiated to find an idle dumping room that can start processing the batch. If all dumping rooms are busy, the chamber is blocked until a dumping room becomes available.

Phase 3.1 is repeated for all batches that finish processing in chambers.

Phase 3.2 Chamber scheduling.

Once a chamber has released its batch to the dumping rooms, it is available to process new batch of product.

1. The algorithm starts with the idle chambers allocated to L_1.
2. In general, either a single chamber may have finished processing, or both chambers may have finished processing simultaneously. The single idle chambers are separated from idle chambers in tandem.
3. For single idle chambers : If there are still some batches of the just completed product that have to be processed, then a batch of this product is scheduled in the chamber. Otherwise, the chamber is left idle. Step 3 is repeated for all single idle

chamber.

4. For idle chambers in tandem : If there are still some batches of the just completed product that have to be processed, then two batches of this product are scheduled to the pair of chambers. If there are no more batches to be processed, then a check is made whether there are other chambers that are still processing batches from that product type. If there are no such chambers, it means that the product is completely processed, and a new product is selected from group L_1 and two batches of this new product are scheduled to the pair of chambers. Otherwise, the pair of chambers is left idle. Step 4 is repeated for all idle chamber in tandem.

5. If some pairs of chambers are still idle, then the algorithm looks for a product type currently being processed which has the largest number of batches waiting to be processed. If there is such a product, two batches of it are scheduled to the pair of chambers. Otherwise, a new product is selected from L_1 and two batches of the product are scheduled in the chambers. If there are no more new products in L_1, the idle pair of chambers is allocated to groups L_2 or L_4. This allocation depends on which group has larger total number of

batches still to be processed.

6. Step 2 through 5 are repeated for idle chambers allocated to L_3. Then, go to step 7.
7. Idle chambers that are allocated to L_2 are examined using the following steps :

(a) A check is made if there is a product that has completely been processed. If the condition is true, a new product is selected from L_2 so that the value of n or m remain constant.

(b) Reallocation of chambers to products is accomplished by recalculating c_i based on the ratio of their remaining chambers' processing times using algorithm in steps 4c through 4f of Phase 2.1. Note that the new value of c_i might be different from c_i obtained in previous computations.

(c) The products are scheduled to the allocated chambers. If the number of chambers being used by the product (cu_i) is greater than the number of chambers assigned in step 7b (c_i), then additional batches are not scheduled to the chambers. Else, batches are scheduled to chambers until cu_i is equal to c_i.

8. Repeat steps 7a through 7c for idle chambers that are allocated to L_4.

Phase 3.3 Dumping room reallocation.

Reassign dumping rooms based on the new chamber allocation by using steps in phase 2.2.

Phase 4 Wet-side and freezing scheduling.

From the previous two phases, the starting time of each chamber operation can be determined. Since the move (transportation) time between stages is negligible and the machines in the wet-side operation and spaces in cold room are always available, then the starting and the completion time in the first two stages (wet-side and freezing) can be determined as follows :

1. The completion time of freezing operation (TC_2) is equal to the starting time of chamber operation (TS_3).
2. The starting time of the freezing operation, TS_2, is given by

$$TS_2 = TC_2 - p_{i2}$$

3. The completion time of wet-side operation (TC_1) is equal to the starting time of freezing operation (TS_2).
4. The starting time of wet-side operation, TS_1, is given by

$$TS_1 = TC_1 - p_{i1}$$

CHAPTER 4. IMPLEMENTATION

4.1 Computer Model

The scheduling system was developed in SIMAN simulation language with FORTRAN interaction (Pegden [1989]). The FORTRAN component was used for modeling the algorithm and generating the results. The SIMAN was primarily used for controlling the parameters. The computer model has been run on a 286 - or higher IBM-compatible microcomputer.

The flow diagram in Figure 4.1 shows the organization of the program. Basically, it follows the algorithm described in the previous chapter. The program starts with data input to the program and initializing system variables. This is followed by initial chamber allocation and scheduling, wet-side and freezing scheduling, and initial dumping room allocation.

Following the identification of chamber that becomes idle, dumping room scheduling for the current product is performed. If there are no more batches or products to be processed, the program sets the current chambers idles, and waits for the next chamber to finish processing. Otherwise, chamber reallocation is performed and a new batch or product is scheduled in chambers followed by wet-side and freezing scheduling and dumping room reallocation.

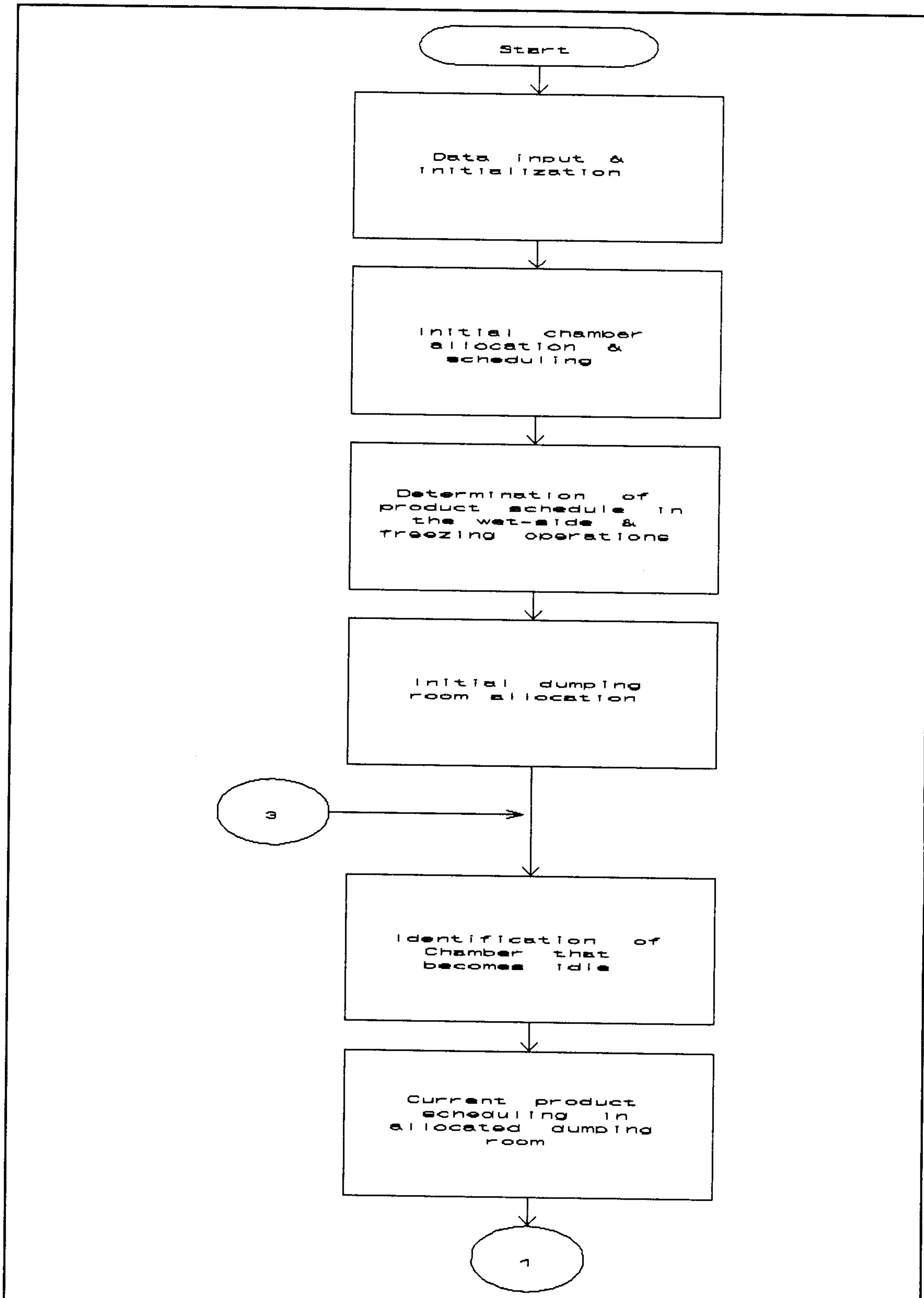

Figure 4.1 : The organization of the program.

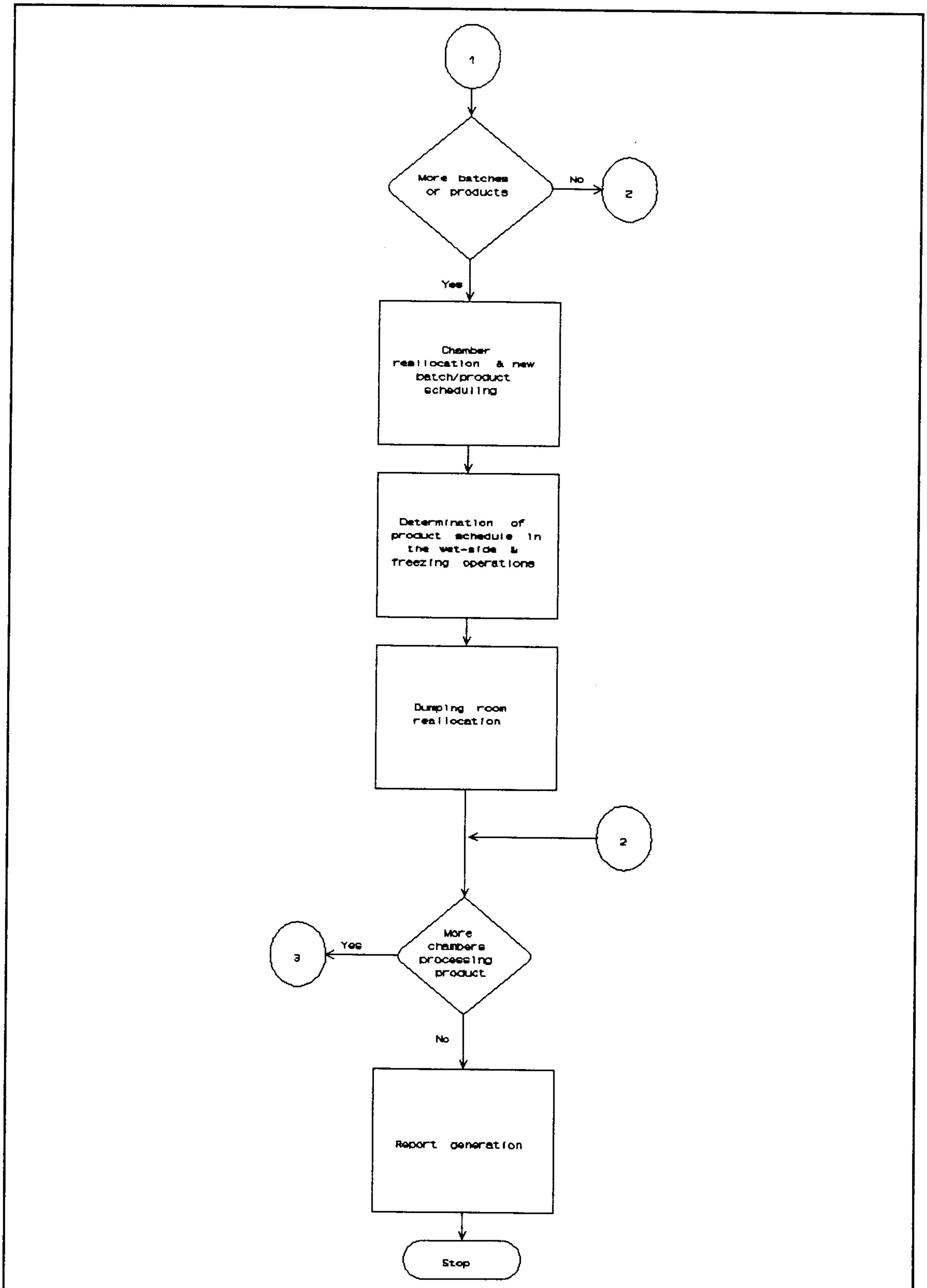

Figure 4.1 : continued.

The program continues until there are no more batches to be processed and no more chambers processing products. The last step in the process is report generation. The program code for the computer model is given in Appendices IA and IB.

4.2 **Input Requirements**

The input to the computer model is a set of products that have to be processed. The data required for each product are as follows :

1. Product number.
2. Product name.
3. Number of batches of the product.
4. Wet-side processing time per batch (hours).
5. Freezing time per batch (hours).
6. Chamber processing time per batch (hours).
7. Vacuum group of the product.
8. Chamber stagger time (hours).
9. Setup time in dumping rooms (hours).
10. Cleaning time in dumping rooms (hours).
11. Dry-side processing time per batch (hours).
12. Type of dumping room required.

4.3 **Output Features**

The report from the computer model includes a production schedule and summary of the system performance measures. The production schedule specifies the starting and completion times at each of the four stages as well as the specific chamber and dumping room numbers where the product is processed.

The performance measures reported are makespan, utilizations and blockage percentage of the chambers, utilization and setup-cleaning percentage of the dumping room, time in system, average of tardiness, and the number of tardy products.

4.4 **Application of Computer Model**

This section illustrates the application of the system and the tradeoffs involved in the selection of the two independent factors - priority rules and values of m and n. The data used in the example has been adapted from the actual production data at OFD, and is shown in Table 4.1.

The data in Table 4.1 is presented in the format used by the computer model. Included in input for each product are : number of batches to be processed, processing time at each stage, vacuum group (i.e., vacuum requirements) in the chambers, setup and cleaning times for dumping rooms, and type of dumping room required.

The two priority rules built in the system are SCPT (Shortest Chamber Processing Time) and SCDPT (Shortest Chamber and Dry-side Processing Time). The values of m and n can be specified by the user. Three values of (m + n) used in this example are 1, 2, and 3. These values were selected to

Table 4.1 : Input data for computer model.

Prod. nmbr	Product name	Nmbr of batches	Wetside proc time	Freezing proc time	Chambers proc time	Chambers vac grp	Chambers stag time	Dry-side setup time	Dry-side clean time	Dry-side proc time	Dry-side type rqrd
1	MRE Pears	40	1.5	8.0	17.0	1	6.0	2.0	6.0	0.75	1
2	Taco sauce	5	3.0	8.0	31.0	3	12.0	1.0	4.0	1.0	3
3	Red raspberries	1	2.0	8.0	20.0	3	4.0	1.0	4.0	0.75	3
4	MH chili	1	2.0	8.0	18.0	3	6.0	1.0	4.0	0.75	3
5	MH chicken stew	2	4.0	0.0	19.0	3	6.0	1.0	4.0	0.75	3
6	Crushed pineapple	6	16.0	8.0	30.0	3	12.0	1.0	6.0	1.0	3
7	N/S chix patties	10	20.0	0.0	14.0	3	2.0	1.0	4.0	2.0	2
8	N/S hbrgr patties	8	16.0	8.0	8.0	3	2.0	1.0	4.0	2.0	2
9	N/S sausage patties	1	2.0	8.0	6.0	3	2.0	1.0	4.0	3.0	2
10	Chpd broccoli w/soy	3	24.0	8.0	18.0	2	3.0	1.0	4.0	0.75	3
11	Pea pods	2	8.0	8.0	16.0	2	2.0	1.0	4.0	0.75	3
12	N/S ground beef	4	10.0	8.0	6.0	2	1.0	1.0	4.0	0.5	3
13	Nissin lobster	3	14.0	8.0	14.0	2	3.0	1.0	4.0	0.75	3
14	Blueberries	1	4.0	8.0	15.0	2	3.0	1.0	4.0	0.75	3
15	Frozen grd orange	2	8.0	8.0	21.0	2	4.0	1.0	6.0	1.0	3
16	Nissin chix w/soy	11	24.0	0.0	8.0	3	2.0	1.0	4.0	0.75	3
17	Niss chix w/herb	4	16.0	0.0	8.0	3	2.0	1.0	4.0	0.75	3

Note :

- Processing, setup, cleaning, and stagger times are in hours/batch
- Dumping room type required in dry-side operation :
  - 1 - Dumping room dedicated to meal-ready-to-eat (MRE).
  - 2 - Dumping room dedicated to patties.
  - 3 - General dumping room.

identify and illustrate the tradeoffs involved in throughput times and resources utilization.

The combination of priority rule and specific value of (m + n) is referred to as a scheduling policy. Thus, there are six policies evaluated in this study as shown in Table 4.2.

Table 4.2 : Six scheduling policies.

Policy	Rule	m + n
1	SCPT	1 + 0 = 1
2	SCPT	2 + 0 = 2
3	SCPT	3 + 0 = 3
4	SCDPT	1 + 0 = 1
5	SCDPT	2 + 0 = 2
6	SCDPT	3 + 0 = 3

The computer model produces two sets of outputs. The first output is associated with the processing schedules in the four stages, in particular chambers and dumping rooms. The second output is the system performance report. Figure 4.2 shows the production schedule report for scheduling policy 3 (i.e., SCPT with m + n equal to 3) for the first 50 hours of operation.

