AN ABSTRACT OF THESIS OF

Nivruti Rai for the degree of Master of Science in Industrial and Manufacturing Engineering presented on October 12, 1993.

Title: Design and Implementation of a Production Scheduling System for Continuous Manufacturing Environment.

Abstract approved: Redacted for Privacy

Dr. Sabah Randhawa

Scheduling a continuous manufacturing flow shop environment with several machines, stochastic arrival of demands, different product requirements and limited resources is a complex task. This research develops a methodology for scheduling products in glass fiber industry. The system consists of two components. An optimizing linear program determines the optimal solution for a sub-problem that accounts for safety stocks, demands, machine capacities, and due dates. The job queues from the LP model are then sequenced based on 'earliest job due date' for machines that have two or more jobs to be performed on the same time. This sequenced solution is then input to a simulation model. The simulation model prioritizes the queue of jobs on each machine so that minimum rate of change of throughput is achieved, while satisfying the due dates. The model was validated for a major fiber galss manufacturer. The results show that the use of an integrated optimizing and heuristic solution system provides better results than current scheduling practice in terms of machine utilization, deviations from target inventories, and on-time jobs.

Design and Implementation of
A Production Scheduling System
For Continuous Manufacturing Environment

by

Nivruti Rai

A THESIS

submitted to

Oregon State University

in partial fulfillment of

the requirements for the degree of

Master of Science

Completed October 12, 1993

Commencement June 1994

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 October 12, 1993

Typed by Nivruti Rai for Nivruti Rai

ACKNOWLEDGEMENTS

I would like to thank Prof. Sabah Randhawa for his help and guidance during this thesis and my graduate studies at Oregon State University. The guidance of Prof. E. Mcdowell and Prof. E. Olsen was invaluable in completing this thesis.

This thesis would not have been possible without the help of my friends in Corvallis, especially Baljinder Singh Ghotra.

The encouragement and support of my mother and father have been important in making this thesis and all of my endeavours possible. I would especially like to thank my imminent baby for not demanding attention during this busy period. Specially, I want to thank my husband Sunit who has not only been helpful, supportive, and extremely loving but also very encouraging.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES TABLES

Design and Implementation of a Production Scheduling System for Continuous Manufacturing Environment

CHAPTER 1

INTRODUCTION

1.1 CONTINUOUS MANUFACTURING SYSTEMS:

A continuous manufacturing system includes both "continuous' and "intermittent" systems. Examples of "continuous" systems include petrochemicals, steel and paper production; "intermittent" systems include paint industry and some metallurgical operations. This research pertains to glass fiber industry. Here the production is in lots (or batches) from different raw materials. A continuous manufacturing system increases production due to process layout resulting in fewer interruptions in production and setup changes. However, production scheduling can be a complex process, particularly in intermittent operations. Such systems require production scheduling that can meet quality and timing goals, and be able to handle changes in demands and raw material mix.

Production in a continuous manufacturing system is planned over a future interval of time, called the 'planning horizon'. During this period the rate of demand for the different product varies. It is assumed that this time interval is divided into periods and that the planning problem is to determine production rate for each period in the planning horizon. The demand rate in each period is assumed to be known, but is not necessarily constant from period to period. When the demand rate varies from one time period to the next, the system is called a dynamic system. Production scheduling involves determining a product schedule that meets product delivery due dates while taking into account several related problems such as a) minimizing the down-time of equipment, b) meeting demand, and c) capacity and production constraints.

1.2 PRODUCTION AND SCHEDULING IN A CONTINUOUS MANUFACTURING ENVIRONMENT:

In production scheduling both resource allocation and sequencing of jobs is important. Production allocation determines the amount of each product to be produced by each machine and product inventory levels. Sequencing specifies the processing order of jobs based on specified priorities, and stabilizes the equipment's usage. Scheduling problems have been broadly classified into two groups as shown in Figure 1 [1]. These are: i) static or deterministic type of problems, and ii) dynamic or stochastic problems. In static or deterministic problems all jobs to be processed are available at the start of the scheduling period, whereas in dynamic problems the jobs keep arriving throughout the scheduling period following a stochastic distribution. These two classifications are further subdivided into single stage problems, where the system consists of only one machine type, or multistage problems, where the system has several machine types. Multistage problems are further classified as parallel, series and hybrid, depending on identical machine combinations which allow similar jobs to be processed.