The initial table in Figure 4.2 is an echo of input data, specifically, product description, its batch size, vacuum group, and dumping room requirement. This is followed by a table showing the initial chamber allocation along with the start and completion times. Thereafter, every time a chamber completes processing a batch, information concerning product

-- OREGON FREEZE DRY --

PRODUCTS

No.	Product name	Number of batches	Vacuum group	D-room reqrd.
1	MRE Pears	40	1	1
2	Taco sauce	5	3	3
3	Red raspberries	1	3	3
4	MH chili	1	3	3
5	MH chicken stew	2	3	3
6	Crushed pineapple	6	3	3
7	N/S chicken patties	10	3	2
8	N/S hbrgr patties	8	3	2
9	N/S sausage patties	1	3	2
10	Chopped broccoli w/soy	3	2	3
11	Pea pods	2	2	3
12	N/S ground beef	4	2	3
13	Nissin lobster	3	2	3
14	Blueberries	1	2	3
15	Frozen ground orange	2	2	3
16	Nissin chicken w/ soy	11	3	3
17	Niss 3/8" chix w/herb	4	3	3

PRODUCTION SCHEDULE

Scheduling product(s) on chamber(s)

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	.00	1.50	1	9.50	26.50
1	6.00	7.50	2	15.50	32.50
1	.00	1.50	3	9.50	26.50
1	6.00	7.50	4	15.50	32.50
1	.00	1.50	5	9.50	26.50
1	6.00	7.50	6	15.50	32.50
9	.00	2.00	7	10.00	16.00
8	.00	16.00	9	24.00	32.00
8	2.00	18.00	10	26.00	34.00
7	.00	20.00	11	20.00	34.00
7	2.00	22.00	12	22.00	36.00
14	.00	4.00	13	12.00	27.00
4	.00	2.00	15	10.00	28.00
3	.00	2.00	17	10.00	30.00

Chamber(s) finishes operation at time 16.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
9	7	.00	2	16.00	19.00

Figure 4.2 : Production schedule report for scheduling policy 3.

Scheduling product(s) on chamber(s) at time 16.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
7	.00	20.00	7	20.00	34.00
7	2.00	22.00	8	22.00	36.00

Chamber(s) finishes operation at time 26.50

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	1	.00	1	26.50	27.25
1	3	.75	1	27.25	28.00
1	5	1.50	1	28.00	28.75

Scheduling product(s) on chamber(s) at time 26.50

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	17.00	18.50	1	26.50	43.50

Chamber(s) finishes operation at time 27.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
14	13	.00	3	27.00	27.75

Scheduling product(s) on chamber(s) at time 27.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
12	9.00	19.00	13	27.00	33.00
12	10.00	20.00	14	28.00	34.00

Scheduling product(s) on chamber(s) at time 27.25

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	17.75	19.25	3	27.25	44.25

Chamber(s) finishes operation at time 28.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
4	15	.00	5	28.00	28.75

Scheduling product(s) on chamber(s) at time 28.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	18.50	20.00	5	28.00	45.00
12	10.00	20.00	15	28.00	34.00
12	11.00	21.00	16	29.00	35.00

Chamber(s) finishes operation at time 30.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
3	17	3.75	5	33.75	34.50

Chamber(s) finishes operation at time 32.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
8	9	.00	2	32.00	34.00

Figure 4.2 : continued.

Scheduling product(s) on chamber(s) at time 32.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
8	8.00	24.00	9	32.00	40.00

Chamber(s) finishes operation at time 32.50

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	2	.00	1	32.50	33.25
1	4	.75	1	33.25	34.00
1	6	1.50	1	34.00	34.75

Scheduling product(s) on chamber(s) at time 32.50

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	23.00	24.50	2	32.50	49.50

Chamber(s) finishes operation at time 33.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
12	13	.00	4	33.00	33.50

Scheduling product(s) on chamber(s) at time 33.25

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	23.75	25.25	4	33.25	50.25

Scheduling product(s) on chamber(s) at time 33.75

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
11	17.75	25.75	17	33.75	49.75
11	19.75	27.75	18	35.75	51.75

Chamber(s) finishes operation at time 34.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
7	7	5.00	2	39.00	41.00
8	10	12.00	2	46.00	48.00
7	11	19.00	2	53.00	55.00
12	14	.00	4	34.00	34.50
12	15	.00	3	34.00	34.50

Scheduling product(s) on chamber(s) at time 34.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	24.50	26.00	6	34.00	51.00
17	18.00	34.00	13	34.00	42.00
17	20.00	36.00	14	36.00	44.00

Chamber(s) finishes operation at time 35.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
12	16	.00	3	35.00	35.50

Scheduling product(s) on chamber(s) at time 35.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
5	31.00	35.00	15	35.00	54.00
5	37.00	41.00	16	41.00	60.00

Figure 4.2 : continued.

Chamber(s) finishes operation at time 36.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
7	8	19.00	2	55.00	57.00
7	12	21.00	2	57.00	59.00

Scheduling product(s) on chamber(s) at time 39.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
7	19.00	39.00	7	39.00	53.00

Chamber(s) finishes operation at time 40.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
8	9	24.00	2	64.00	66.00

Chamber(s) finishes operation at time 42.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
17	13	.00	4	42.00	42.75

Scheduling product(s) on chamber(s) at time 42.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
17	26.00	42.00	13	42.00	50.00

Chamber(s) finishes operation at time 43.50

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	1	.00	1	43.50	44.25

Scheduling product(s) on chamber(s) at time 43.50

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	34.00	35.50	1	43.50	60.50

Chamber(s) finishes operation at time 44.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
17	14	.00	4	44.00	44.75

Scheduling product(s) on chamber(s) at time 44.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
17	28.00	44.00	14	44.00	52.00

Chamber(s) finishes operation at time 44.25

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	3	.00	1	44.25	45.00

Scheduling product(s) on chamber(s) at time 44.25

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	34.75	36.25	3	44.25	61.25

Figure 4.2 : continued.

Chamber(s) finishes operation at time 45.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	5	.00	1	45.00	45.75

Scheduling product(s) on chamber(s) at time 45.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	35.50	37.00	5	45.00	62.00

Scheduling product(s) on chamber(s) at time 46.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
8	22.00	38.00	10	46.00	54.00

Chamber(s) finishes operation at time 49.50

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	2	.00	1	49.50	50.25

Scheduling product(s) on chamber(s) at time 49.50

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	40.00	41.50	2	49.50	66.50

Chamber(s) finishes operation at time 49.75

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
11	17	.00	5	49.75	50.50

Chamber(s) finishes operation at time 50.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
17	13	.00	4	50.00	50.75

Chamber(s) finishes operation at time 50.25

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	4	.00	1	50.25	51.00

Scheduling product(s) on chamber(s) at time 50.25

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	40.75	42.25	4	50.25	67.25

Chamber(s) finishes operation at time 51.00

Finished Product #	Chamber number	Blocking time	D-room number	Starting time	Completion time
1	6	.00	1	51.00	51.75

Scheduling product(s) on chamber(s) at time 51.00

Prod number	Strt.time at wetside	Strt.time at cold rm	Chamber number	Strt time	Compln time
1	41.50	43.00	6	51.00	68.00

Figure 4.2 : continued.

type, dumping room scheduling, chamber blockage, and the chamber schedule of new batch is printed. Since scheduling of other resources (i.e., wet-side and freezing operation) is organized around chamber operation, decisions related to these resources are also made at times when chamber scheduling is processed. Other information concerning the resources is internally maintained by the system and is summarized in system performance report (Figure 4.3) printed after the weekly production requirements have all been scheduled.

The system performance report in Figure 4.3 is also for scheduling policy 3. The report contains information on a number of output statistics. This includes :

1. Makespan or the total processing time for all products. For the data in Table 4.1, the makespan was 168 hours. Thus, with policy 3, scheduling of all production specified at the start of the week was completed within that week.
2. Chamber performance, including utilization (or proportion busy), blockage, and idle time. Note that the sum of utilization, blockage, and idle time should equal 100%. The average utilization of all chambers was 54%. However, there was a significant amount of variation with minimum utilization being around 32% and maximum about 81%. The utilization of the first six chambers was relatively higher than other chamber because of product number 1 (see Table 4.1) that can only be processed on the first

SUMMARY REPORT

* Makespan : 168.00 hours.

* Chambers.

Chamber number	Utilization percentage	Blockage percentage	Idle-time percentage
1	70.83	.00	29.17
2	70.83	.00	29.17
3	70.83	.45	28.72
4	60.71	.45	38.84
5	70.83	.89	28.27
6	60.71	.89	38.39
7	33.33	33.33	33.33
8	16.67	22.02	61.31
9	36.90	40.48	22.62
10	32.14	35.71	32.14
11	42.86	32.74	24.40
12	34.52	23.21	42.26
13	80.95	.00	19.05
14	67.26	.00	32.74
15	65.48	.00	34.52
16	44.05	.00	55.95
17	75.00	2.23	22.77
18	36.31	.00	63.69

The utilization percentage of chambers.
Average : 53.90
Standard deviation : 19.10
The blockage percentage of chambers.
Average : 10.69
Standard deviation : 15.47

* Dumping rooms.

D-room number	Utilization percentage	Clean & Setup percentage	Idle-time percentage
1	17.86	1.19	80.95
2	23.21	39.29	37.50
3	3.72	12.50	83.78
4	11.61	12.50	75.89
5	6.10	12.50	81.40

The utilization percentage of dumping rooms.
Average : 12.50
Standard deviation : 8.10
The cleaning & setup percentage of dumping rooms.
Average : 15.60
Standard deviation : 14.12

Figure 4.3 : System performance report for sheduling policy 3.

* Time in systems.

Product number	Time in system (hours)
1	135.25
2	80.25
3	34.50
4	28.75
5	29.75
6	75.00
7	119.00
8	128.00
9	19.00
10	68.75
11	34.75
12	26.50
13	50.75
14	27.75
15	42.00
16	72.75
17	34.75

The average time in system of products : 77.50 hours.

* **Tardiness.**
  Average tardiness of products : .00
  Number of tardy products : 0

Figure 4.3 : continued.

six chambers (vacuum group 1) and required large set of batches to be processed. The blockage was high for chambers 7 through 12 since these chambers were used by three product types that had to share one dedicated dumping room (dumping room 2). This situation increased the setup and cleaning activities in the dumping room, thus causing the blockage in the chambers.

3. Dumping room performance shows that a substantial portion of the time is spent on cleaning and setup; the proportion of busy time when a product is actually being processed is fairly low. For the set of products analyzed here, this is somewhat obvious from Table 4.1 where cleaning and setup times are very high as compared to processing times and the number of batches for products are fairly low so change over of products in dumping room is quite frequent.

4. Time in system (or throughput time) for each product. This is the total time required to process all batches for a product. For policy 3, the average time in system was 77.50 hours.

5. Average tardiness and number of tardy products. Since all products are completed within a week, there are no tardy products to report.

The information in Figure 4.2 can be used to display information on processing sequence in chambers and dumping rooms. Figure 4.4 shows Gantt charts for chambers 1, 9, and

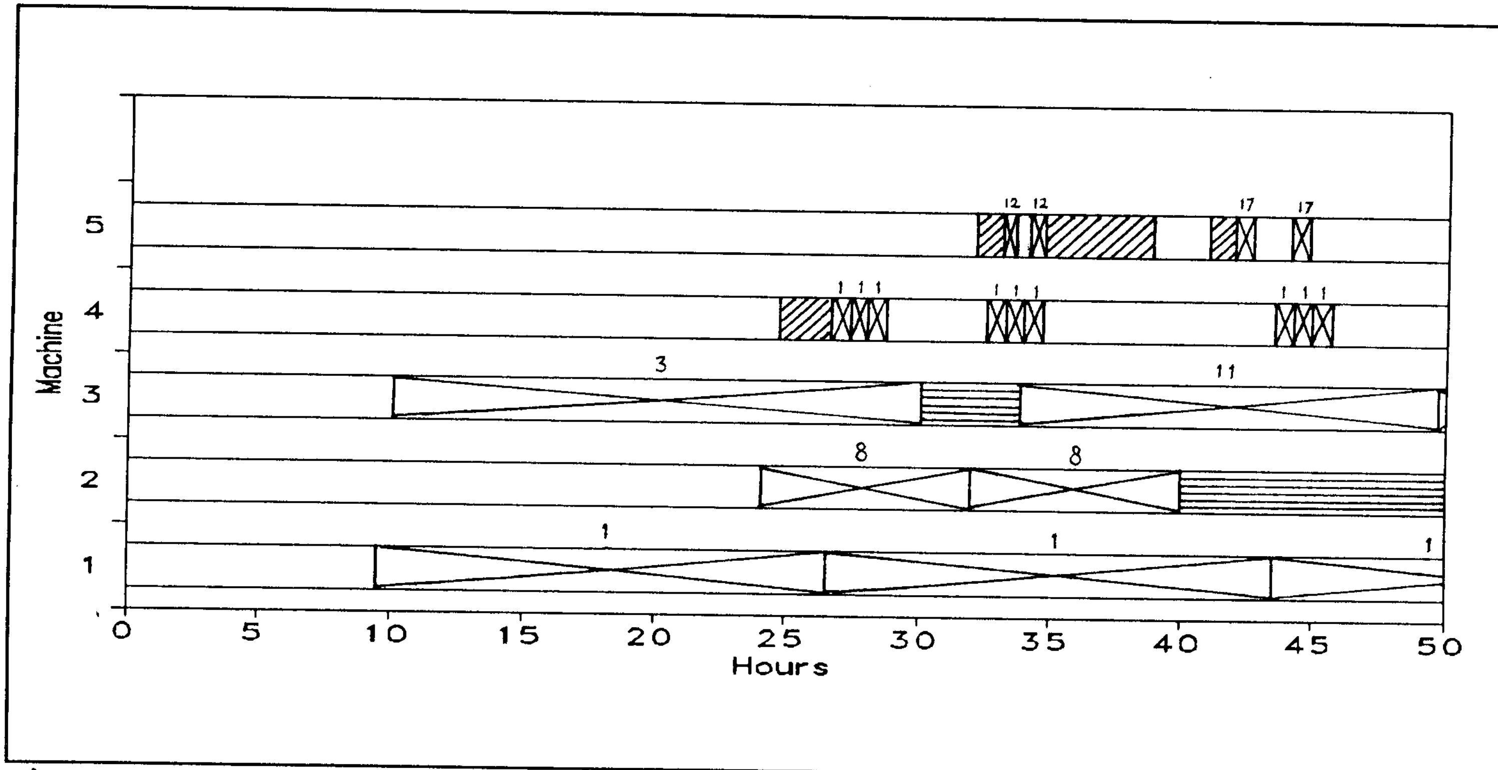

Figure 4.4 : Gantt chart showing processing sequence in chambers and dumping rooms.

<u>Note</u> :

- Machines 1, 2, and 3 represent chambers 1, 9, and 17, respectively.
- Machines 4 and 5 represent dumping rooms 1 and 4, respectively.
- [Idle symbol] : Idle; [Busy symbol] : Busy; [Blocked symbol] : Blocked; [Cleaned & setup symbol] : Cleaned & setup

17, and dumping rooms 1 and 4, for the first 50 hours of operation.

A comparison of the performance measures obtained for the six scheduling policies is presented in Table 4.3 and Figures 4.5 through 4.8. Figure 4.5 shows the graphical comparison of the makespan resulting from six scheduling policies. The average time in system is presented in Figure 4.6. Figures 4.7 and 4.8 show the average tardiness and number of tardy products, respectively. Table 4.3 presents the average as well as the standard deviation of chamber utilizations, chamber blockage percentages, dumping room utilizations, and dumping room cleaning and setup percentages.

Figure 4.5 shows that policies 3 (SCPT rule and (m+n) equals 3) and 6 (SCDPT rule and (m+n) equals 3) give low makespans (168 hours and 167.75 hours, respectively) whereas policies 1 (SCPT rule and (m+n) equals 1) and 4 (SCDPT rule and (m+n) equals 1) result in high makespans (284.75 hours for both policies). Table 4.3 shows that average utilization for all chambers for policies 3 and 6 is high (53.90% and 53.98%, respectively) as compared to that of policies 1 and 4 (31.80%). This indicates that for the set of product analyzed here, because the number of batches for product are small (see Table 4.1), processing more than one type of product simultaneously gives better makespan and chamber utilizations; processing only one type of product results in low utilization and high makespan. On the other hand, if number of batches per product were high, smaller values of (m+n) would be desirable, because large value would result in chamber blockage due to dumping room queues.

Table 4.3 : Performance measures of six scheduling policies.

Measures	Policy					
	1	2	3	4	5	6
Chamber Util						
Average	31.80	50.24	53.90	31.80	50.03	53.98
Std. dev.	28.32	24.79	19.10	28.27	26.10	18.82
Chamber blockage						
Average	6.31	9.86	10.69	6.31	9.83	10.71
Std. dev.	9.29	14.49	15.47	9.29	14.42	15.50
D-room Util						
Average	7.37	11.65	12.50	7.37	11.60	12.52
Std. dev.	4.48	7.21	8.10	4.49	7.11	7.71
D-room cln+set						
Average	12.43	16.31	15.60	12.01	16.91	16.69
Std. dev.	7.99	12.82	14.12	7.98	12.72	14.18

Although dumping room utilizations of policies 3 and 6 are higher than those of policies 1 and 4 (see Table 4.3), policies 3 and 6 require more cleaning and setup activities than do policies 1 and 4. This results in higher chamber blockage and throughput times for policies 3 and 6 as compared to policies 1 and 4 (Figure 4.6).

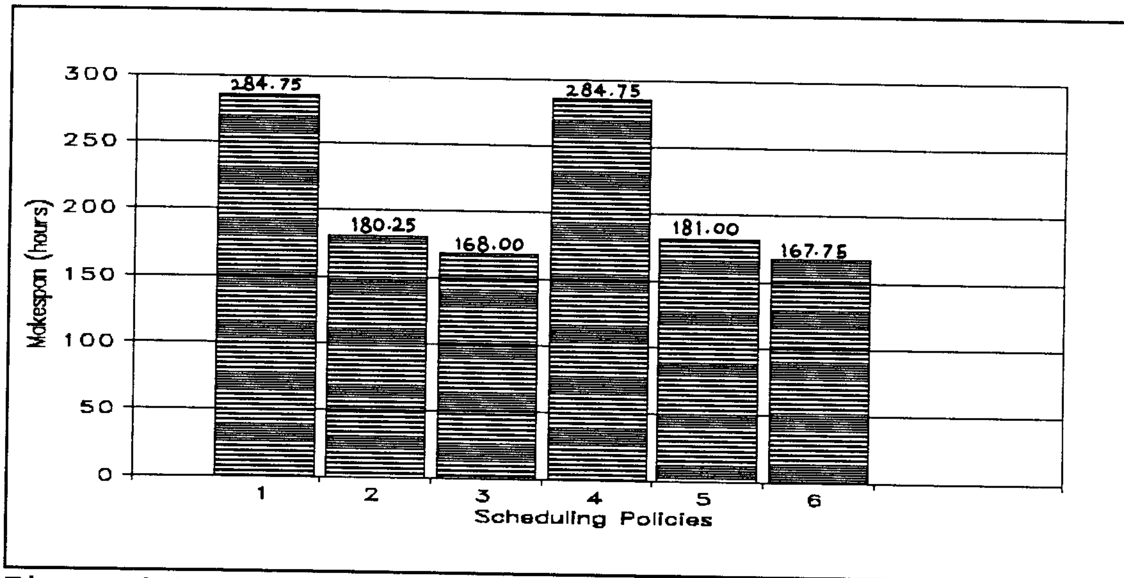

Figure 4.5 : Makespans for six policies.

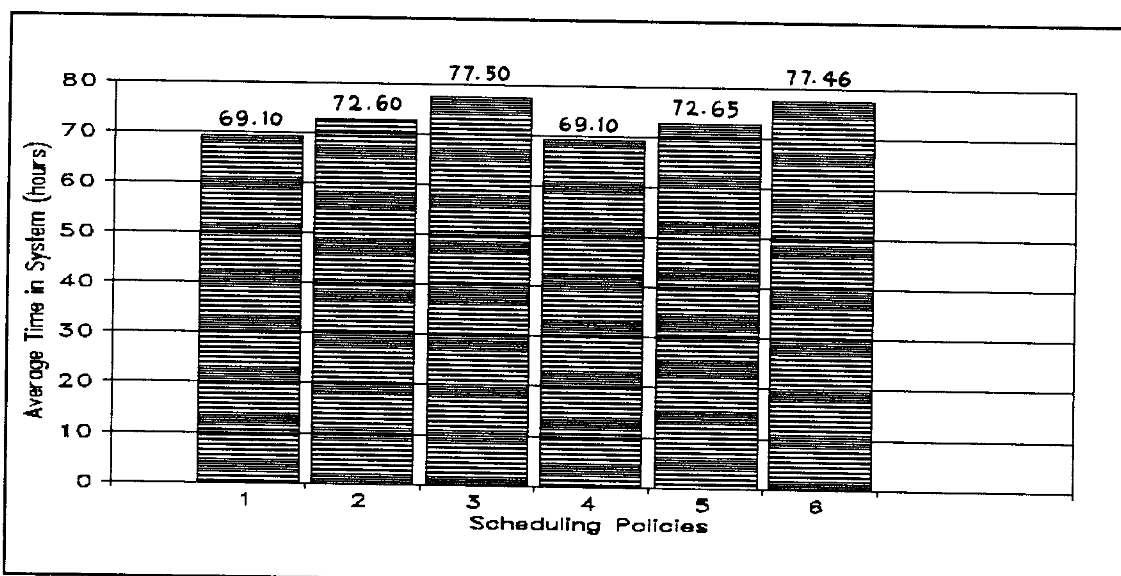

Figure 4.6 : Average times in system for six policies.

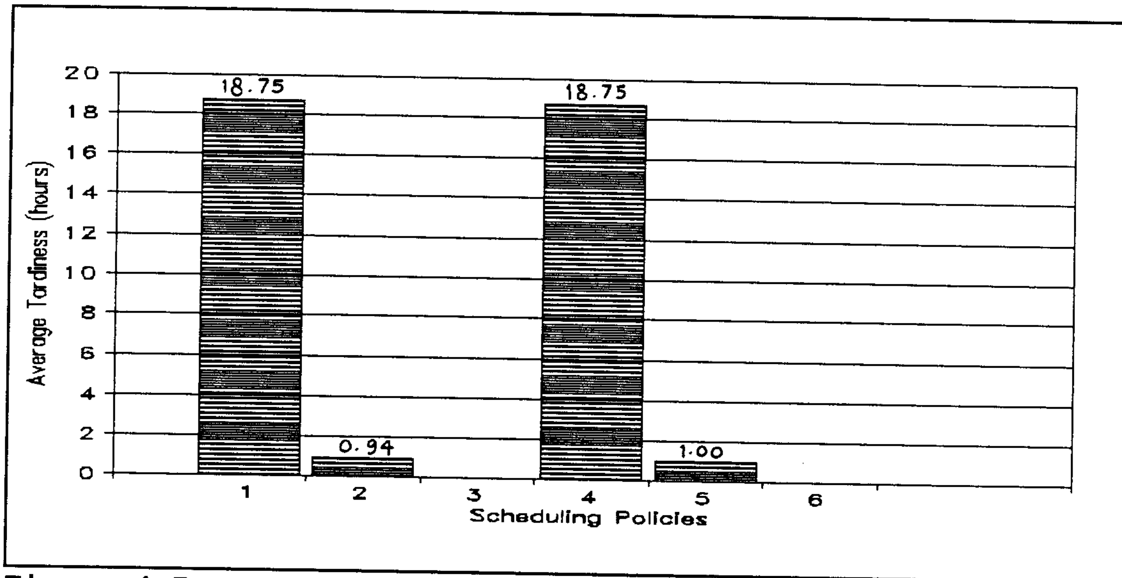

Figure 4.7 : Averages of tardiness for six policies.

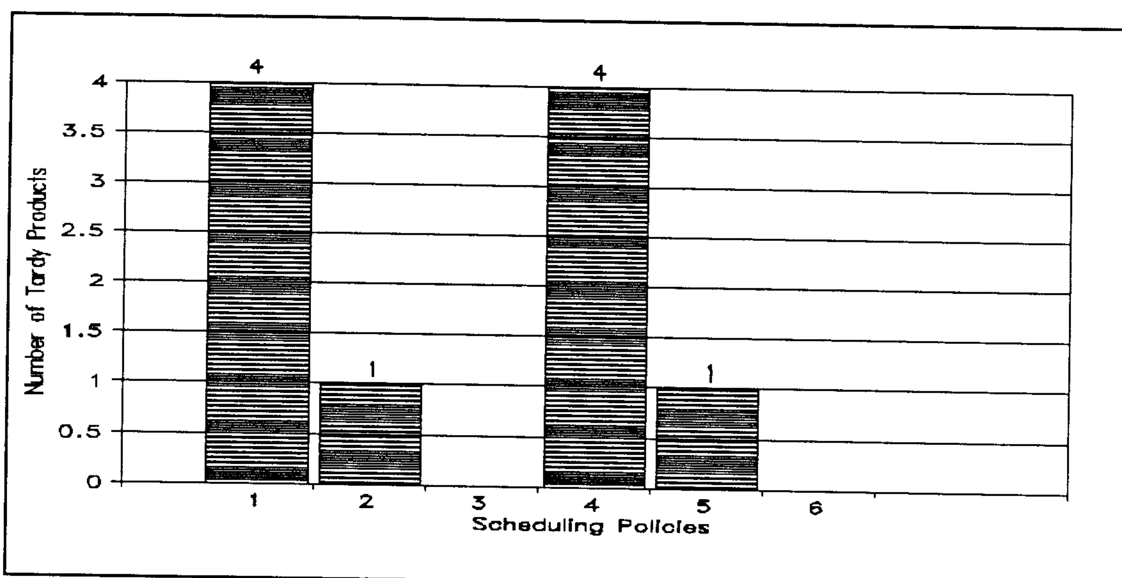

Figure 4.8 : Number of tardy products for six policies.

Makespans of policies 3 and 6 are equal to or less than 168 hours (or one week); therefore there are no tardy products obtained (Figure 4.8). On the other hand, makespans of policies 1 and 4 are higher as compared to other policies.

Policies 1 and 4 give the highest average tardiness (Figure 4.7) and number of tardy products (Figure 4.8).

The decision about which scheduling policy should be used depends on the priorities set by the user. For example, if the user considers the makespan to be of highest priority, scheduling policy 6 gives the best results. This policy also results in highest average chamber utilization and the utilizations of individual chambers are more balanced (see Table 4.3). Besides, there are no tardy products (see Figure 4.8). However, chamber blockage, dumping room cleaning+setup time, and the average time a job spends in the system are high as compared to the other policies (see Table 4.3 and Figure 4.6).

An important feature of the system developed in this research is that it can be used to explore these tradeoffs for given production requirements before production decisions are made on the shop floor.

CHAPTER 5. CONCLUSIONS AND FUTURE RESEARCH

5.1 Conclusions

A heuristic algorithm was developed for scheduling a multistage production system in food processing industry. The algorithm takes into account product characteristics and constraints associated with the resources and system variables.

The algorithm is developed and implemented for a major food manufacturer in Albany, Oregon using the freeze-drying process. The system under study can be categorized as a hybrid system since it involves several stages with a number of machines in each stage. The algorithm provides a production schedule on a weekly basis and a summary of the primary system performance measures.

There are two parameters in the algorithm that affect the system performances. These are the priority rule and the number of types of product that are processed simultaneously. There are two priority rules used in this study, both based on processing time in the most critical stages. The specification of the number of product types processed simultaneously is user-specified; the value will depend on number of products to be processed, number of batches of each product, and product type. The combination of these two parameters yields several scheduling policies; each policy results in a different production schedule and consequently, different system performance. The selection of an appropriate scheduling policy for a particular set of products depends on the product set and priorities of the user.

Since the production schedule along with the system performance measures can be determined in advance, it allows the user to analyze the system before the production takes place. For example, a schedule may show that the set of products to be processed can be completed within a week with relatively low chamber utilizations. In this case, the user may increase the production load by increasing the number of products or batches to be processed. On the other hand, if the makespan exceeds the planning horizon of one week, the number of products or batches need to be reduced (given a fixed resource level). This may require the user to prioritize products so that higher priority item can be first processed; conversely, this priority may be determined by the nature of the raw material, as for example, limited shelf life.

5.2 Directions for Future Research

There are two directions for future research.

One, to investigate alternative scheduling policies and consider additional characteristics not currently included in the model developed here. Example of heuristic that can be investigated would be to include dumping room cleaning and setup times in job prioritization since generally it is a high proportion of dumping room time. Including specific due dates would be a logical extension of this system.

Second, to develop a framework for adapting the system developed here for real time control. This study provides an initial production schedule for the system. Some problems that may arise during operation include mechanical failures and delay in raw materials. Extension and/or modification of the algorithm to accommodate these problems would be a real time application of the system.

BIBLIOGRAPHY

Baker, Kenneth R., "Introduction to Sequencing and Scheduling", John Wiley & Sons, Inc., New York, 1974.

Barker, Jeffrey R. and Graham B. McMahon, "Scheduling the General Job Shop", Management Science v31, 5, May 1985.

Bedworth, David D. and James E. Bailey, "Integrated Production Control Systems : Management, Analysis, and Design", 2nd edition, John Wiley & Sons, New York, 1987.

Buzacott, J.A. and J.G. Shantikumar, "On Approximate Queueing Models of Dynamic Job Shops", Management Science v31, 7, July 1985.

Conner, Jon L., "A Heuristic Method for Solving Jobshop Sequencing Problems", Production & Inventory Management v13, 1, 1st quarter, 1972.

Conway, R.W., "Priority Dispatching and Work-in-Process Inventory in a Job Shop", The Journal of Industrial Engineering v16, 2, 123-130, 1965.

Dar-El, Ezey M., "Job Scheduling - A Systematic Approach", Journal of Manufacturing Engineering Systems 1, 1, 1982.

Day, James E. and Michael P. Hottenstein, "Review of Sequencing Research", Naval Research Logistics Quarterly, 118-146, March 1970.

Dudek, Richard A. and Ottis Foy Teuton, Jr., "Development of M-Stage Decision Rule for Scheduling n Jobs through M Machines", Operations Research v12, 3, May - June 1964.

Elmaghraby, Salah E. and Sung H. Park, "Scheduling Jobs on a Number of Identical Machines", AIIE Transactions, March 1974.

Gelders, Ludo F. and Narayansamy Sambandam, "Four Simple Heuristics for Scheduling a Flow-Shop", Int. J. Prod. Res. v16, 3, 221-231, 1978.

Gere, Jr., William S., "Heuristic in Job Shop Scheduling", Management Science v13, 3, 167-190, 1966.

Gonzales, Teofilo and Sartaj Sahni, "Flowshop and Jobshop Schedules : Complexity and Approximation", Operations Research v26, 1, January-February 1978.

Gutcho, Marcia H., "Freeze-Drying Process for the Food Industry", Noyes Data Corporation, Park Ridge, New Jersey, 1997.

Jackson, James R., "Queues with Dynamic Priority Discipline", Factory Scheduling Conference, Carnegie Institute of Technology, May 10-12, 1961.

Johnson, S.M., "Optimal Two and Three Stage Production Schedules with Setup Times Included", Naval Research Logistics Quarterly 1, 1, 61-68, March 1954.

Mellor, J.D., "Fundamentals of Freeze-Drying", Academic Press, New York, 1978.

Panwalkar, S.S. and Wafik Iskander, "A Survey of Scheduling Rules", Operations Research v25, 1, January-February 1977.

Pegden, Dennis C., "Introduction to SIMAN", Systems Modeling Corporation, Sewickley, Pennsylvania, 1989.

Pritsker, A.A.B., Louis W. Miller, Richard J. Zinkl, "Sequencing n Products Involving m Independent Jobs on m Machines", AIIE Transactions v3, 1, 1971.

Wismer, D.A., "Solution of the Flowshop-Scheduling Problem with No Intermediate Queues", Operations Research v20, 3, May-June 1972.