For a finite set of tasks the utilization of resources is inversely proportional to the time required to accomplish all the tasks [2]. This time is referred to as the make span of the schedule. In general, production and scheduling objectives are as follows: i) reduce the make span, ii) reduce the deviations from the target inventories, iii) reduce the in-process inventory levels, iv) reduce tardiness, v) reduce the setup times, and

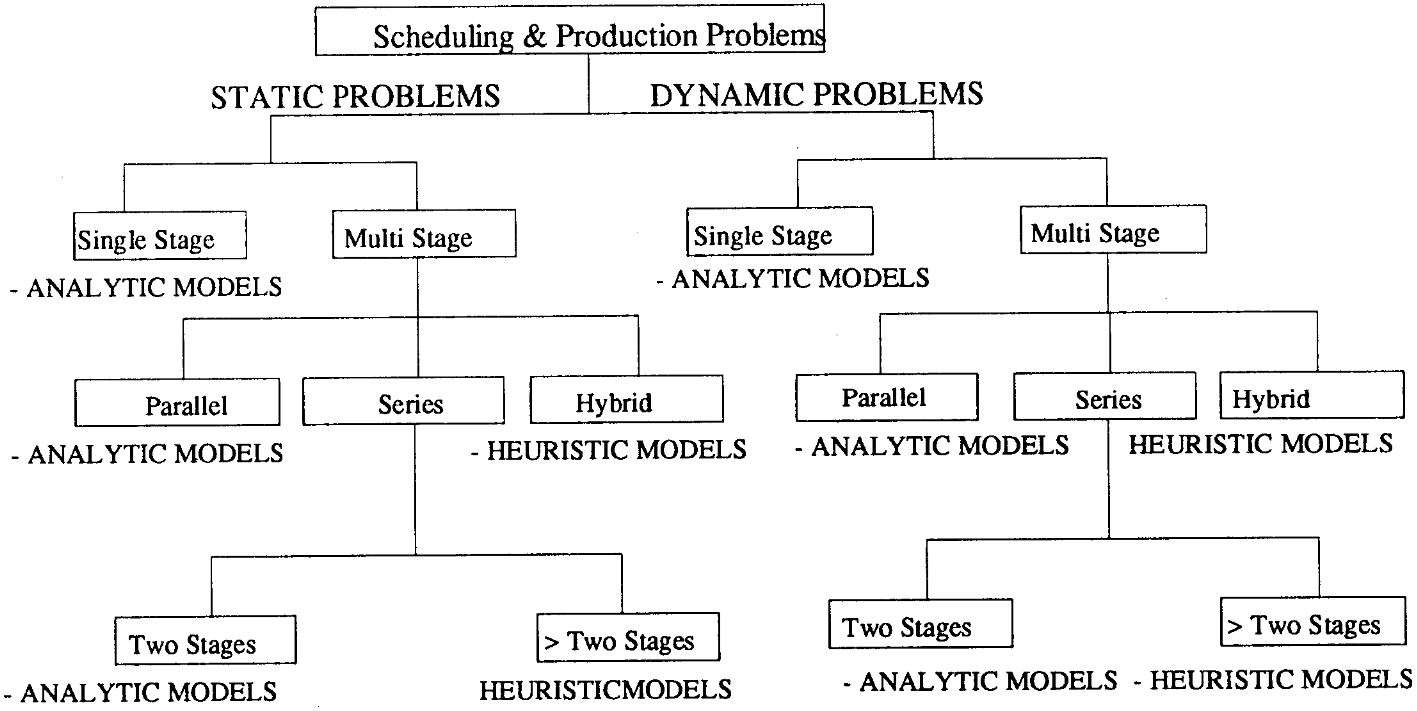

Figure 1 : Classification of scheduling problems, models.

vi) reduce machine overload.

Typical real-world problems are often very complex. It is not unusual to address all the above objectives separately. The scheduler can deal with assignment and sequencing or job shop scheduling in a variety of ways. The simplest is to ignore the problem and constraints and accomplish the tasks in a random order. A number of quantitative approaches to several types of scheduling problems have been proposed. For 'production allocation' linear programming model is often used, while heuristic-based simulation models have been frequently used for 'sequencing' [3]. To use mathematical programming models, availability of the raw materials and inventory is specified. In addition, for each task the constraints as to when a task can be started, the due date, the processing time, and the amount to be manufactured have to be determined. A large set of heuristic scheduling decision rules are reported in published literature; it has been shown that for both static and dynamic problems, the 'Shortest Processing Time' scheduling rule gives a better or comparable performance as more complex heuristic decision rules [1,2,4].

1.3 RESEARCH OBJECTIVES:

The objective of this research project is to develop a scheduling decision support system for a major fiber glass production facility. The system integrates a linear programming (LP) model with a heuristic-based simulation system. Production of different fiber glass products takes place on two independent lines each having multiple settings. The demands for various products is assumed to follow a stochastic distribution over a future time period horizon. The system under consideration is thus a multi-stage parallel machine setting system.

The current production scheduling technique used by the fiber glass production facility, is based on a simple spreadsheet system. Products are scheduled if they have a forecasted demand and amount of each product is determined based on the quantity required and the available inventory. Rest of the corporations scheduling system is proprietary. This method is neither efficient in terms of time and resource utilization, nor does it assure optimality or justification of the process used. The objective of this project is to develop an integrated scheduling system for glass melting process to minimize cost of production, in terms of over or under production that means excessive storage costs or loss of profit if orders are not met.