APPENDICES

Appendix I-A : Program list of computer model.

```
*************************************************************************
*  HEURISTIC SCHEDULING ALGORITHM FOR MULTI-STAGE PRODUCTION SYSTEM    *
*                                                                       *
*                           Cynthia Juwono                              *
*          Industrial and Manufacturing Engineering Department          *
*              Oregon State University, Corvallis, Oregon               *
*                             Winter 1992                               *
*************************************************************************
*                                                                       *
*  Attributes :                                                         *
*    1 -  Product number.                                               *
*    2 -  Number of batches.                                            *
*    3 -  Wet-side processing time  / batch (p1).                       *
*    4 -  Chamber processing time / batch (p3).                         *
*    5 -  Setup time (dry-side operation).                              *
*    6 -  Cleaning time (dry-side operation).                           *
*    7 -  Dry-side processing time / batch (p4).                        *
*    8 -  Vacuum group.                                                 *
*    9 -  Chamber stagger time.                                         *
*   10 -  Type of dumping room needed by products :  1 -  MRE          *
*                                                    2 -  Patties      *
*                                                    3 -  General      *
*   11 -  Total processing time / batch [3 + 4 + 7 + 21].               *
*   12 -  Chamber processing time / product [4 * 2].                    *
*   13 -  Chamber and dry-side processing time / product [(4+7) * 2].  *
*   14 -  Total processing time / product [11 * 2].                     *
*   15 -  Product group number.                                         *
*   16 -  Number of chambers assigned to a product.                     *
*   17 -  Number of chambers being used by a product.                   *
*   18 -  Number of chambers reserved.                                  *
*   19 -  Number of d-rooms assigned  to a product.                     *
*   20 -  D-room used :  1 -  dedicated d-room.                         *
*                        2 -  shared d-room.                            *
*   21 -  Freezing processing time.                                     *
*   22 -  Time at which product starts processing in the system.        *
*   23 -  Time at which product completes processing in the system.     *
*                                                                       *
*  Files :                                                              *
*    7 -  All products.                                                 *
*    1 -  Products in vacuum group 1 that need dedicated d-rooms.       *
*    2 -  Products in vacuum group 1 that need general d-rooms.         *
*    3 -  Products in vacuum groups 2 & 3 that need dedicated d-rooms.*
*    4 -  Products in vacuum groups 2 & 3 that need general d-rooms.   *
*    5 -  Products being processed in chambers.                         *
*    6 -  Finished products.                                            *
*************************************************************************
C
      SUBROUTINE PRIME
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
      DIMENSION DATA(14)
      INTEGER CART,VAC,DRTYPE
      CHARACTER*20 FILENAM
      CHARACTER*25 PRNAME
      CHARACTER*20 OUTFIL
```

```
C
C**   Open data and output files specified by user.
      WRITE(*,'(//5X,A,$)')'Enter data file name      :   '
      READ(*,'(A20)') FILENAM
      WRITE(*,'(5X,A,$)')'Enter number of products :    '
      READ *, NN
      OPEN(8,FILE=FILENAM,STATUS='OLD')
C
      WRITE(*,'(/5X,A,$)')'Enter output file name    :    '
      READ(*,'(A20)') OUTFIL
      OPEN(10,FILE=OUTFIL,STATUS='UNKNOWN')
      WRITE(10,'(20X,A)')'--  OREGON FREEZE DRY  --'
      WRITE(10,'(/2X,A/)')'PRODUCTS'
      WRITE(10,'(2X,A)')
     1' No.          Product                Number of   Vacuum    D-room'
      WRITE(10,'(2X,A)')
     1'               name                    batches    group     reqrd.'
      WRITE(10,'(2X,A)')
     1'-----------------------------------------------------------------'
C
C**   Read data from file, set products' attributes.
      DO 10 I=1,NN
        READ(8,*,END=20)NUM,PRNAME,CART,WET,FRZ,CH,VAC,STAG,SETUP,
     1  CLEAN,DRY,DRTYPE
C
        WRITE(10,'(2X,I3,A,A22,I9,I11,I10)')
     1  NUM,'     ',PRNAME,CART,VAC,DRTYPE
C
        P14 = WET + FRZ + CH + DRY
        P34 = CH + DRY
        P3T = CH * CART
        P34T = P34 * CART
        P14T= P14 * CART
        CALL CREATE(IPROD)
        CALL SETA(IPROD,1,REAL(NUM))
        CALL SETA(IPROD,2,REAL(CART))
        CALL SETA(IPROD,3,WET)
        CALL SETA(IPROD,21,FRZ)
        CALL SETA(IPROD,4,CH)
        CALL SETA(IPROD,5,SETUP)
        CALL SETA(IPROD,6,CLEAN)
        CALL SETA(IPROD,7,DRY)
        CALL SETA(IPROD,8,REAL(VAC))
        CALL SETA(IPROD,9,STAG)
        CALL SETA(IPROD,10,REAL(DRTYPE))
        CALL SETA(IPROD,11,P14)
        CALL SETA(IPROD,12,P3T)
        CALL SETA(IPROD,13,P34T)
        CALL SETA(IPROD,14,P14T)
C
C**   Initialization of KPRDR(,)
        KPRDR(I,1) = NUM
C
C**   Put products into the appropriate files.
        CALL COPY(IPROD,DATA)
        CALL CREATE(LPROD)
        CALL ASSIGN(LPROD,DATA)
        CALL INSERT(LPROD,7)
        IF ((NINT(A(IPROD,8)).EQ.1).AND.
     1  ((NINT(A(IPROD,10)).EQ.1).OR.(NINT(A(IPROD,10)).EQ.2))) THEN
          CALL SETA(IPROD,15,1.0)
          CALL INSERT(IPROD,1)
```

```
        ELSE IF ((NINT(A(IPROD,8)).EQ.1).AND.
     1  (NINT(A(IPROD,10)).EQ.3)) THEN
          CALL SETA(IPROD,15,2.0)
          CALL INSERT(IPROD,2)
        ELSE IF ((NINT(A(IPROD,8)).NE.1).AND.
     1  ((NINT(A(IPROD,10)).EQ.1).OR.(NINT(A(IPROD,10)).EQ.2))) THEN
          CALL SETA(IPROD,15,3.0)
          CALL INSERT(IPROD,3)
        ELSE
          CALL SETA(IPROD,15,4.0)
          CALL INSERT(IPROD,4)
        ENDIF
   10 CONTINUE
   20 CLOSE(8)
C
C**   Initialization of chamber numbers
      DO 25 K = 1,18,2
        NUMBER(K) = 1
        NUMBER(K+1) = 0
   25 CONTINUE
C
      WRITE(10,'(///2X,A//)')'PRODUCTION SCHEDULE'
      CALL INCHSC
      RETURN
      END
C
***********************************************************************
      SUBROUTINE INCHSC
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
      DIMENSION NCART(2)
C
      TIMNOW = 0.0
      WRITE(10,'(2X,A)')'Scheduling product(s) on chamber(s)'
      WRITE(10,'(5X,A)')
     1' Prod     Strt.time   Strt.time        Chamber   Strt    Compln'
      WRITE(10,'(5X,A)')
     1'number    at wetside  at cold rm       number    time     time'
C
C******    CHAMBER ALLOCATION FOR PRODUCTS IN FILE 1     *******
C
C**   Reserve some chambers (#1 - #6) for products in file 1.
      NC1 = 6
      N1STRT = 1
  100 IF (NC1.LE.0) GO TO 125
      IF (NQ(1).LE.0) GO TO 125
      CALL RESCH(1,NC1,N1STRT)
      GO TO 100
  125 NC1R = 6 - NC1
C
C******    CHAMBER ALLOCATION FOR PRODUCTS IN FILE 3     *******
C
C**   Reserve some chambers (#7 - #18) for products in file 3.
      NC2 = 12
      N2STRT = 7
  140 IF (NC2.LE.0) GO TO 200
      IF (NQ(3).LE.0) GO TO 200
      CALL RESCH(3,NC2,N2STRT)
```

```
      GO TO 140
  200 NC2R = 12 - NC2
C
C******   CHAMBER ALLOCATION FOR PRODUCTS IN FILES 2 & 4    ******
C
C**   Calculate total number of batches in each file.
      DO 210 M=2,4,2
        IF (NQ(M).LE.0) GO TO 210
        DO 220 I = 1,NQ(M)
          IF (I.EQ.1) THEN
            IPROD = LFR(M)
          ELSE
            IPROD = LSUCC(IPROD)
          ENDIF
          NCART(M) = NCART(M) + NINT(A(IPROD,2))
  220   CONTINUE
  210 CONTINUE
C
      IF ((NCART(2).LE.0).AND.(NCART(4).LE.0)) THEN
        GO TO 667
      ELSE IF (NCART(2).LE.0) THEN
        GO TO 300
      ELSE IF (NCART(4).LE.0) THEN
        GO TO 400
      ELSE IF (NCART(2).LT.NCART(4)) THEN
        GO TO 500
      ELSE
        GO TO 600
      ENDIF
C
C**   NCART(2) = 0; NCART(4) > 0
C
C**   Use all the unreserved chambers (#1 - #18) for products in file 4
C**   Check the availability of the chambers.
C**   If only two chambers available, schedule 1st product in file 4.
C**   Otherwise, schedule INPR2 products.
  300 NC2ASS = NC1 + NC2
      NC2 = NC2ASS
      NC1 = 0
      NC1ASS = 0
      DO 304 K = 1,18
        IF (NGROUP(K).EQ.0) THEN
          NGROUP(K) = 4
        ENDIF
  304 CONTINUE
C
      IF (NC2ASS.LE.0) THEN
        GO TO 667
      ELSE IF (NC2ASS.LE.2) THEN
        GO TO 390
      ELSE
        GO TO 395
      ENDIF
C
  390 INPR2 = 1
      NPROD2 = INPR2
      CALL CHASSN(1,4,NC2,NC2ASS)
      GO TO 397
C
  395 WRITE(*,'(5X,A,$)')'Enter value of m  :    '
      READ *, INPR2
      NPROD2 = INPR2
```

```
      IF ((NPROD2.GT.NQ(4)).OR.(NPROD2.GT.(NC2/2))) THEN
        WRITE(*,'(//5X,A)')'****** DECREASE THE VALUE OF M ******'
        WRITE(*,'(5X,A)')'*************************************'
        WRITE(*,'(5X,A)')'*************************************'
        RETURN
      ENDIF
      CALL CHASSN(NPROD2,4,NC2,NC2ASS)
C
  397 IF (NBR.GT.0) THEN
        CALL DRASSN
      ENDIF
      GO TO 667
C
C**   NCART(2) > 0; NCART(4) = 0
C
C**   Use the unreserved chambers (#1 - #6) for products in file 2.
C**   Check the availability of the chambers.
C**   If only two chambers available, schedule 1st product in file 2.
C**   Otherwise, schedule INPR1 products.
  400 NC1ASS = NC1
      NC2ASS = 0
      DO 404 K = 1,6
        IF (NGROUP(K).EQ.0) THEN
          NGROUP(K) = 2
        ENDIF
  404 CONTINUE
C
      IF (NC1ASS.LE.0) THEN
        GO TO 667
      ELSE IF (NC1ASS.LE.2) THEN
        GO TO 410
      ELSE
        GO TO 430
      ENDIF
C
  410 INPR1 = 1
      NPROD1 = INPR1
      CALL CHASSN(1,2,NC1,NC1ASS)
      GO TO 435
C
  430 WRITE(*,'(5X,A,$)')'Enter value of n  :   '
      READ *, INPR1
      NPROD1 = INPR1
      IF ((NPROD1.GT.NQ(2)).OR.(NPROD1.GT.(NC1/2))) THEN
        WRITE(*,'(//5X,A)')'****** DECREASE THE VALUE OF N ******'
        WRITE(*,'(5X,A)')'*************************************'
        WRITE(*,'(5X,A)')'*************************************'
        RETURN
      ENDIF
      CALL CHASSN(NPROD1,2,NC1,NC1ASS)
C
  435 IF (NBR.GT.0) THEN
        CALL DRASSN
      ENDIF
      GO TO 667
C
C**   NCART(2) < NCART(4)
C
C**   Check the availability of chambers (#1 - #6).
  500 IF (NC1.LE.0) THEN
        NC1ASS = 0
        GO TO 530
```

```
      ELSE
        GO TO 510
      ENDIF
C
C**   Schedule only a product in file 2 to two chambers of NC1.
  510 NC1ASS = 2
      INPR1 = 1
      NPROD1 = INPR1
      DO 514 K = NC1R+1, NC1R+2
        NGROUP(K) = 2
  514 CONTINUE
      CALL CHASSN(1,2,NC1,NC1ASS)
C
C**   Use the rest of chambers for products in file 4.
C**   Check the availability of NC2ASS.
C**   If only two chambers available, schedule 1st product in file 4.
C**   Otherwise, schedule INPR2 products.
  530 NC2ASS = NC1 + NC2
      NC2 = NC2ASS
      DO 532 K = 1,18
        IF (NGROUP(K).EQ.0) THEN
          NGROUP(K) = 4
        ENDIF
  532 CONTINUE
C
      IF (NC2ASS.LE.0) THEN
        GO TO 555
      ELSE IF (NC2ASS.LE.2) THEN
        GO TO 540
      ELSE
        GO TO 550
      ENDIF
C
  540 INPR2 = 1
      NPROD2 = INPR2
      CALL CHASSN(1,4,NC2,NC2ASS)
      GO TO 555
C
  550 WRITE(*,'(5X,A,$)')'Enter value of m  :    '
      READ *, INPR2
      NPROD2 = INPR2
      IF ((NPROD2.GT.NQ(4)).OR.(NPROD2.GT.(NC2/2))) THEN
        WRITE(*,'(//5X,A)')'****** DECREASE THE VALUE OF M ******'
        WRITE(*,'(5X,A)')'**************************************'
        WRITE(*,'(5X,A)')'**************************************'
        RETURN
      ENDIF
      CALL CHASSN(NPROD2,4,NC2,NC2ASS)
C
  555 IF (NBR.GT.0) THEN
        CALL DRASSN
      ENDIF
      GO TO 667
C
C**   NCART(2) >=  NCART(4)
C
C**   Use the unreserved chambers (#1 - #6) for products in file 2.
C**   If only 2 chambers available, schedule 1st product in file 2
C**   Otherwise, schedule INPR1 products.
  600 NC1ASS = NC1
      DO 604 K = 1,6
        IF (NGROUP(K).EQ.0) THEN
```

```
            NGROUP(K) = 2
          ENDIF
  604 CONTINUE
C
      IF (NC1ASS.LE.0) THEN
        GO TO 640
      ELSE IF (NC1ASS.LE.2) THEN
        GO TO 610
      ELSE
        GO TO 630
      ENDIF
C
  610 INPR1 = 1
      NPROD1 = INPR1
      CALL CHASSN(1,2,NC1,NC1ASS)
      GO TO 640
C
  630 WRITE(*,'(5X,A,$)')'Enter value of n  :   '
      READ *, INPR1
      NPROD1 = INPR1
      IF ((NPROD1.GT.NQ(2)).OR.(NPROD1.GT.(NC1/2))) THEN
        WRITE(*,'(//5X,A)')'****** DECREASE THE VALUE OF N ******'
        WRITE(*,'(5X,A)')'*************************************'
        WRITE(*,'(5X,A)')'*************************************'
        RETURN
      ENDIF
      CALL CHASSN(NPROD1,2,NC1,NC1ASS)
C
C**   Use unreserved chambers (#7 - #18) for products in file 4.
C**   If only two chambers available, schedule 1st product in file 4.
C**   Otherwise, schedule INPR2 products.
  640 NC2ASS = NC2
      DO 644 K = 7,18
        IF (NGROUP(K).EQ.0) THEN
          NGROUP(K) = 4
        ENDIF
  644 CONTINUE
C
      IF (NC2ASS.LE.0) THEN
        GO TO 665
      ELSE IF (NC2ASS.LE.2) THEN
        GO TO 650
      ELSE
        GO TO 660
      ENDIF
C
  650 INPR2 = 1
      NPROD2 = INPR2
      CALL CHASSN(1,4,NC2,NC2ASS)
      GO TO 665
C
  660 WRITE(*,'(5X,A,$)')'Enter value of m  :   '
      READ *, INPR2
      NPROD2 = INPR2
      IF ((NPROD2.GT.NQ(4)).OR.(NPROD2.GT.(NC2/2))) THEN
        WRITE(*,'(//5X,A)')'****** DECREASE THE VALUE OF M ******'
        WRITE(*,'(5X,A)')'*************************************'
        WRITE(*,'(5X,A)')'*************************************'
        RETURN
      ENDIF
      CALL CHASSN(NPROD2,4,NC2,NC2ASS)
C
```

```
  665 IF (NBR.GT.0) THEN
        CALL DRASSN
      ENDIF
C
  667 CALL TIME
      RETURN
      END
C
********************************************************************
      SUBROUTINE RESCH(M,NC,NSTRT)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
C
C**   Get 1st product in file M
      IPROD = LFR(M)
C