The glass fiber production and scheduling problem approach uses a stepwise decision process. The first decision is the amount of products to be produced. Once this is completed each machine has a number of tasks waiting to be run. The next decision is the order in which the waiting tasks should be processed. These two decisions, called assignment and sequencing, are referred to as job shop scheduling. A complete flow of information is shown in Figure 2.

The thesis is organized as follows. Chapter 2 provides a brief review of the fiber glass production process, production and scheduling constraints, and a discussion of scheduling reported in the literature. In Chapter 3, the production scheduling model is described in detail. Chapter 4 discusses the results obtained from the model, and finally Chapter 5 summarizes the work performed.

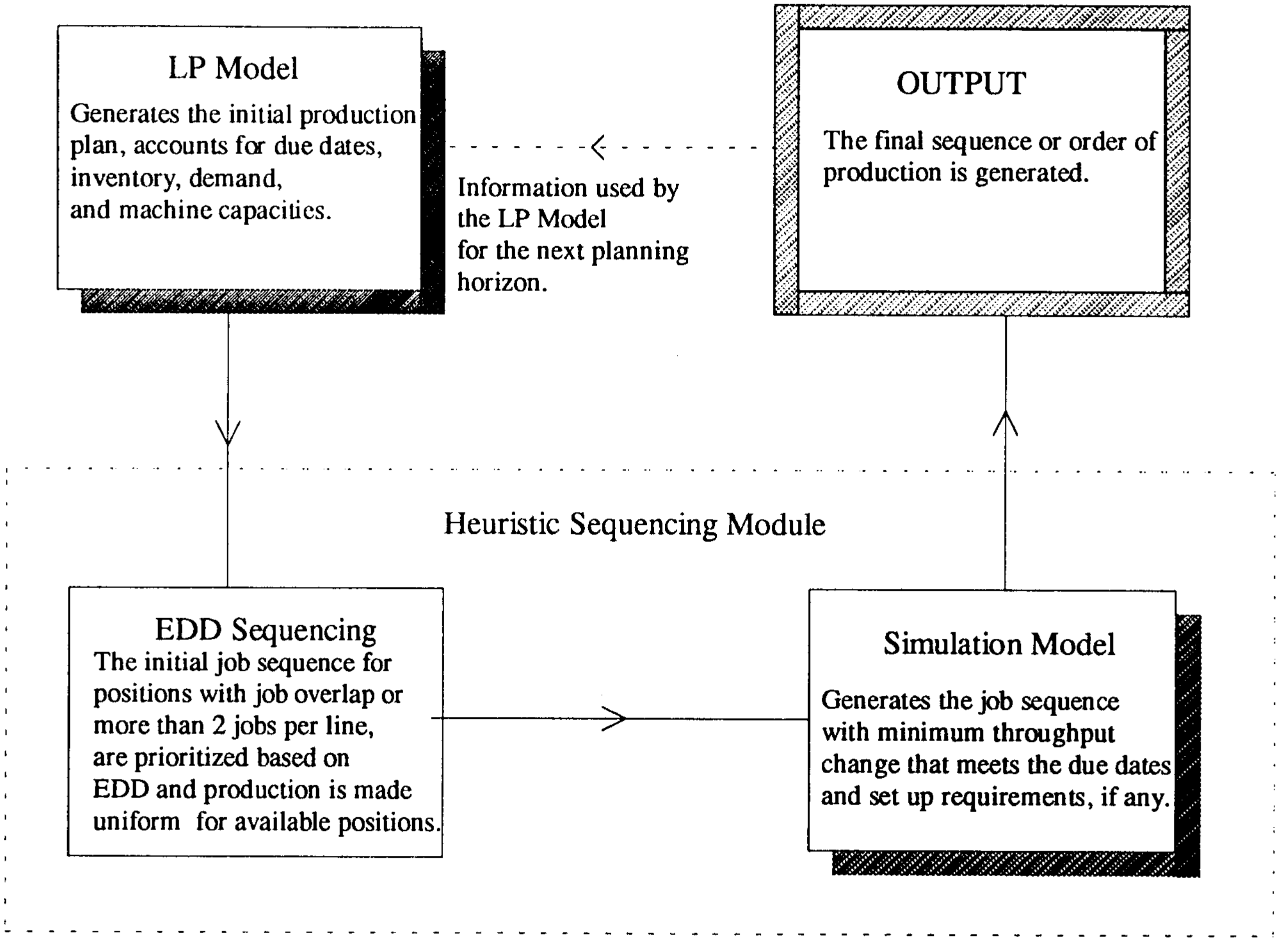

Figure 2. Modeling Approach.

CHAPTER 2

PROCESS DESCRIPTION AND LITERATURE SURVEY

In this chapter a brief description of the fiber glass manufacturing process is provided. Then the complexity of the scheduling problem and different production constraints are presented. Finally, similar scheduling problems reported in the literature are discussed.

2.1 GLASS FIBER MANUFACTURING PROCESS:

A brief description of the manufacturing process is given here. Details of the process specific to the production facility analyzed are not provided due to the proprietary and sensitive nature of this information.

2.1.1 Types of Fibers:

Glass fiber is made in two basic forms, continuous filaments (0.5 to 15 μm diameter) and fibers (6 to 9 μm diameter and 20 to 40 cm long). Single filaments can be combined into a strand that is easily unwound from a spool. The product is a twisted textile yarn. Staple fibers that meet the diameter and minimum length requirements are twisted together into a tow that can also be easily unwound from a spool. The end product is staple yarn. Glass fibers that do not meet these requirements are not suited for textile applications, but may be used for thermal and sound insulation in the form of glass wool. This study pertains to a system used for the manufacture of Glass Micro-

Fiber which is an extremely pure, inert borosilicate glass wool available in a wide variety of grades and chemical compositions, ranging from 0.5 to 9.7 μm (microns) in average fiber diameter (AFD) [5]. The glass micro-fiber is used in a variety of ways based on their AFD [6]:

1) Economical fiber for paper enhancements, alternative for mineral and asbestos.

2) Standard quality fiber is used in non critical filtration as automotive fuel, hydraulic fluid, food and beverage industry, and medical applications.

3) Special acid resistant fiber are used for sealed lead acid battery separator applications.

4) High quality fiber is used in high performance filtration media.

2.1.2 Process Description:

Glass of different chemical properties is fed into a melter. The molten glass is then pulled and a fiberizer is used to make a fiber. The classification of the products is based on the diameter of the fiber glass and its chemical properties, as discussed above. The production specifications for each product are defined in terms of pull rate, or the amount of glass the fiberizer pulls from the melter, the mode of collection, the fiberizing process required, and packaging system used. Thus, molten glass is pulled into different manufacturing positions and then various fiberizers are used on each positions based on the product that is to be manufactured. A transition product with a combined chemistry also results during the melting process, but the production of transition product can be controlled based on market requirements [7].

Figure 3 shows the system being studied. There are two available product lines. Each line has a set of seven positions and a "patty" (by-product) manufacturing position. Glass is fed into these positions from their melters. The capacities of