C**   Calculate number of chambers assigned (NCR) to the product.
C**   Number of chambers should be even and no greater than number of
C**   chambers available. Calculate number of chambers actually used.
      CR = A(IPROD,13)/168.
      NCR = LINT(CR)
      IF ((NCR/2.0).NE.AINT(NCR/2.0)) THEN
        NCR = NCR + 1
      ELSE
        NCR = NCR
      ENDIF
      NCR = MIN0(NC,NCR)
      CALL SETA(IPROD,16,REAL(NCR))
C
C**   Schedule batches to chambers
C**   and update number of chambers available.
      CHUSED = 0.
      DO 110 K = NSTRT, NSTRT+NCR-1
        IF (NUMBER(K).EQ.0) GO TO 110
        IF (NINT(A(IPROD,2)).LE.0) THEN
          GO TO 115
        ELSE
          CHUSED = CHUSED + 2.
          NCH(K) = 1
          NCHPR(K) = NINT(A(IPROD,1))
          NGROUP(K) = M
          TCSTRT(K) =  TIMNOW + A(IPROD,3) + A(IPROD,21)
          TCFIN(K) = TCSTRT(K) + A(IPROD,4)
          TFR1 = TCSTRT(K) - A(IPROD,21)
          TWET1 = TFR1 - A(IPROD,3)
          SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
          CRT = A(IPROD,2) - 1.0
          CALL SETA(IPROD,2,CRT)
          NGROUP(K+1) = M
          WRITE(10,'(5X,I4,2F12.2,I14,F11.2,F9.2)')
     1    NCHPR(K),TWET1,TFR1,K,TCSTRT(K),TCFIN(K)
          IF (NINT(A(IPROD,2)).GT.0) THEN
            NCH(K+1) = 1
            NCHPR(K+1) = NINT(A(IPROD,1))
            IF (A(IPROD,9).LE.0.) THEN
              TCSTRT(K+1) = TCSTRT(K)
            ELSE
              TCSTRT(K+1) = TCSTRT(K) + A(IPROD,9)
```

```
            ENDIF
            TCFIN(K+1) = TCSTRT(K+1) + A(IPROD,4)
            TFR2 = TCSTRT(K+1) - A(IPROD,21)
            TWET2 = TFR2 - A(IPROD,3)
            SUMCH1(K+1) = SUMCH1(K+1) + A(IPROD,4)
            CRT = A(IPROD,2) - 1.0
            CALL SETA(IPROD,2,CRT)
            WRITE(10,'(5X,I4,2F12.2,I14,F11.2,F9.2)')
     1      NCHPR(K+1),TWET2,TFR2,K+1,TCSTRT(K+1),TCFIN(K+1)
          ENDIF
        ENDIF
  110 CONTINUE
C
  115 CALL SETA(IPROD,17,CHUSED)
      CALL SETA(IPROD,22,TIMNOW)
      CALL REMOVE(IPROD,M)
      CALL INSERT(IPROD,5)
      NC = NC - INT(CHUSED)
      NSTRT = NSTRT + NINT(CHUSED)
      RETURN
      END
C
*********************************************************************
      SUBROUTINE CHASSN(NPROD,M,NC,NCASS)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
      DIMENSION MPRD(6),CR(6),NCR(6),NCACT(6)
C
C**   Calculate the total of chamber processing times (P3T) of NPROD.
      DO 310 I=1,NPROD
        IF (I.EQ.1) THEN
          IPROD = LFR(M)
        ELSE
          IPROD = LSUCC(IPROD)
        ENDIF
        MPRD(I) = NINT(A(IPROD,1))
        SUMP2 = SUMP2 + A(IPROD,12)
  310 CONTINUE
C
C**   Chamber assignment.
C**   Calculate number of chamber assigned to each product based on
C**   ratio of P3T.
      DO 320 I = 1,NPROD
        DO 325 J = 1,NQ(M)
          IF (J.EQ.1) THEN
            IPROD = LFR(M)
          ELSE
            IPROD = LSUCC(IPROD)
          ENDIF
          IF (NINT(A(IPROD,1)).EQ.MPRD(I)) THEN
            CR(I) = (A(IPROD,13)/SUMP2) * REAL(NCASS)
            GO TO 320
          ENDIF
  325   CONTINUE
  320 CONTINUE
C
C**   Sort  products in MPRD(I) based on low-value first of CR(I).
      IF (NPROD.EQ.1) THEN
```

```
        GO TO 350
      ELSE
        KPROD = NPROD - 1
        DO 330 I = 1, KPROD
          CMIN = CR(I)
          MMIN = MPRD(I)
          JMIN = I
          L = I + 1
          DO 340 J=L,NPROD
            IF (CMIN.GT.CR(J)) THEN
              CMIN = CR(J)
              MMIN = MPRD(J)
              JMIN = J
            ENDIF
  340     CONTINUE
          MPRD(JMIN) = MPRD(I)
          MPRD(I) = MMIN
          CR(JMIN) = CR(I)
          CR(I) = CMIN
  330   CONTINUE
      ENDIF
C
C**   Number of chambers assigned to a product must be even and
C**   smaller than number of chambers available.
  350 DO 360 I = 1, NPROD
        NCR(I) = LINT(CR(I))
        IF ((NCR(I)/2.0).NE.AINT(NCR(I)/2.0)) THEN
          NCR(I) = NCR(I) + 1
        ELSE
          NCR(I) = NCR(I)
        ENDIF
        NCR(I) = MIN0(NCR(I),NC)
C
        DO 370 J = 1,NQ(M)
          IF (J.EQ.1) THEN
            IPROD = LFR(M)
          ELSE
            IPROD = LSUCC(IPROD)
          ENDIF
          IF (NINT(A(IPROD,1)).EQ.MPRD(I)) THEN
            NCACT(I) = MIN0(NCR(I),NINT(A(IPROD,2)))
            IF ((NCACT(I)/2.0).NE.AINT(NCACT(I)/2.0)) THEN
              NCR(I) = NCACT(I) + 1
            ELSE
              NCR(I) = NCACT(I)
            ENDIF
            CALL SETA(IPROD,16,REAL(NCR(I)))
            GO TO 375
          ENDIF
  370   CONTINUE
C
C**   Schedule products to chambers.
C**   Update number of chambers available.
  375   CHUSED = 0.
        DO 380 K = 1,18
          IF (NGROUP(K).NE.M) GO TO 380
          IF (NCH(K).EQ.1) GO TO 380
          IF (NUMBER(K).EQ.0) GO TO 380
          IF (NINT(A(IPROD,2)).LE.0) THEN
            GO TO 385
          ELSE
            CHUSED = CHUSED + 2.
```

```
          NCH(K) = 1
          NCHPR(K) = NINT(A(IPROD,1))
          TCSTRT(K) = TIMNOW + A(IPROD,3) + A(IPROD,21)
          TCFIN(K) = TCSTRT(K) + A(IPROD,4)
          TFR1 = TCSTRT(K) - A(IPROD,21)
          TWET1 = TFR1 - A(IPROD,3)
          SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
          CRT = A(IPROD,2) - 1.0
          CALL SETA(IPROD,2,CRT)
          WRITE(10,'(5X,I4,2F12.2,I14,F11.2,F9.2)')
     1    NCHPR(K),TWET1,TFR1,K,TCSTRT(K),TCFIN(K)
          IF (NINT(A(IPROD,2)).GT.0) THEN
            NCH(K+1) = 1
            NCHPR(K+1) = NINT(A(IPROD,1))
            IF (A(IPROD,9).LE.0) THEN
              TCSTRT(K+1) = TCSTRT(K)
            ELSE
              TCSTRT(K+1) = TCSTRT(K) + A(IPROD,9)
            ENDIF
            TCFIN(K+1) = TCSTRT(K+1) + A(IPROD,4)
            TFR2 = TCSTRT(K+1) - A(IPROD,21)
            TWET2 = TFR2 - A(IPROD,3)
            SUMCH1(K+1) = SUMCH1(K+1) + A(IPROD,4)
            CRT = A(IPROD,2) - 1.0
            CALL SETA(IPROD,2,CRT)
            WRITE(10,'(5X,I4,2F12.2,I14,F11.2,F9.2)')
     1      NCHPR(K+1),TWET2,TFR2,K+1,TCSTRT(K+1),TCFIN(K+1)
          ENDIF
          IF (INT(CHUSED).GE.NCR(I)) GO TO 385
        ENDIF
  380   CONTINUE
C
C**   Find number of chamber actually used by the product.
  385   NBR = NBR + 1
        MPR(NBR) = NINT(A(IPROD,1))
        NCHBR(NBR) = NCACT(I)
C
        CALL SETA(IPROD,17,CHUSED)
        CALL SETA(IPROD,22,TIMNOW)
        CALL REMOVE(IPROD,M)
        CALL INSERT(IPROD,5)
        NC = NC - NINT(CHUSED)
  360 CONTINUE
      RETURN
      END
C
**********************************************************************
      SUBROUTINE DRASSN
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
      DIMENSION DR(6),NDR(6),IDRTP(6),LPRDR(6,4),LDRUSD(3)
C
C**   DUMPING ROOM ALLOCATION.
C
C**   Calculate ratio of chamber used by each product.
      KSUM = 0
      DO 700 I = 1, NBR
        KSUM = KSUM + NCHBR(I)
```

```
  700 CONTINUE
      DO 710 I = 1, NBR
        DR(I) = (REAL(NCHBR(I))/REAL(KSUM)) * 3.0
  710 CONTINUE
C
C**   Sort products in MPR(I) based on high-value first of DR(I).
      IF (NBR.EQ.1) THEN
        GO TO 720
      ELSE
        KBR = NBR - 1
        DO 730 I = 1, KBR
          DMAX = DR(I)
          MPMAX = MPR(I)
          NMAX = NCHBR(I)
          JMAX = I
          L = I + 1
          DO 740 J = L, NBR
            IF (DMAX.LT.DR(J)) THEN
              DMAX = DR(J)
              MPMAX = MPR(J)
              NMAX = NCHBR(J)
              JMAX = J
            ENDIF
  740     CONTINUE
          DR(JMAX) = DR(I)
          DR(I) = DMAX
          MPR(JMAX) = MPR(I)
          MPR(I) = MPMAX
          NCHBR(JMAX) = NCHBR(I)
          NCHBR(I) = NMAX
  730   CONTINUE
      ENDIF
C
C**   Calculate number of d-rooms assigned to each product.
  720 IF (DR(1).GE.3.0) THEN
        NDR(1) = 3
        IDRTP(1) = 1
C
      ELSE IF ((DR(1).GE.2.0).AND.(DR(1).LT.3.0)) THEN
        NDR(1) = 2
        IDRTP(1) = 1
        DO 750 I = 2,NBR
          NDR(I) = 1
          IF (NBR.LE.2) THEN
            IDRTP(I) = 1
          ELSE
            IDRTP(I) = 2
          ENDIF
  750   CONTINUE
C
      ELSE IF ((DR(1).GE.1.0).AND.(DR(1).LT.2.0)) THEN
        IF (NBR.EQ.2) THEN
          NDR(1) = 2
          IDRTP(1) = 1
          NDR(2) = 1
          IDRTP(2) = 1
        ELSE
          NDR(1) = 1
          IDRTP(1) = 1
          IF ((DR(2).GE.1.0).AND.(DR(2).LT.2.0)) THEN
            NDR(2) = 1
            IDRTP(2) = 1
```

```
          DO 760 I = 3,NBR
            NDR(I) = 1
            IF (NBR.LE.3) THEN
              IDRTP(I) = 1
            ELSE
              IDRTP(I) = 2
            ENDIF
  760     CONTINUE
        ELSE
          DO 770 I = 2,NBR
            NDR(I) = 2
            IDRTP(I) = 2
  770     CONTINUE
        ENDIF
      ENDIF
C
    ELSE
      DO 780 I = 1,NBR
        NDR(I) = 3
        IDRTP(I) = 2
  780 CONTINUE
    ENDIF
C
C**   Get the old dumping room assignment of each product (stored in
C**   KPRDR(,)); put into LPRDR(,)
      DO 790 I = 1,NBR
        LPRDR(I,1) = MPR(I)
  830   DO 850 L = 1,NN
          IF (KPRDR(L,1).NE.MPR(I)) GO TO 850
          DO 860 J = 2,4
            LPRDR(I,J) = KPRDR(L,J)
  860     CONTINUE
          GO TO 790
  850   CONTINUE
  790 CONTINUE
C
C**   DUMPING ROOM NUMBERS ASSIGNMENT
C
C**   Assign d-room numbers to each product (give new values to
C**   LPRDR(I,2-4)).
      LDR = 0
      DO 1840 I = 1,NBR
        IF (LPRDR(I,2).LE.0) GO TO 1840
        DO 1850 J = 2,NDR(I)+1
C
C**   There is old d-room number in LPRDR(I,J):
          IF (LPRDR(I,J).GT.0) THEN
C
C**   Check if the d-room has been used.
            IF (LDR.LE.0) THEN
              GO TO 1860
            ELSE
              LUSD = 0
              DO 1870 L = 1,LDR
                IF (LDRUSD(L).EQ.LPRDR(I,J)) THEN
                  LUSD = 1
                ENDIF
 1870         CONTINUE
              IF (LUSD.LE.0) THEN
                GO TO 1860
              ELSE
                GO TO 1880
```

```
            ENDIF
          ENDIF
C
C**   The d-room has not been assigned to other product; assign
C**   this d-room.
 1860     LPRDR(I,J) = LPRDR(I,J)
          LDR = LDR + 1
          LDRUSD(LDR) = LPRDR(I,J)
          GO TO 1850
C
C**   The d-room has been assigned to other product;
C**   check the d-room number in LPRDR(I,J+1).
 1880     NXT = J
 1885     NXT = NXT + 1
          IF (NXT.GT.4) THEN
            GO TO 1845
          ELSE
C
C**   There is a d-room number in LPRDR(I,J+1), assign this d-room.
            IF (LPRDR(I,NXT).GT.0) THEN
              IF (LDR.LE.0) THEN
                GO TO 1890
              ELSE
                LUSD = 0
                DO 1895 L = 1,LDR
                  IF (LDRUSD(L).EQ.LPRDR(I,NXT)) THEN
                    LUSD = 1
                  ENDIF
 1895           CONTINUE
                IF (LUSD.LE.0) THEN
                  GO TO 1890
                ELSE
                  GO TO 1885
                ENDIF
              ENDIF
 1890         LPRDR(I,J) = LPRDR(I,NXT)
              LDR = LDR + 1
              LDRUSD(LDR) = LPRDR(I,J)
              GO TO 1850
C
C**   There is no d-room number in LPRDR(I,NXT).
            ELSE
              GO TO 1845
            ENDIF
          ENDIF
C
C**   There is no old DR number in LPRDR(I,J)
        ELSE
          GO TO 1845
        ENDIF
C
C
C**   The product will not share d-room(s) with other product(s).
C**   Find an unassigned d-room.
 1845   IF (IDRTP(I).EQ.1) THEN
          DO 1900 N = 3,5
            IF (LDR.LE.0) THEN
              GO TO 1910
            ELSE
              LUSD = 0
              DO 1920 L = 1,LDR
                IF (LDRUSD(L).EQ.N) THEN
```

```
                LUSD = 1
              ENDIF
 1920       CONTINUE
            IF (LUSD.LE.0) THEN
              GO TO 1910
            ELSE
              GO TO 1900
            ENDIF
          ENDIF
 1900     CONTINUE
 1910     LPRDR(I,J) = N
          LDR = LDR + 1
          LDRUSD(LDR) = N
          GO TO 1850
C
C**   The product will share d-room(s) with other product(s).
C**   Find product(s) which the product has to share d-room with.
C**   If those products already have a d-room number assigned to them,
C**   use that d-room.
        ELSE
          LASSGN = 0
          IF (I.GE.NBR) GO TO 1945
          DO 1930 M = I+1,NBR
            IF (IDRTP(M).NE.2) GO TO 1930
            NXT1 = 1
 1935       NXT1 = NXT1 + 1
            IF (NXT1.GT.4) GO TO 1930
            IF (LPRDR(M,NXT1).LE.0) THEN
              GO TO 1930
            ELSE
              IF (LDR.LE.0) THEN
                GO TO 1940
              ELSE
                LUSD = 0
                DO 1950 L = 1,LDR
                  IF (LDRUSD(L).EQ.LPRDR(M,NXT1)) THEN
                    LUSD = 1
                  ENDIF
 1950           CONTINUE
                IF (LUSD.LE.0) THEN
                  GO TO 1940
                ELSE
                  GO TO 1935
                ENDIF
              ENDIF
 1940         LPRDR(I,J) = LPRDR(M,NXT1)
              LDR = LDR + 1
              LDRUSD(LDR) = LPRDR(M,NXT1)
              LASSGN = 1
              GO TO 1945
            ENDIF
 1930     CONTINUE
 1945     IF (LASSGN.GT.0) THEN
            GO TO 1850
C
C**   No d-room already assigned to them. Find an unassigned d-room.
          ELSE
            DO 1955 N = 3,5
              IF (LDR.LE.0) THEN
                GO TO 1957
              ELSE
                LUSD = 0
```

```
                 DO 1956 L = 1,LDR
                   IF (LDRUSD(L).EQ.N) THEN
                     LUSD = 1
                   ENDIF
 1956            CONTINUE
                 IF (LUSD.LE.0) THEN
                   GO TO 1957
                 ELSE
                   GO TO 1955
                 ENDIF
               ENDIF
 1955        CONTINUE
 1957        LPRDR(I,J) = N
             LDR = LDR + 1
             LDRUSD(LDR) = N
             GO TO 1850
           ENDIF
         ENDIF
C
 1850    CONTINUE
C
         IF (NDR(I).LT.3) THEN
           JSTRT = NDR(I)+2
           DO 1990 J = JSTRT,4
             LPRDR(I,J) = 0
 1990      CONTINUE
         ENDIF
C
C
C**   If the product has to share d-rooms with the other products,
C**   copy all the d-room numbers to those products.
         IF (IDRTP(I).EQ.1) THEN
           GO TO 1840
         ELSE
           DO 2000 II = 1,NBR
             IF ((IDRTP(II).EQ.1).OR.(LPRDR(II,1).EQ.MPR(I)))
     1       GO TO 2000
             DO 2010 J = 2,4