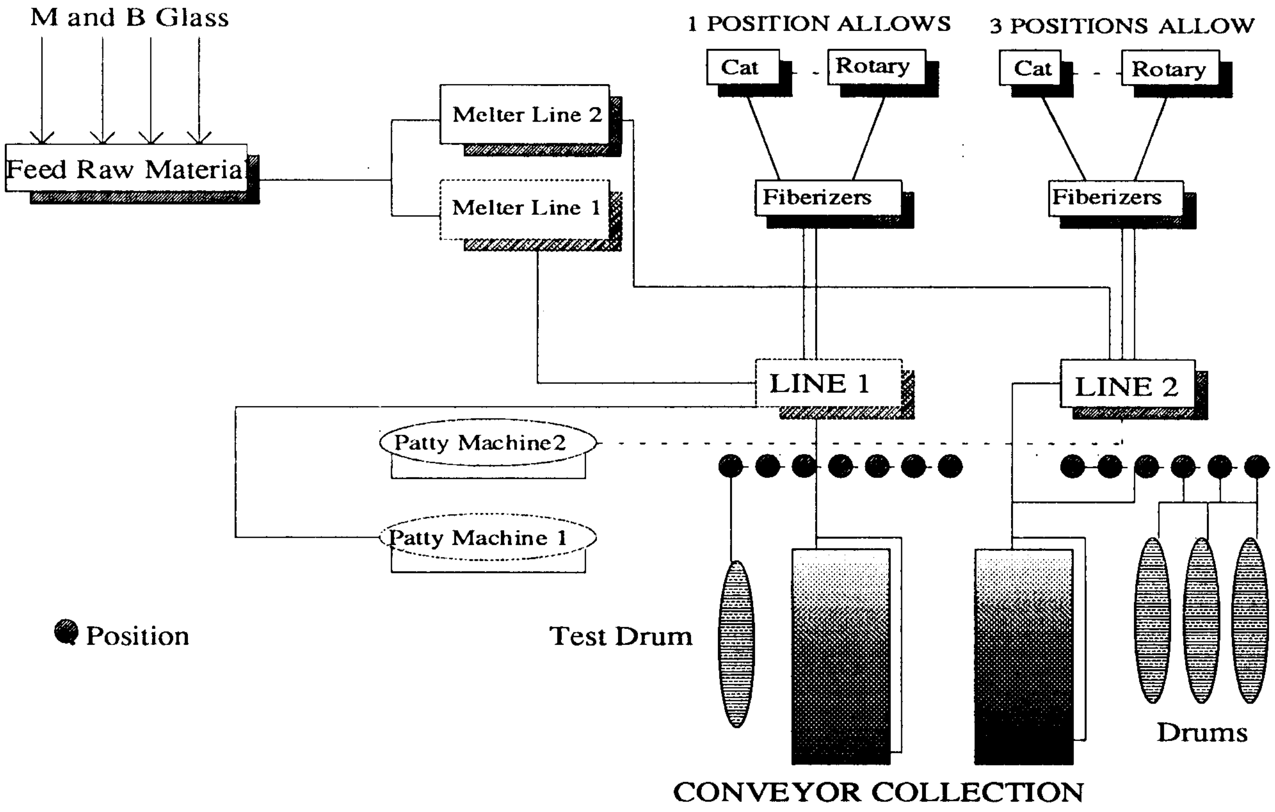

FIGURE: 3 SCHEMATIC OF THE ROTARY PRODUCTION SYSTEM the melters for the two lines vary. The glass fiber system has two types of fiberizers : a) Cat, and b) Rotary. The products manufactured by the Cat fiberizer require the 'drum mode' of transfer to move the fiber glass to the bailing (or packaging) stations. The drum collection system involves rolling of the fiber glass onto wooden drums. Rotary fiberizer uses the drum mode only for finer products; a ski-conveyor system is used for coarser products. The ski-conveyor allows the sheets of glass fibers to be transferred in the form of sheets. In general the cat fiberizer is used for the finer products. Line 2 has three flexible positions that can be switched between Cat to Rotary fiberizers; and Line 1 has only one. The collection modes for both the manufacturing lines are shown in Tables 2.1 and 2.2.

Table 2.1 : Collection Modes for Line 1.

Position	Collection Mode
1	Drum
2	Conveyor
3	Conveyor
4	Conveyor
5	Conveyor
6	Conveyor
7	Conveyor
8	Patty

Table 2.2 : Collection Modes for Line 2.

Positions	Collection Mode
9	Patty
10	Drum
11	Drums
12	Drum
13	Conveyor
14	Conveyor
15	Conveyor

2.2 PROCESS CONSTRAINTS AND PROBLEM COMPLEXITY:

2.2.1 Process Constraints:

Several constraints affect the job assignment. Some of the important ones are:

1) No more than two regular products and a by-product 'patty' (used as a raw material for another process) can be made at any given time on each line.

2) If a high energy consuming or a thick diameter product is being produced, then idling of certain positions is required such that the threshold pull rate is not exceeded. Idling refers to letting the position run without pulling any glass from the melter. Idling is preferred over downtime of a position because downtime results in glass deposition, which is difficult to remove.

3) Product demands and current inventory levels dictate the production and scheduling of products.

2.2.2 Problem Complexity:

A number of product and process features makes the system complex to schedule and control. These include:

1) The pull rate of all the products is different for the two lines.

2) Inventory levels for the raw material, work-in-process, the amount of transition and finished products must be controlled, and the demand for finished products must be satisfied.

3) Idling of the positions necessitated during the production of a high pull rate product, must be minimized.

4) Switching of the fiberizer between cat and rotary requires a substantial set up time. Therefore, switching between cat and rotary fiberizing position, needs to be minimized.

5) Manufacture of the transition material in switching between products need to be minimized.

6) The total amount of capacity for collecting fiber glass is limited, and maximizing its utilization is desired.

7) Each product has specifications that require the use of a specific type of position.

2.2.3 Assumptions:

The demand for different products arrive at various times, thus the system is dynamic in nature. The following assumptions are made:

1) Inventory is updated when a job is scheduled.

2) A time unit of four days is used, and a non-integral time unit can be subsequently converted to obtain a particular date. This was done in order to cover a planning horizon of one month, while keeping the number of decision variables within

computational limits for the optimization software available on personal computers.

3) Information about products and the positions on which they can be manufactured are deterministic for this study but can be changed as and when the collection modes change.

2.3 LITERATURE REVIEW:

2.3.1 Glass Fiber Production:

The manufacture of glass fibers is done primarily as continuous filaments that can be combined as a yarn. This yarn can be used for electrical and heat resistant applications [8]. Discontinuous fibers can be used as textiles for the various purposes as thermal and electrical insulation's.

Commercial glass fibers are based on silica, which is a glass former. When silica is heated until fluid and then cooled, it forms a glass. In glass silica (SiO_4) exists as a polymer $(SiO_2)_n$; it does not have a melting point but gradually softens as the temperature is raised to 2000 °C. Commercial glasses are made by melting/fusing a mixture of materials in the temperature ranges of 1300 °C to 1500 °C. In these glasses, the silica network has been modified by the addition of other oxides. These oxides include a) Network formers such as boric oxide (B_2O_3) and alumina (Al_2O_3), b) Network modifiers such as di-sodium oxide (Na_2O), di-potassium oxide (K_2O), and c) Intermediate oxides, such as magnesium oxide (MgO). Network formers replace a Si atom but maintain the basic glass-like nature of the network. Network modifiers break down the Si-O-Si bonds thereby reducing the temperature of glass melting and forming. Intermediate oxides can work both ways as network former or modifier depending on the composition and temperature history [11].