               LPRDR(II,J) = LPRDR(I,J)
 2010        CONTINUE
 2000      CONTINUE
           GO TO 2015
         ENDIF
C
 1840 CONTINUE
C
C**   Find products that has no d-room assigned to them yet.
C**   Assign unassigned d-rooms to them.
 2015 DO 2020 I = 1,NBR
         IF (LPRDR(I,2).GT.0) GO TO 2020
         DO 2030 J = 2,NDR(I)+1
           DO 2040 N = 3,5
             IF (LDR.LE.0) THEN
               GO TO 2050
             ELSE
               LUSD = 0
               DO 2060 L = 1,LDR
                 IF (LDRUSD(L).EQ.N) THEN
                   LUSD = 1
                 ENDIF
 2060          CONTINUE
               IF (LUSD.LE.0) THEN
```

```
              GO TO 2050
            ELSE
              GO TO 2040
            ENDIF
          ENDIF
 2040     CONTINUE
 2050     LPRDR(I,J) = N
          LDR = LDR + 1
          LDRUSD(LDR) = N
 2030   CONTINUE
        IF (NDR(I).LT.3) THEN
          DO 2035 J = NDR(I)+2,4
            LPRDR(I,J) = 0
 2035     CONTINUE
        ENDIF
C
C**   If this product has to share d-room, copy all d-room numbers
C**   to products which this product has to share d-rooms with.
        IF (IDRTP(I).EQ.1) THEN
          GO TO 2020
        ELSE
          DO 2045 II = 1,NBR
            IF ((IDRTP(II).EQ.1).OR.(LPRDR(II,1).EQ.MPR(I)))
     1      GO TO 2045
            DO 2055 J = 2,4
              LPRDR(II,J) = LPRDR(I,J)
 2055       CONTINUE
 2045     CONTINUE
        ENDIF
 2020 CONTINUE
C
C**   Store this new assignment to KPRDR(,).
      DO 2070 I = 1,NBR
        DO 3000 N = 1,NQ(7)
          IF (N.EQ.1) THEN
            IPROD = LFR(7)
          ELSE
            IPROD = LSUCC(IPROD)
          ENDIF
          IF (NINT(A(IPROD,1)).EQ.MPR(I)) GO TO 3010
 3000   CONTINUE
 3010   CALL SETA(IPROD,19,REAL(NDR(I)))
        CALL SETA(IPROD,20,REAL(IDRTP(I)))
        DO 2080 L = 1,NN
          IF (KPRDR(L,1).NE.MPR(I)) GO TO 2080
          DO 2090 J = 2,4
            KPRDR(L,J) = LPRDR(I,J)
 2090     CONTINUE
          GO TO 2070
 2080   CONTINUE
 2070 CONTINUE
C
      NBR = 0
      RETURN
      END
C
**********************************************************************
      SUBROUTINE TIME
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
```

```
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
C
C**   Find time at which the earliest chamber finishes processing
C**   If no more chambers are busy, find total processing time.
C**   Else, set TIMNOW to TIMNXT and call DRSCHD.
      TIMNXT = 300.
      DO 70 K = 1,18
        ITNXT = NINT(TIMNXT * 100)
        ITCFIN = NINT(TCFIN(K) * 100)
        ITCBLK = NINT(TCBLK(K) * 100)
        IF ((NCH(K).EQ.0).AND.(NBLOCK(K).EQ.0)) GO TO 70
        IF ((NBLOCK(K).EQ.0).AND.(ITNXT.GT.ITCFIN)) THEN
          TIMNXT = TCFIN(K)
        ELSE IF ((NBLOCK(K).EQ.1).AND.(ITNXT.GT.ITCBLK)) THEN
          TIMNXT = TCBLK(K)
        ELSE
          TIMNXT = TIMNXT
        ENDIF
   70 CONTINUE
C
      IF (NINT(TIMNXT).EQ.300) THEN
        DO 72 N = 1,5
          IF (N.EQ.1) THEN
            TPROC = TDFIN(N)
          ELSE
            IF (TDFIN(N).GT.TPROC) THEN
              TPROC = TDFIN(N)
            ENDIF
          ENDIF
   72   CONTINUE
      ELSE
        TIMNOW = TIMNXT
        CALL DRSCHD
      ENDIF
      RETURN
      END
C
***********************************************************************
      SUBROUTINE DRSCHD
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
      DIMENSION NRQST(18),TD(5),ITD(5)
      REAL MAXIDL,MINBLK,MINIDL
      INTEGER DRNUM
C
C**   Find chambers that finish processing batches of product or become
C**   available at TIMNOW.
C**   Store the chamber numbers that require d-rooms to send the
C**   batches to array NRQST.
      KC = 0
      ITMNOW = NINT(TIMNOW * 100)
      DO 900 K = 1,18
        ITCFIN = NINT(TCFIN(K) * 100)
        ITCBLK = NINT(TCBLK(K) * 100)
        IF ((ITCFIN.EQ.ITMNOW).OR.(ITCBLK.EQ.ITMNOW)) THEN
          IF (NBLOCK(K).EQ.1) THEN
            NBLOCK(K) = 0
```

```
          TCBLK(K) = 0.0
        ELSE
          NCH(K) = 0
          KC = KC + 1
          NRQST(KC) = K
        ENDIF
      ENDIF
  900 CONTINUE
C
C**   If no chamber requires d-room, call CHSCHD.
      IF (KC.LE.0) THEN
        GO TO 911
      ENDIF
C
      WRITE(10,'(/2X,A,F6.2)')
     1'Chamber(s) finishes operation at time ',TIMNOW
      WRITE(10,'(5X,A)')
     1'Finished   Chamber  Blocking    D-room   Starting   Completion'
      WRITE(10,'(5X,A)')
     1'Product #  number     time        number     time        time'
C
C**   FIND A D-ROOM FOR EACH CHAMBER IN NRQST
C
      DO 910 I = 1,KC
        K = NRQST(I)
        DRNUM = 0
C
C**   Access the product number and time at which each d-room can
C**   start processing the batch.
C
        DO 920 M = 1,NQ(7)
          IF (M.EQ.1) THEN
            IPROD = LFR(7)
          ELSE
            IPROD = LSUCC(IPROD)
          ENDIF
          IF (NINT(A(IPROD,1)).EQ.NCHPR(K)) GO TO 930
  920   CONTINUE
  930   DO 940 N = 1,5
C
C**   No previous product: just include the setup time.
          IF (NDRPR(N).LE.0) THEN
            TD(N) = TDFIN(N) + A(IPROD,5)
C
C**   The previous product processed in d-room is the same type of
C**   product: cleaning & setup time not included.
          ELSE IF (NDRPR(N).EQ.NINT(A(IPROD,1))) THEN
            TD(N) = TDFIN(N)
C
C**   Type of the previous product is different: include cleaning &
C**   setup time.
          ELSE
            DO 960 M = 1,NQ(7)
              IF (M.EQ.1) THEN
                KPROD = LFR(7)
              ELSE
                KPROD = LSUCC(KPROD)
              ENDIF
              IF (INT(A(KPROD,1)).EQ.NDRPR(N)) GO TO 950
  960       CONTINUE
  950       TD(N) = TDFIN(N) + A(KPROD,6) + A(IPROD,5)
          ENDIF
```

```
          ITD(N) = NINT(TD(N) * 100)
  940   CONTINUE
C
C
C**   Check the type of d-room the batch has to be sent.
        IF (NINT(A(IPROD,10)).LT.3) THEN
          GO TO 970
        ELSE
          GO TO 1000
        ENDIF
C
C**   The batch has to be sent to a dedicated d-room.
C**   Compare TIMNOW and the time at which the d-room can start
C**   processing the product.
  970   DRNUM = INT(A(IPROD,10))
        IF (ITD(DRNUM).LE.ITMNOW) THEN
          GO TO 980
        ELSE
          GO TO 990
        ENDIF
        GO TO 914
C
C**   The batch has to be sent to regular d-rooms.
C**   Find number of d-rooms & d-room numbers assigned to the product.
 1000   MAXIDL = -100
        MINBLK = 1000
        MINIDL = 1000
        NDRASS = INT(A(IPROD,19))
        DO 1010 L = 1,NN
          IF (KPRDR(L,1).EQ.NINT(A(IPROD,1))) GO TO 1020
 1010   CONTINUE
C
C**   If there is d-room whose product is the same as the coming
C**   product, check if the coming product can be scheduled to that
C**   d-room without causing blocking in chamber.
C**   Select d-room (if more one) which will give minimum idle time.
 1020   DO 1030 J = 2,NDRASS+1
          N = KPRDR(L,J)
          IF (NDRPR(N).EQ.INT(A(IPROD,1))) THEN
            IF (ITD(N).LE.ITMNOW) THEN
              TIDLE = TIMNOW - TD(N)
              IF (TIDLE.LT.MINIDL) THEN
                MINIDL = TIDLE
                DRNUM = N
              ENDIF
            ENDIF
          ENDIF
 1030   CONTINUE
        IF (DRNUM.GT.0) THEN
          GO TO 980
C
C**   There is no such d-room. Check the idle time of each d-room and
C**   select a d-room that has the longest idle time.
        ELSE
          DO 1040 J = 2, NDRASS+1
            N = KPRDR(L,J)
            IF (ITD(N).LE.ITMNOW) THEN
              TIDLE = TIMNOW - TD(N)
              IF (TIDLE.GT.MAXIDL) THEN
                MAXIDL = TIDLE
                DRNUM = N
              ENDIF
```

```
            ENDIF
 1040     CONTINUE
          IF (DRNUM.GT.0) THEN
            GO TO 980
C
C**   No idle d-room at all.
C**   Select a d-room that becomes available first.
          ELSE
            DO 1050 J = 2, NDRASS+1
              N = KPRDR(L,J)
              BTIME = TD(N) - TIMNOW
              IF (BTIME.LT.MINBLK) THEN
                MINBLK = BTIME
                DRNUM = N
              ENDIF
 1050       CONTINUE
            GO TO 990
          ENDIF
        ENDIF
C
C
C**   Scheduling batch to an available d-room.
  980   TDSTRT(DRNUM) = TIMNOW
        TDFIN(DRNUM) = TDSTRT(DRNUM) + A(IPROD,7)
        SUMDR1(DRNUM) = SUMDR1(DRNUM) + A(IPROD,7)
        IF (NDRPR(DRNUM).EQ.0) THEN
          TDSET(DRNUM) = TDSTRT(DRNUM) - A(IPROD,5)
          SUMDR2(DRNUM) = SUMDR2(DRNUM) + A(IPROD,5)
        ELSE IF (NDRPR(DRNUM).EQ.NINT(A(IPROD,1))) THEN
          TDSET(DRNUM) = TDSTRT(DRNUM)
        ELSE
          TDSET(DRNUM) = TDSTRT(DRNUM) - A(IPROD,5)
          SUMDR2(DRNUM) = SUMDR2(DRNUM) + A(KPROD,6) + A(IPROD,5)
        ENDIF
        NDRPR(DRNUM) = NINT(A(IPROD,1))
        BLK = 0.0
        WRITE(10,'(5X,I5,I10,F11.2,I11,F12.2,F11.2)')
     1  NDRPR(DRNUM),K,BLK,DRNUM,TDSTRT(DRNUM),TDFIN(DRNUM)
        GO TO 914
C
C**   Scheduling batch to an unavailable d-room.
C**   Block the chamber.
  990   TDSTRT(DRNUM) = TD(DRNUM)
        TDFIN(DRNUM) = TDSTRT(DRNUM) + A(IPROD,7)
        SUMDR1(DRNUM) = SUMDR1(DRNUM) + A(IPROD,7)
        IF (NDRPR(DRNUM).EQ.0) THEN
          TDSET(DRNUM) = TDSTRT(DRNUM) - A(IPROD,5)
          SUMDR2(DRNUM) = SUMDR2(DRNUM) + A(IPROD,5)
        ELSE IF (NDRPR(DRNUM).EQ.INT(A(IPROD,1))) THEN
          TDSET(DRNUM) = TDSTRT(DRNUM)
        ELSE
          TDSET(DRNUM) = TDSTRT(DRNUM) - A(IPROD,5)
          SUMDR2(DRNUM) = SUMDR2(DRNUM) + A(KPROD,6) + A(IPROD,5)
        ENDIF
        NDRPR(DRNUM) = NINT(A(IPROD,1))
        BLK = TDSTRT(DRNUM) - TIMNOW
        TCBLK(K) = TIMNOW + BLK
        NBLOCK(K) = 1
        SUMCH2(K) = SUMCH2(K) + BLK
        WRITE(10,'(5X,I5,I10,F11.2,I11,F12.2,F11.2)')
     1  NDRPR(DRNUM),K,BLK,DRNUM,TDSTRT(DRNUM),TDFIN(DRNUM)
C
```

```
C**   Check if the batch is the last batch of the product.
  914   DO 915 M = 1,NQ(5)
          IF (M.EQ.1) THEN
            MPROD = LFR(5)
          ELSE
            MPROD = LSUCC(MPROD)
          ENDIF
          IF (NINT(A(MPROD,1)).NE.NINT(A(IPROD,1))) THEN
            GO TO 915
          ENDIF
          IF (TDFIN(DRNUM).GT.A(MPROD,23)) THEN
            CALL SETA(MPROD,23,TDFIN(DRNUM))
            GO TO 910
          ENDIF
  915   CONTINUE
C
  910 CONTINUE
C
  911 CALL CHSCHD
      RETURN
      END
C
**********************************************************************
      SUBROUTINE CHSCHD
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
      DIMENSION MPRD(6),PTIME(6),CR(6),NCR(6),IDLE(18),IDL1(18),
     1IDL2(18),ICRT(4),NEWASS(18),TWET(18),TFREEZ(18)
C
C***  Find idle chambers. If no idle chambers, move to TIMNXT
      NUMCH = 0
      DO 1100 K = 1,18
        IF ((NCH(K).EQ.0).AND.(NBLOCK(K).EQ.0)) THEN
          NUMCH = NUMCH + 1
          IDLE(NUMCH) = K
        ENDIF
 1100 CONTINUE
      IF (NUMCH.LE.0) THEN
        CALL TIME
        RETURN
      ENDIF
C
      NNWAS = 0
C
C******************   M = 1   &   M = 3   ********************
C
      DO 1110 M = 1,3,2
C
C**  Find idle chambers used by group M.
C**  Separate single idle chambers from pair (tandem) idle chambers.
        II1 = 0
        II2 = 0
        DO 1120 I = 1,NUMCH
          K = IDLE(I)
          IF (NGROUP(K).EQ.M) THEN
            IF (NUMBER(K).EQ.1) THEN
              IF ((NCH(K+1).EQ.0).AND.(NBLOCK(K+1).EQ.0)) THEN
```

```
              II2 = II2 + 1
              IDL2(II2) = K
            ELSE
              II1 = II1 + 1
              IDL1(II1) = K
            ENDIF
          ELSE
            IF ((NCH(K-1).EQ.0).AND.(NBLOCK(K-1).EQ.0)) THEN
              II2 = II2 + 1
              IDL2(II2) = K
            ELSE
              II1 = II1 + 1
              IDL1(II1) = K
            ENDIF
          ENDIF
        ENDIF
 1120   CONTINUE
C
        IF (II1.GT.0) THEN
          GO TO 1220
        ELSE IF (II2.GT.0) THEN
          GO TO 1260
        ELSE
          GO TO 1110
        ENDIF
C
C**  Single idle chambers
C
C**  Get product number.
 1220   DO 1230 I = 1,II1
          K = IDL1(I)
          IF (NCHPR(K).EQ.0) GO TO 1230
          DO 1240 L = 1,NQ(5)
            IF (L.EQ.1) THEN
              IPROD = LFR(5)
            ELSE
              IPROD = LSUCC(IPROD)
            ENDIF
            IF (NINT(A(IPROD,1)).EQ.NCHPR(K)) GO TO 1250
 1240     CONTINUE
C
C**  There are batches to be processed, schedule one batch to chamber.
 1250     IF (NINT(A(IPROD,2)).GT.0) THEN
            IF (A(IPROD,9).LE.0.0) THEN
              TCSTRT(K) = TIMNOW
            ELSE
              IF (NUMBER(K).EQ.1) THEN
                TC = TCSTRT(K+1) + A(IPROD,9)
              ELSE
                TC = TCSTRT(K-1) + A(IPROD,9)
              ENDIF
              TCSTRT(K) = AMAX1(TIMNOW,TC)
            ENDIF
            TCFIN(K) = TCSTRT(K) + A(IPROD,4)
            TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
            TWET(K) = TFREEZ(K) - A(IPROD,3)
            NCH(K) = 1
            SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
            CRT = A(IPROD,2) - 1.0
            CALL SETA (IPROD,2,CRT)
            NNWAS = NNWAS + 1
            NEWASS(NNWAS) = K
```

```
C
C**  No more batches to be processed. The chamber becomes idle.
          ELSE
            NCH(K) = 0
            NCHPR(K) = 0
            TCSTRT(K) = 0.0
            TCFIN(K) = 0.0
          ENDIF
 1230   CONTINUE
C
C**  Idle chambers in pair.
C
C**   Get the product number in each pair.
 1260   DO 1270 I = 1,II2
          K = IDL2(I)
          IF (NUMBER(K).EQ.0) GO TO 1270
          LPRNUM = 0
          IF (NCHPR(K).NE.0) THEN
            LPRNUM = NCHPR(K)
          ELSE
            LPRNUM = NCHPR(K+1)
          ENDIF
          IF (LPRNUM.LE.0) GO TO 1270
          DO 1280 L = 1,NQ(5)
            IF (L.EQ.1) THEN
              IPROD = LFR(5)
            ELSE
              IPROD = LSUCC(IPROD)
            ENDIF
            IF (NINT(A(IPROD,1)).EQ.LPRNUM) GO TO 1290
 1280     CONTINUE
C
C**  There are batches to be processed.
C**  Schedule two batches to two chambers.
 1290     IF (NINT(A(IPROD,2)).GT.0) THEN
            NCH(K) = 1
            TC = 0.+ A(IPROD,3)+ A(IPROD,21)
            TCSTRT(K) = AMAX1(TIMNOW,TC)
            NCHPR(K) = NINT(A(IPROD,1))
            TCFIN(K) = TCSTRT(K) + A(IPROD,4)
            TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
            TWET(K) = TFREEZ(K) - A(IPROD,3)
            SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
            CRT = A(IPROD,2) - 1.0
            CALL SETA(IPROD,2,CRT)
            NNWAS = NNWAS + 1
            NEWASS(NNWAS) = K
            IF (NINT(A(IPROD,2)).GT.0) THEN
              NCH(K+1) = 1
              NCHPR(K+1) = NINT(A(IPROD,1))
              IF (A(IPROD,9).LE.0.) THEN
                TCSTRT(K+1) = TCSTRT(K)
              ELSE
                TCSTRT(K+1) = TCSTRT(K) + A(IPROD,9)
              ENDIF
              TCFIN(K+1) = TCSTRT(K+1) + A(IPROD,4)
              TFREEZ(K+1) = TCSTRT(K+1) - A(IPROD,21)
              TWET(K+1) = TFREEZ(K+1) - A(IPROD,3)
              SUMCH1(K+1) = SUMCH1(K+1) + A(IPROD,4)
              CRT = A(IPROD,2) - 1.0
              CALL SETA(IPROD,2,CRT)
              NNWAS = NNWAS + 1
```

```
              NEWASS(NNWAS) = K+1
            ENDIF
C
C**   No more batches to be processed.
C**   Check if all batches of the product are done in other chambers.
          ELSE
            CHUS = A(IPROD,17) - 2.0
            CALL SETA(IPROD,17,CHUS)
            NCHPR(K) = 0
            NCHPR(K+1) = 0
            TCSTRT(K) = 0.0
            TCSTRT(K+1) = 0.0
            TCFIN(K) = 0.0
            TCFIN(K+1) = 0.0
            JJ = 0
            DO 1300 J = 1,18
              IF (NGROUP(J).NE.M) GO TO 1300
              IF (NCHPR(J).EQ.NINT(A(IPROD,1))) THEN
                JJ = 1
              ENDIF
 1300       CONTINUE
C
C**   The product is not done yet. Leave the chambers idle.
            IF (JJ.GT.0) THEN
              GO TO 1270
C
C**   The product is done.
C**   Pick a new product from file M (if any) and schedule two batches
C**   to the pair.
            ELSE
              CALL REMOVE(IPROD,5)
              CALL INSERT(IPROD,6)
              IF (NQ(M).GT.0) THEN
                IPROD = LFR(M)
                NCH(K) = 1
                NCHPR(K) = NINT(A(IPROD,1))
                TC = 0.+ A(IPROD,3)+ A(IPROD,21)
                TCSTRT(K) = AMAX1(TIMNOW,TC)
                TCFIN(K) = TCSTRT(K) + A(IPROD,4)
                TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
                TWET(K) = TFREEZ(K) - A(IPROD,3)
                SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
                CRT = A(IPROD,2) - 1.0
                CALL SETA(IPROD,2,CRT)
                NNWAS = NNWAS + 1
                NEWASS(NNWAS) = K
                IF (NINT(A(IPROD,2)).GT.0) THEN
                  NCH(K+1) = 1
                  NCHPR(K+1) = NINT(A(IPROD,1))
                  IF (A(IPROD,9).LE.0.) THEN
                    TCSTRT(K+1) = TCSTRT(K)
                  ELSE
                    TCSTRT(K+1) = TCSTRT(K) + A(IPROD,9)
                  ENDIF
                  TCFIN(K+1) = TCSTRT(K+1) + A(IPROD,4)
                  TFREEZ(K+1) = TCSTRT(K+1) - A(IPROD,21)
                  TWET(K+1) = TFREEZ(K+1) - A(IPROD,3)
                  SUMCH1(K+1) = SUMCH1(K+1) + A(IPROD,4)
                  CRT = A(IPROD,2) - 1.0
                  CALL SETA(IPROD,2,CRT)
                  NNWAS = NNWAS + 1
                  NEWASS(NNWAS) = K + 1
```

```
                  ENDIF
                  CALL SETA(IPROD,17,2.0)
                  CALL SETA(IPROD,22,TWET(K))
                  CALL REMOVE(IPROD,M)
                  CALL INSERT(IPROD,5)
C
C**   No new product in file M, leave the chamber idle.
                ELSE
                  GO TO 1270
                ENDIF
              ENDIF
            ENDIF
 1270     CONTINUE
C
C**   Go over the idle chambers again and find pairs of chambers that
C**   can be utilized.
C
          DO 1310 I = 1,II2
            K = IDL2(I)
            IF (NGROUP(K).NE.M) GO TO 1310
            IF (NUMBER(K).EQ.0) GO TO 1310
            IF ((NCH(K).EQ.1).OR.(NBLOCK(K).EQ.1).OR.
     1      (NCH(K+1).EQ.1).OR.(NBLOCK(K+1).EQ.1)) GO TO 1310
C
C**   A pair of chamber is idle.
C**   Find a being-processed product in group M that has largest number
C**   of batches to be processed.
            IREM = 0
            IPRN = 0
            DO 1320 L = 1,NQ(5)
              IF (L.EQ.1) THEN
                IPROD = LFR(5)
              ELSE
                IPROD = LSUCC(IPROD)
              ENDIF
              IF (NINT(A(IPROD,15)).NE.M) GO TO 1320
              IF (NINT(A(IPROD,2)).GT.IREM) THEN
                IREM = NINT(A(IPROD,2))
                IPRN = NINT(A(IPROD,1))
              ENDIF
 1320       CONTINUE
C
C**   There is such product, schedule 2 batches to the chambers.
            IF (IPRN.GT.0) THEN
              DO 1330 L = 1,NQ(5)
                IF (L.EQ.1) THEN
                  IPROD = LFR(5)
                ELSE
                  IPROD = LSUCC(IPROD)
                ENDIF
                IF (NINT(A(IPROD,1)).EQ.IPRN) GO TO 1335
 1330         CONTINUE
 1335         NCH(K) = 1
              NCHPR(K) = NINT(A(IPROD,1))
              TC = 0.+ A(IPROD,3)+ A(IPROD,21)
              TCSTRT(K) = AMAX1(TIMNOW,TC)
              TCFIN(K) = TCSTRT(K) + A(IPROD,4)
              TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
              TWET(K) = TFREEZ(K) - A(IPROD,3)
              SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
              CRT = A(IPROD,2) - 1.0
              CALL SETA(IPROD,2,CRT)
```

```
            NNWAS = NNWAS + 1
            NEWASS(NNWAS) = K
            IF (NINT(A(IPROD,2)).GT.0) THEN
              NCH(K+1) = 1
              NCHPR(K+1) = NINT(A(IPROD,1))
              IF (A(IPROD,9).LE.0) THEN
                TCSTRT(K+1) = TCSTRT(K)
              ELSE
                TCSTRT(K+1) = TCSTRT(K) + A(IPROD,9)
              ENDIF
              TCFIN(K+1) = TCSTRT(K+1) + A(IPROD,4)
              TFREEZ(K+1) = TCSTRT(K+1) - A(IPROD,21)
              TWET(K+1) = TFREEZ(K+1) - A(IPROD,3)
              SUMCH1(K+1) = SUMCH1(K+1) + A(IPROD,4)
              CRT = A(IPROD,2) - 1.0
              CALL SETA(IPROD,2,CRT)
              NNWAS = NNWAS + 1
              NEWASS(NNWAS) = K + 1
           ENDIF
            CHUS = A(IPROD,17) + 2.0
            CALL SETA(IPROD,17,CHUS)
C
C**  No largest being-processed product. Check group (file) M.
C**  If group M is not empty, hold the chambers for group M.
          ELSE
            IF (NQ(M).GT.0) THEN
              GO TO 1310
C
C**   If group M (the unprocessed products) is empty, find a largest
C**   group.   M = 1 -- check groups 2,3,4.
C**            M = 3 -- check group 4
            ELSE
              LARGE = 0
              CART = 0.0
              IF (M.EQ.1) THEN
                DO 1340 L = 1,4
                  IF (NQ(L).LE.0) GO TO 1340
                  TOTCRT = 0.0
                  DO 1350 J = 1,NQ(L)
                    IF (J.EQ.1) THEN
                      JPROD = LFR(L)
                    ELSE
                      JPROD = LSUCC(JPROD)
                    ENDIF
                    TOTCRT = TOTCRT + A(JPROD,2)
 1350             CONTINUE
                  IF (TOTCRT.GT.CART) THEN
                    CART = TOTCRT
                    LARGE = L
                  ENDIF
 1340           CONTINUE
              ELSE
                IF (NQ(4).GT.0) THEN
                  LARGE = 4
                ENDIF
              ENDIF
C
C**   If there is a largest group, assign chambers to that group.
              IF (LARGE.GT.0) THEN
                NGROUP(K) = LARGE
                NGROUP(K+1) = LARGE
                IF (M.EQ.1) THEN
```

```
                NC1R = NC1R - 2
              ELSE
                NC2R = NC2R - 2
              ENDIF
              IF (LARGE.EQ.2) THEN
                NC1ASS = NC1ASS + 2
              ELSE IF (LARGE.EQ.3) THEN
                NC2R = NC2R + 2
              ELSE
                NC2ASS = NC2ASS + 2
              ENDIF
C
C**   There is no largest group, find the largest group of 'being-
C**   processed product'.  M = 1 --  check 2,3,4
C**                        M = 3 --  check 4
            ELSE
              DO 1360 L = 2,4
                ICRT(L) = 0
 1360         CONTINUE
              DO 1365 J = 1,NQ(5)
                IF (J.EQ.1) THEN
                  JPROD = LFR(5)
                ELSE
                  JPROD = LSUCC(JPROD)
                ENDIF
                L = NINT(A(JPROD,15))
                ICRT(L) = ICRT(L) + NINT(A(IPROD,2))
 1365         CONTINUE
              LARGE = 0
              KCART = 0
              IF (M.EQ.1) THEN
                DO 1370 L = 2,4
                  IF (KCART.GT.ICRT(L)) THEN
                    KCART = ICRT(L)
                    LARGE = L
                  ENDIF
 1370           CONTINUE
              ELSE
                IF (ICRT(4).GT.0) THEN
                  LARGE = 4
                ENDIF
              ENDIF
C
C**   If there is such group, assign the chambers to that group.
C**   Otherwise, hold the chambers for group M.
              IF (LARGE.LE.0) THEN
                GO TO 1310
              ELSE
                NGROUP(K) = LARGE
                NGROUP(K+1) = LARGE
                IF (M.EQ.1) THEN
                  NC1R = NC1R - 2
                ELSE
                  NC2R = NC2R - 2
                ENDIF
                IF (LARGE.EQ.2) THEN
                  NC1ASS = NC1ASS + 2
                ELSE IF (LARGE.EQ.3) THEN
                  NC2R = NC2R + 2
                ELSE
                  NC2ASS = NC2ASS + 2
                ENDIF
```

```
              ENDIF
            ENDIF
          ENDIF
        ENDIF
 1310   CONTINUE
 1110 CONTINUE
C
C*****************      M = 2  &  M = 4      *******************
C
      DO 1400 M = 2,4,2
C
C**   Find idle chambers used by group M.
C**   Separate single idle chambers from pair (tandem) idle chambers.
        II1 = 0
        II2 = 0
        DO 1410 I = 1,NUMCH
          K = IDLE(I)
          IF (NGROUP(K).EQ.M) THEN
            IF (NUMBER(K).EQ.1) THEN
              IF ((NCH(K+1).EQ.0).AND.(NBLOCK(K+1).EQ.0)) THEN
                II2 = II2 + 1
                IDL2(II2) = K
              ELSE
                II1 = II1 + 1
                IDL1(II1) = K
              ENDIF
            ELSE
              IF ((NCH(K-1).EQ.0).AND.(NBLOCK(K-1).EQ.0)) THEN
                II2 = II2 + 1
                IDL2(II2) = K
              ELSE
                II1 = II1 + 1
                IDL1(II1) = K
              ENDIF
            ENDIF
          ENDIF
 1410   CONTINUE
        NEW = 0
        IF (II2.LE.0) GO TO 1475
C
C**   Go through the pair chambers to check if new product is needed.
C
C**   Get the product number.
        DO 1440 I = 1,II2
          K = IDL2(I)
          IF (NUMBER(K).EQ.0) GO TO 1440
          KPRNUM = 0
          IF (NCHPR(K).NE.0) THEN
            KPRNUM = NCHPR(K)
          ELSE
            KPRNUM = NCHPR(K+1)
          ENDIF
          IF (KPRNUM.LE.0) GO TO 1440
          DO 1450 L = 1,NQ(5)
            IF (L.EQ.1) THEN
              IPROD = LFR(5)
            ELSE
              IPROD = LSUCC(IPROD)
            ENDIF
            IF (NINT(A(IPROD,1)).EQ.KPRNUM) GO TO 1460
 1450     CONTINUE
C
```

```
C**   If either chamber is reserved, set it unreserved.
 1460     IF ((NRESV(K).EQ.1).OR.(NRESV(K+1).EQ.1)) THEN
            NRESV(K) = 0
            NRESV(K+1) = 0
            USED = A(IPROD,17) - 1.0
            RES = A(IPROD,18) - 1.0
            CALL SETA(IPROD,17,USED)
            CALL SETA(IPROD,18,RES)
          ELSE
            USED = A(IPROD,17) - 2.0
            CALL SETA(IPROD,17,USED)
          ENDIF
          NCHPR(K) = 0
          NCHPR(K+1) = 0
          TCSTRT(K) = 0.0
          TCSTRT(K+1) = 0.0
          TCFIN(K) = 0.0
          TCFIN(K+1) = 0.0
C
C**   Check if the pair needs new product.
C**   If all batches of the product are done in all chambers, a new
C**   product is needed.
          IF (NINT(A(IPROD,2)).GT.0) THEN
            GO TO 1440
          ELSE
            JJ = 0
            DO 1470 J = 1,18
              IF (NGROUP(J).NE.M) GO TO 1470
              IF (NCHPR(J).EQ.NINT(A(IPROD,1))) THEN
                JJ = 1
              ENDIF
 1470       CONTINUE
            IF (JJ.LE.0) THEN
              NEW = NEW + 1
              CALL REMOVE(IPROD,5)
              CALL INSERT(IPROD,6)
            ELSE
              GO TO 1440
            ENDIF
          ENDIF
 1440   CONTINUE
C
C
C**   Get all product numbers that are being processed on chambers.
 1475   NP = 0
        DO 1480 I = 1,NQ(5)
          IF (I.EQ.1) THEN
            IPROD = LFR(5)
          ELSE
            IPROD = LSUCC(IPROD)
          ENDIF
          IF (NINT(A(IPROD,15)).EQ.M) THEN
            NP = NP+1
            MPRD(NP) = NINT(A(IPROD,1))
            P2 = A(IPROD,2) * A(IPROD,4)
            DO 1490 J = 1,18
              IF (NCHPR(J).EQ.NINT(A(IPROD,1))) THEN
                IF (TCFIN(J).GT.TIMNOW) THEN
                  P2 = P2 + (TCFIN(J) - TIMNOW)
                ENDIF
              ENDIF
 1490       CONTINUE
```

```
            PTIME(NP) = P2
          ENDIF
 1480   CONTINUE
C
C**    Get new product (if any) from group M.
        IF (M.EQ.4) THEN
          IF (NLACK.GT.0) THEN
            NEW = NEW + NLACK
          ENDIF
        ENDIF
        IF (NEW.LE.0) GO TO 1510
 1500   IF (NQ(M).GT.0) THEN
          IPROD = LFR(M)
          NP = NP + 1
          MPRD(NP) = NINT(A(IPROD,1))
          PTIME(NP) = A(IPROD,13)
          CALL SETA(IPROD,22,-10.)
          CALL REMOVE(IPROD,M)
          CALL INSERT(IPROD,5)
          NEW = NEW - 1
          IF (NEW.GT.0) THEN
            GO TO 1500
          ELSE
            IF (M.EQ.2) THEN
              NPROD1 = NP
            ELSE
              NPROD2 = NP
            ENDIF
            GO TO 1510
          ENDIF
        ELSE
          IF (M.EQ.2) THEN
            NPROD1 = NP
            NLACK = NEW
          ELSE
            NPROD2 = NP
          ENDIF
        ENDIF
C
C**    Check if some chambers of M = 2 should be assigned to M = 4.
 1510   IF (M.EQ.4) THEN
          IF (NPROD1.LT.INPR1) THEN
            DO 1680 J = 1,NUMCH
              L = IDLE(J)
              IF (NGROUP(L).NE.2) GO TO 1680
              IF (NUMBER(L).EQ.0) GO TO 1680
              IF ((NCH(L).EQ.0).AND.(NBLOCK(L).EQ.0).AND.
     1          (NCH(L+1).EQ.0).AND.(NBLOCK(L+1).EQ.0)) THEN
                NGROUP(L) = 4
                NGROUP(L+1) = 4
                NC1ASS = NC1ASS - 2
                NC2ASS = NC2ASS + 2
              ENDIF
 1680       CONTINUE