The following factors are important for fiber glass composition formulation: i) properties and quality requirements of fibers, ii) working characteristics for glass forming and fiberizing iii) chemical durability and resistance, and iv) cost. Major composition types used for fibers include alkali borosilicate glasses, and calcium alumino-borosilicate glasses [12].

Glass Fibers are made by first melting the glass in furnaces. The furnaces are either a) fuel fired melters, or b) electric furnaces. Electric furnaces are preferred for fluoride, boron, and lead containing glasses, as use of a fuel fired furnace will lead to loss of volatile components. To make continuos filaments, the molten glass is either fed directly or in form of rods, to an electrically heated platinum/rhodium bushing with cylindrical nozzles. Glass coming out of these nozzles is mechanically pulled into fibers from 4 to 20 μm (μm equals micron or 10^{-6} meter) in diameter at speeds in excess of 5000 ft/minute. The glass wool fibers are made either using a) Rotary fiberizer or b) Flame blown process. The rotary wool fibers are produced similar to cotton candy. Molten glass falls into a hollow rotating cylinder, called the 'spinner', which has many holes in its vertical sidewall edge. Centrifugal force of the rotating spinner forces the glass to extrude laterally out of the numerous perforations. These fibers are discontinuous with diameter ranging from 0.5 μm to 7 μm. In the flame blown process coarse fibers are pulled through rollers and are thinned by high kinetic energy of a hot gas stream [13,7,11,12].

2.3.2 Production Scheduling in Continuous Manufacturing Systems:

Production scheduling in a continuous environment has been studied to facilitate the production scheduling decision making [1,2]. Scheduling has been considered as a discrete event continuous time decision making process. The fiber glass production system is a dynamic single-stage system, with forecasted demands available at the onset of planning. Problem solving and scheduling for similar systems has been reported in literature. Some examples of these will now be discussed.

Use of simulation tools to support regular repetitive decision making and production planning and control are discussed by Rogers and Flanagan [3]. Simulation is used for setting long term goals such as master production schedule, for medium term planning such as materials requirements plans, and for production activity control (PAC). PAC monitors the progress of plans, making short term modifications required by changes in shop floor status. This technique provides real time scheduling decisions in a dynamic manufacturing environment allowing planning around unexpected events on the shop floor. However, in order to make valid simulations it is important to model the shop floor accurately and in detail, with access to complete information such as the material requirements plan. The speed and cost of simulation as well as the confidence in the results influence the use of this technique.

Scheduling in a dynamic process plant using expert systems has been reported in [14,15]. Using rule-based expert system methodologies and extensive data processing, a scheduling system is resident on shop floor allowing definition of long and short term planning. Automated data transfer between the scheduling system and process control environment allows decision making to incorporate complete shop floor information. The decision making logistics and operating rules are entered in forms of if-then-else statements and the recommended action is based on heuristics. Such systems have been used for production planning, troubleshooting, and operator training [14]. Production scheduling is well described by a set of event driven activities operating on a global database, thus a rule based approach is used [15]. Expert systems, which provide a certain amount of reasoning capability and expert knowledge, are useful for scheduling in flexible manufacturing systems as no direct algorithmic solution is feasible for this complex, ill-structured problem. Knowledge representation and problem solving are amenable to expert system techniques. The scheduler follows simple guidelines given by the priority of lots, which is determined by its urgency and process effect on the manufacturing system. The priority is computed as a function of remaining time, due-date, and release time. The scheduler introduces lots into the systems meeting availability, demand and any suitable resource constraints [15].

The use of neural network approach for job-shop scheduling is reported in [16]. Using linear energy functions, a scheduling problem is mapped onto a simple neural net where the number of neural processor equals the number of subjobs, and interconnections grow linearly with total number of operations [16].

On-line scheduling of real-time tasks has been reported in [17]. The scheduler uses an extension of the deadline algorithm, where at any time t, the scheduler schedules that active task whose deadline is closest to t. It can be shown that such a schedule is optimal for a single processor. However, the paper reports an optimal algorithm for multi-processor system with one common deadline. It is also shown that no optimal on-line scheduler can exist for multi-processor systems with two deadlines.

Use of dynamic programming for linear scheduling problems is reported in [18]. Two state variables are used for each activity: the duration required to complete work at a location, and possible interrupt duration's between work at adjacent locations. The overall project duration is minimized. This approach is useful for sequencing of tasks, taking into account any precedence relationships.

Linear programming provides optimal solutions, and has been used to best utilize available resources and scheduling of recycling tasks [19] in waste recycling applications. Linear programming has also been extensively used to minimize production costs, and optimize resource utilization in fishing industry [20].

CHAPTER 3

METHODOLOGY

3.1 APPROACH:

A combination of optimization and heuristic approaches is used to model the problem. This approach is used to increase resource utilization, reduce the in-process inventory, reduce job tardiness, reduce deviations from the target inventory, and reduce the rate of change of throughput from the melter into the manufacturing lines. Four factors describe the state of system and are the primary inputs to the model:

1) The job arrival pattern: the arrival of jobs is assumed available at the start of the scheduling period.