          ENDIF
        ENDIF
C
C**   CHAMBER REASSIGNMENT
C
C**    Calculate number of chambers assigned to each product based on
C**    the ratio of remaining chamber processing time.
        IF (NP.LE.0) GO TO 1400
```

```
      SUMP2 = 0.0
      DO 1520 I = 1,NP
        SUMP2 = SUMP2 + PTIME(I)
 1520 CONTINUE
      IF (SUMP2.LE.0.0) GO TO 1400
      IF (M.EQ.2) THEN
        NCAV = NC1ASS
      ELSE
        NCAV = NC2ASS
      ENDIF
      DO 1530 I = 1,NP
        CR(I) = (PTIME(I)/SUMP2) * REAL(NCAV)
 1530 CONTINUE
      IF (NP.EQ.1) THEN
        GO TO 1540
      ELSE
        LP = NP - 1
        DO 1550 I = 1,LP
          CMIN = CR(I)
          MMIN = MPRD(I)
          JMIN = I
          L = I + 1
          DO 1560 J = L,NP
            IF (CMIN.GT.CR(J)) THEN
              CMIN = CR(J)
              MMIN = MPRD(J)
              JMIN = J
            ENDIF
 1560     CONTINUE
          MPRD(JMIN) = MPRD(I)
          MPRD(I) = MMIN
          CR(JMIN) = CR(I)
          CR(I) = CMIN
 1550   CONTINUE
      ENDIF
 1540 DO 1570 I = 1,NP
        NCR(I) = LINT(CR(I))
        IF ((NCR(I)/2.).NE.AINT(NCR(I)/2.)) THEN
          NCR(I) = NCR(I) + 1
        ELSE
          NCR(I) = NCR(I)
        ENDIF
        NCR(I) = MIN0(NCR(I),NCAV)
        DO 1580 J = 1,NQ(5)
          IF (J.EQ.1) THEN
            IPROD = LFR(5)
          ELSE
            IPROD = LSUCC(IPROD)
          ENDIF
          IF (NINT(A(IPROD,1)).EQ.MPRD(I)) GO TO 1590
 1580   CONTINUE
 1590   CALL SETA(IPROD,16,REAL(NCR(I)))
        NCAV = NCAV - NCR(I)
 1570 CONTINUE
C
C**   PRODUCT SCHEDULING
C
      DO 1600 I = 1,NP
        DO 1610 J = 1,NQ(5)
          IF (J.EQ.1) THEN
            IPROD = LFR(5)
          ELSE
```

```
            IPROD = LSUCC(IPROD)
          ENDIF
          IF (NINT(A(IPROD,1)).EQ.MPRD(I)) GO TO 1620
 1610   CONTINUE
C
C**   Schedule batches to single idle chambers.
C
 1620   IF (II1.LE.0) GO TO 1640
        DO 1630 L = 1,II1
          K = IDL1(L)
          IF (NCHPR(K).NE.NINT(A(IPROD,1))) GO TO 1630
          IF (NINT(A(IPROD,2)).LE.0) THEN
            NCHPR(K) = 0
            NRESV(K) = 0
            TCSTRT(K) = 0.0
            TCFIN(K) = 0.0
            GO TO 1630
          ENDIF
C
C**   The chamber is unreserved chamber.
          IF (NRESV(K).EQ.0) THEN
C**   Number of chambers used <= number of chambers assigned.
            IF (A(IPROD,17).LE.A(IPROD,16)) THEN
              NCH(K) = 1
              IF (A(IPROD,9).LE.0.0) THEN
                TCSTRT(K) = TIMNOW
              ELSE
                IF (NUMBER(K).EQ.1) THEN
                  TC = TCSTRT(K+1) + A(IPROD,9)
                ELSE
                  TC = TCSTRT(K-1) + A(IPROD,9)
                ENDIF
                TCSTRT(K) = AMAX1(TIMNOW,TC)
              ENDIF
              TCFIN(K) = TCSTRT(K) + A(IPROD,4)
              TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
              TWET(K) = TFREEZ(K) - A(IPROD,3)
              SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
              CRT = A(IPROD,2) - 1.0
              CALL SETA (IPROD,2,CRT)
              NNWAS = NNWAS + 1
              NEWASS(NNWAS) = K
C**   Number of chambers used > number of chambers assigned.
            ELSE
              IF (A(IPROD,18).LE.0.0) THEN
                NRESV(K) = 1
                TCSTRT(K) = 0.0
                TCFIN(K) = 0.0
                RES = A(IPROD,18) + 1.
                USED = A(IPROD,17) - 1.
                CALL SETA(IPROD,18,RES)
                CALL SETA(IPROD,17,USED)
              ELSE
                IDIFF = A(IPROD,17) - A(IPROD,16)
                IF (IDIFF.GT.A(IPROD,18)) THEN
                  NRESV(K) = 1
                  TCSTRT(K) = 0.0
                  TCFIN(K) = 0.0
                  RES = A(IPROD,18) + 1.
                  USED = A(IPROD,17) - 1.
                  CALL SETA(IPROD,18,RES)
                  CALL SETA(IPROD,17,USED)
```

```
                 ELSE
                   NCH(K) = 1
                   IF (A(IPROD,9).LE.0.0) THEN
                     TCSTRT(K) = TIMNOW
                   ELSE
                     IF (NUMBER(K).EQ.1) THEN
                       TC = TCSTRT(K+1) + A(IPROD,9)
                     ELSE
                       TC = TCSTRT(K-1) + A(IPROD,9)
                     ENDIF
                     TCSTRT(K) = AMAX1(TIMNOW,TC)
                   ENDIF
                   TCFIN(K) = TCSTRT(K) + A(IPROD,4)
                   TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
                   TWET(K) = TFREEZ(K) - A(IPROD,3)
                   SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
                   CRT = A(IPROD,2) - 1.0
                   CALL SETA (IPROD,2,CRT)
                   NNWAS = NNWAS + 1
                   NEWASS(NNWAS) = K
                 ENDIF
               ENDIF
             ENDIF
C
C**   The chamber is a reserved chamber.
           ELSE
             IDIFF = NINT(A(IPROD,17)) - NINT(A(IPROD,16))
             IF (IDIFF.LT.NINT(A(IPROD,18))) THEN
               NRESV(K) = 0
               NCH(K) = 1
               IF (A(IPROD,9).LE.0.0) THEN
                 TCSTRT(K) = TIMNOW
               ELSE
                 IF (NUMBER(K).EQ.1) THEN
                   TC = TCSTRT(K+1) + A(IPROD,9)
                 ELSE
                   TC = TCSTRT(K-1) + A(IPROD,9)
                 ENDIF
                 TCSTRT(K) = AMAX1(TIMNOW,TC)
               ENDIF
               TCFIN(K) = TCSTRT(K) + A(IPROD,4)
               TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
               TWET(K) = TFREEZ(K) - A(IPROD,3)
               SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
               CRT = A(IPROD,2) - 1.0
               CALL SETA (IPROD,2,CRT)
               NNWAS = NNWAS + 1
               NEWASS(NNWAS) = K
               RES = A(IPROD,18) - 1.
               USED = A(IPROD,17) + 1.
               CALL SETA(IPROD,18,RES)
               CALL SETA(IPROD,17,USED)
             ELSE
               GO TO 1630
             ENDIF
           ENDIF
 1630    CONTINUE
C
C**   Schedule batches to pair idle chambers (if needed).
C
1640     IF (NINT(A(IPROD,2)).LE.0) GO TO 1660
         JADD = NINT(A(IPROD,16)) - NINT(A(IPROD,17))
```

```
      IF (JADD.LE.0) GO TO 1660
C
C**   Find pair idle chambers of group M and schedule two batches.
      II2 = 0
      DO 1642 J = 1,NUMCH
        K = IDLE(J)
        IF (NGROUP(K).EQ.M) THEN
          IF ((NCH(K).EQ.0).AND.(NBLOCK(K).EQ.0)) THEN
            IF (NUMBER(K).EQ.1) THEN
              IF ((NCH(K+1).EQ.0).AND.(NBLOCK(K+1).EQ.0)) THEN
                II2 = II2 + 1
                IDL2(II2) = K
              ENDIF
            ELSE
              IF ((NCH(K-1).EQ.0).AND.(NBLOCK(K-1).EQ.0)) THEN
                II2 = II2 + 1
                IDL2(II2) = K
              ENDIF
            ENDIF
          ENDIF
        ENDIF
 1642 CONTINUE
      IF (II2.LE.0) GO TO 1660
      DO 1650 L = 1,II2
        K = IDL2(L)
        IF (NUMBER(K).EQ.0) GO TO 1650
        IF ((NCH(K).EQ.1).OR.(NBLOCK(K).EQ.1).OR.
     1  (NCH(K+1).EQ.1).OR.(NBLOCK(K+1).EQ.1)) GO TO 1650
        NCH(K) = 1
        TC = 0.+ A(IPROD,3)+ A(IPROD,21)
        TCSTRT(K) = AMAX1(TIMNOW,TC)
        NCHPR(K) = NINT(A(IPROD,1))
        TCFIN(K) = TCSTRT(K) + A(IPROD,4)
        TFREEZ(K) = TCSTRT(K) - A(IPROD,21)
        TWET(K) = TFREEZ(K) - A(IPROD,3)
        SUMCH1(K) = SUMCH1(K) + A(IPROD,4)
        CRT = A(IPROD,2) - 1.0
        CALL SETA(IPROD,2,CRT)
        NNWAS = NNWAS + 1
        NEWASS(NNWAS) = K
        IF (NINT(A(IPROD,2)).GT.0) THEN
          NCH(K+1) = 1
          NCHPR(K+1) = NINT(A(IPROD,1))
          IF (A(IPROD,9).LE.0.) THEN
            TCSTRT(K+1) = TCSTRT(K)
          ELSE
            TCSTRT(K+1) = TCSTRT(K) + A(IPROD,9)
          ENDIF
          TCFIN(K+1) = TCSTRT(K+1) + A(IPROD,4)
          TFREEZ(K+1) = TCSTRT(K+1) - A(IPROD,21)
          TWET(K+1) = TFREEZ(K+1) - A(IPROD,3)
          SUMCH1(K+1) = SUMCH1(K+1) + A(IPROD,4)
          CRT = A(IPROD,2) - 1.0
          CALL SETA(IPROD,2,CRT)
          NNWAS = NNWAS + 1
          NEWASS(NNWAS) = K + 1
        ENDIF
        USED = A(IPROD,17) + 2.0
        CALL SETA(IPROD,17,USED)
        IF (A(IPROD,22).LT.-5.) THEN
          CALL SETA(IPROD,22,TWET(K))
        ENDIF
```

```
            GO TO 1640
 1650     CONTINUE
C
C
C**   Find number of chambers actually used.
 1660     NCBR = 0
          DO 1670 J = 1,18
            IF ((NCHPR(J).EQ.NINT(A(IPROD,1))).AND.(NCH(J).EQ.1).AND.
     1        (NBLOCK(J).EQ.0)) THEN
              NCBR = NCBR + 1
            ENDIF
 1670     CONTINUE
          IF (NCBR.LE.0) THEN
            GO TO 1600
          ELSE
            NBR = NBR + 1
            MPR(NBR) = NINT(A(IPROD,1))
            NCHBR(NBR) = NCBR
          ENDIF
 1600   CONTINUE
C
 1400 CONTINUE
      NLACK = 0
      IF (NBR.GT.0) THEN
        CALL DRASSN
      ENDIF
C
      IF (NNWAS.LE.0) GO TO 1114
C
      WRITE(10,'(/2X,A,F6.2)')
     1'Scheduling product(s) on chamber(s) at time ',TIMNOW
      WRITE(10,'(5X,A)')
     1' Prod    Strt.time   Strt.time       Chamber   Strt    Compln'
      WRITE(10,'(5X,A)')
     1'number   at wetside  at cold rm      number    time     time'
      DO 1112 MM = 1,NNWAS
        K = NEWASS(MM)
        WRITE(10,'(5X,I4,2F12.2,I14,F11.2,F9.2)')
     1  NCHPR(K),TWET(K),TFREEZ(K),K,TCSTRT(K),TCFIN(K)
 1112 CONTINUE
C
 1114 CALL TIME
      RETURN
      END
C
**********************************************************************
      FUNCTION LINT(AREAL)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
C
C**   Round up AREAL to the smallest integer that larger than AREAL
      IF (ANINT(AREAL).LT.AREAL) THEN
        LINT = NINT(AREAL) + 1
      ELSE
        LINT = NINT(AREAL)
      ENDIF
      RETURN
      END
```

```
C
*************************************************************************
      SUBROUTINE WRAPUP
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER/NCHPR(18),NCH(18),NGROUP(18),TCSTRT(18),TCFIN(18),
     1TIMNOW,TIMNXT,MPR(6),NN,KPRDR(20,4),TPROC,NCHBR(6),NBR,NDRPR(5),
     2TDSTRT(5),TDFIN(5),TDSET(5),NBLOCK(18),TCBLK(18),NC1R,NC2R,
     3NC1ASS,NC2ASS,NPROD1,NPROD2,NRESV(18),INPR1,INPR2,SUMCH1(18),
     4SUMCH2(18),SUMDR1(5),SUMDR2(5),NUMBER(18)
C
      WRITE(10,'(///2X,A/)')'SUMMARY REPORT'
C
C**   Makespan.
      WRITE(10,'(2X,A,F8.2,A/)')'*  Makespan : ',TPROC,' hours.'
      WRITE(10,'(2X,A)')'*  Chambers.'
C
C**   Chamber utilizations and blockages.
      WRITE(10,'(/8X,A)')
     1'Chamber       Utilization      Blockage      Idle-time'
      WRITE(10,'(8X,A)')
     1'number        percentage      percentage    percentage'
      WRITE(10,'(8X,A)')
     1'-------------------------------------------------------'
       DO 42 K = 1, 18
        UTIL = (SUMCH1(K) / TPROC) * 100
        BLOCK = (SUMCH2(K) / TPROC) * 100
        PIDL = 100. - (UTIL + BLOCK)
        WRITE(10,'(5X,I7,F17.2,F16.2,F15.2)')K,UTIL,BLOCK,PIDL
        SCUTL = SCUTL + UTIL
        SCUSQ = SCUSQ + (UTIL**2)
        SCBLK = SCBLK + BLOCK
        SCBSQ = SCBSQ + (BLOCK**2)
   42 CONTINUE
      WRITE(10,'(8X,A)')
     1'-------------------------------------------------------'
      AVCUTL = SCUTL / 18
      SDCU = SQRT((SCUSQ - 18*(AVCUTL**2)) / 17)
      AVCBL = SCBLK / 18
      SDCB = SQRT((SCBSQ - 18*(AVCBL**2)) / 17)
      WRITE(10,'(/8X,A)')'The utilization percentage of chambers.'
      WRITE(10,'(10X,A,F10.2)')'Average            : ',AVCUTL
      WRITE(10,'(10X,A,F10.2)')'Standard deviation : ',SDCU
      WRITE(10,'(8X,A)')'The blockage percentage of chambers.'
      WRITE(10,'(10X,A,F10.2)')'Average            : ',AVCBL
      WRITE(10,'(10X,A,F10.2)')'Standard deviation : ',SDCB
C
C**   Dumping room utilization.
      WRITE(10,'(/2X,A)')'*  Dumping rooms.'
      WRITE(10,'(/8X,A)')
     1'D-room       Utilization      Clean & Setup      Idle-time'
      WRITE(10,'(8X,A)')
     1'number       percentage       percentage         percentage'
      WRITE(10,'(8X,A)')
     1'------------------------------------------------------------'
       DO 44 N = 1, 5
        UTIL = (SUMDR1(N) / TPROC) * 100
        CLEAN = (SUMDR2(N) / TPROC) * 100
        PIDL = 100. - (UTIL + CLEAN)
        WRITE(10,'(5X,I7,F16.2,2F18.2)')N,UTIL,CLEAN,PIDL
        SDUTL = SDUTL + UTIL
        SDUSQ = SDUSQ + (UTIL**2)
        SDCLN = SDCLN + CLEAN
```

```
         SDCLSQ = SDCLSQ + (CLEAN**2)
   44 CONTINUE
      WRITE(10,'(8X,A)')
     1'-----------------------------------------------------------'
      AVDUTL = SDUTL / 5
      SDDU = SQRT((SDUSQ - 5*(AVDUTL**2)) / 4)
      AVDCL = SDCLN / 5
      SDDC = SQRT((SDCLSQ - 5*(AVDCL**2)) / 4)
      WRITE(10,'(/8X,A)')'The utilization percentage of dumping rooms.'
      WRITE(10,'(10X,A,F10.2)')'Average               : ',AVDUTL
      WRITE(10,'(10X,A,F10.2)')'Standard deviation : ',SDDU
      WRITE(10,'(8X,A)')
     1'The cleaning & setup percentage of dumping rooms.'
      WRITE(10,'(10X,A,F10.2)')'Average               : ',AVDCL
      WRITE(10,'(10X,A,F10.2)')'Standard deviation : ',SDDC
C
C**   Time in systems.
      WRITE(10,'(/2X,A)')'*  Time in systems.'
      WRITE(10,'(/8X,A)')
     1'Product      Time in system'
      WRITE(10,'(8X,A)')
     1'number           (hours)'
      WRITE(10,'(8X,A)')
     1'-------------------------'
      DO 46 I = 1, NN
        IF (I.EQ.1) THEN
          IPROD = LFR(6)
        ELSE
          IPROD = LSUCC(IPROD)
        ENDIF
        LPD = NINT(A(IPROD,1))
        TSYS = A(IPROD,23) - A(IPROD,22)
        WRITE(10,'(8X,I4,F18.2)')LPD,TSYS
        TOTSYS = TOTSYS + TSYS
        IF (A(IPROD,23).GT.168.) THEN
          NTARDY = NTARDY + 1
          TARDNS = TARDNS + (A(IPROD,23) - 168.)
        ENDIF
   46 CONTINUE
      WRITE(10,'(8X,A)')
     1'-------------------------'
      AVTSYS = TOTSYS / 13
      WRITE(10,'(/8X,A,F8.2,A)')
     1'The average time in system of products : ',AVTSYS,' hours.'
C
C**   Tardiness.
      WRITE(10,'(/2X,A)')'*  Tardiness.'
      AVTRDN = TARDNS / 13
      WRITE(10,'(8X,A,F8.2)')'Average tardiness of products : ', AVTRDN
      WRITE(10,'(8X,A,I8)')'Number of tardy products       : ',NTARDY
C
      CLOSE(10)
      RETURN
      END
```

Appendix I-B : Experimental frame for computer model.

```
BEGIN;
PROJECT,OFD,CINDY;
DISCRETE,35,23,7;
RANKINGS:1-4,LVF(12):6,LVF(1);
REPLICATE,1,0,50;
END;
```