2) The total number of positions on each manufacturing line.

3) The flow process of jobs.

4) The criterion for evaluating the performance of the shop. The complex nature of the present production system necessitates the use of two smaller sub-models rather than a single model to obtain a feasible and acceptable solution (Figure 2). The two sub-models are required because available software packages cannot efficiently handle the original problem.

First a linear programming (LP), model is used to obtain quantities of each product to be produced on each position and near target inventory levels, while minimizing tardy jobs and maximizing melter utilization.

The results from LP are used as an input to the heuristic sub-model 2. Overlapping jobs or more than two jobs per line are ordered based on earliest due date and production is uniformly distributed on all available positions. This minimizes idling of the positions. The heuristic-based simulation model then considers all the possible sequences and selects the sequence which reduces the rate of change of throughput, while meeting the due dates or giving priority to the earliest due date.

3.1.1 Resources Used:

The following computer resource were used for the two sub-models.

1) The solution of the mathematical program was obtained using the LINDO software [21] (Linear INteractive Device for Optimization) for a 386 or higher microprocessor based personal computer system.

2) The simulation of the system was accomplished by writing a program in C, run in Borland C++ [22] environment.

Details of these two sub-models are given below.

3.2 SUB-MODEL I: OPTIMIZATION MODEL

The linear program (LP) determines the initial schedule which reduces the make span and inventory, and meets demands in order to reduce tardy jobs. The decisions made in production planning affect several classes of costs and revenues. Production costs involve fixed production cost and costs incurred due to over- or under-production. The planning decisions also affect the inventory levels, too much or too little inventory mean extra holding costs or costs due to unforeseen demands. Failure to meet customer demand on time may result in tangible loss of profit and intangible loss of good will. Hence, results from the linear program would also ensure lower production costs or higher expected profits. The formulation of the linear program is described now:

Indices

The indices used in the linear programming model are:

i = classification of fiber glass products.

j = manufacturing positions on the two lines.

t = time unit of arrival time of demand of product.

Decision Variables

The decision variables used are:

S_{ijt} = Start time for production of product i, on position j, the demand of which came at time t.

E_{ijt} = End time for production of product i, on position j, the demand of which came at time t.

x_{ijt} = Amount of product i, scheduled on position j at time t.

I_{it} = Inventory of product i at time t.

Ux_{it} = Amount of inventory of i less than the maximum inventory at time t.

On_{it} = Amount of inventory over the minimum or buffer inventory at time t.

E_{it} = Maximum end time of production of i, the demand for which came at time t, for all positions j.

S_{it} = Minimum start time of production of i, the demand for which came at time t, for all positions j.

Sp_{it} = Number of days beyond the due date for product i, the demand for which came at time t.

Sn_{it} = Number of days before the due date for product i, the demand for which came at time t.

Input Variables

Input variables or constants in the model are:

1) Capacity of melter.
2) Forecasted demand for each time period, D_{it}.
3) Requirement for product i at time t or the due dates, d_{it}.
4) Amount of inventory on hand at the start of production, I_{i0}.
5) The safety stock required for each product.
6) Maximum allowable inventory for each product.
7) Transition from a M type to a B type product takes a setup of 2 to 7 days or on an average of 1.12 time units.
8) Transition from a B type to a M type product takes a setup of 2 days or 0.5 time units.
9) Collection capacity of each conveyor: 500 lbs/hr.
10) The arrival time of demand of product i : A_{it}.

Objective Function

The objective function is a combination of minimizing inventory while retaining buffer stock and maximizing melter utilization.

Constraints

1) Demand constraints: constraints to meet demands using the available inventory and production.

2) Arrival, production and completion times constraints: completion time ideally need to be within the due dates.

3) Melter Capacity: the capacity of each melter is fixed.

4) Unit Production capacity: each product has constant pull rate depending on it's diameter.

5) Collection Capacity: conveyor collection capacity is fixed.

6) Buffer and maximum allowable inventory: for each product the buffer or the safety stock and the maximum allowable inventory is fixed.

Outputs

Relevant outputs of the program in terms of the decision variables are:

x_{ijt}, E_{ijt}, S_{ijt}, I_{it}, Sp_{it}, and Ux_{it}.

3.2.1 Optimization Model:

The scheduling time span is 28 days; each time unit represents 4 days.

Demand constraints: The sum of products i manufactured on all allowable positions and the inventory on hand must satisfy the demand and establish inventory for the following period.

$\Sigma_j x_{ijt} + I_{it-1} = D_{it} + I_{it}$ for all i and t ... 3.1

Arrival, production and completion times constraints: These constraints express the following:

1) the maximum end time of production should preferably be less than the due date,

2) the start times have to be less than the end times,

3) since production planning is on a 4-week basis (or seven time units) manufacturing time is restricted to seven time units unless a product has a due date beyond this time frame.

$\text{Max}(E_{ijt}) + Sn_{it} - Sp_{it} = d_{it}$ for all i, t. ...3.2

Let $\text{Max}(E_{ijt}) = E_{it}$ and

$\text{Min}(S_{ijt}) = S_{it}$ then

$E_{it} \geq E_{ijt}$ for all i, j, t ...3.3

$S_{it} \leq S_{ijt}$ for all i, j, t. 3.4

$S_{it} \leq E_{it}$ for all i, j, t ...3.5

Melter Capacity: The melter for lines 1 and 2 have a fixed capacity of 21 and 15 tons per day, respectively. However, since the process is in discrete time, the continuous state space problem is modeled by considering the line capacity for the entire time horizon (7 time units of 4 days each) in order to avoid non-linearity of constraints.

$\Sigma\Sigma\Sigma x_{ijt} + P1_t \leq 28$ * Line 1 Melter capacity ...3.6

$\Sigma\Sigma\Sigma x_{ijt} + P2_t \leq 28$ * Line 2 Melter capacity ...3.7

(P1 and P2 are patty produced on lines 1 and 2)

Unit Production capacity: Each product has a deterministic pull rate and hence the amount of product manufactured is a multiple of the time for which it was manufactured and its pull rate.

$x_{ijt} \leq (E_{ijt} - S_{ijt}) *$ Pull rate of i for i,j,t ...3.8

Collection Capacity: The capacity of the conveyor collection is fixed.

For Line 1,

$\Sigma\Sigma\Sigma\,(x_{ijt}) \leq$ Conveyor capacity * t ...3.9

for i,j,t of products that can be made on those positions j with the conveyor collection, for the entire time span.

For Line 2,

$\Sigma\Sigma\Sigma\,(x_{ijt}) \leq$ Conveyor capacity * t ...3.10

For i,j,t of products that can be made on those positions j with the conveyor collection, for the entire time span.

Buffer stock and Maximum allowable Inventory: The forecast of the minimum inventory for each product and the maximum allowable inventory is used for these constraints. Due to the restrictions imposed by the implementation software, the buffer stock is maintained and the inventory is kept below the maximum level for the last time unit, so the depleted inventory is rebuilt.

$I_{i7} + Ux_{i7} =$ (Maximum inventory)$_i$ for all i. ...3.11

$I_{i7} - On_{i7} =$ (Minimum inventory)$_i$ for all i. ...3.12

Non-negativity: All variables considered for the model are greater than or equal to 0.

Objective function : Melter utilization and deviation from the maximum inventory is maximized. Maximizing deviation from maximum inventory minimizes inventory over safety stock.

$$\text{Max } \Sigma\Sigma\Sigma\,(x_{ijt}) + \Sigma(Ux_{i7}) \qquad \text{for all i, j, and t.} \qquad ...3.13$$

The LP formulation as an input file for LINDO is given in Appendix 1.

3.3 SUB-MODEL II: SIMULATION MODEL

Since scheduling can be considered as a discrete-event, continuous-time decision making process, discrete event simulation can be used for scheduling decisions [10]. This simulation can be characterized as a 'What if' tool, capable of giving detailed answers to the questions of the form "what will be the effects on performance, if this change were made?". The production plans can be viewed as hierarchy of three levels: i) setting production levels using resource requirements, ii) production schedule to accommodate overlapping of production on machines, and iii) integrating other constraints and requirements of the system, and evaluating the system.

3.3.1 Algorithm Description:

Sub-Model II simulates the current system and improves the scheduling by stabilizing the throughput, from the melters to the fiberizers. The primary objective is to reduce the change in the throughput, while meeting the due dates. The heuristic decision rules: 'minimum rate of change in throughput' and 'earliest due date' are be used to optimize the outputs of sub-model I. The deviation from the absolute minimum change in the throughput is also computed. The output from the linear programming model, for the positions that have job overlap, are arranged in a sequence based on the earliest due date. This sequence is fed into the simulation model. The model considers all the permutations of the jobs sequence where the rate of change in throughput is minimized and the due dates are met. The sequences which allows increase in tardy jobs if there already were any, are not considered. Switch of products at a particular position j from one chemistry to another chemistry, is also checked in this model.

Generation of all possible arrangements of N numbers is as follows: A recursive function, F(N) is defined, where F(2) generates the two arrangements of given two numbers. When F(m) is called with a input vector of order m, one by one, all the m numbers are assigned the top position. This is done (m-1)! times. Having fixed a position for a number, F(m) then calls F(m-1) (recursively) for the remaining (m-1) numbers. These recursive calls terminate for m = 2 as F(2) is known. Thus from an input vector of order N, N! different vectors are generated. Each unique vector is checked for meeting due dates, and the change in throughput is computed. The vectors satisfying the due dates are called valid vectors and the throughput change is computed. The vector with the minimum throughput change ignoring the due date is also generated but the valid vector with minimum change in throughput is the recommended sequence.

3.3.2 Input/Output of the Simulation program:

The input file contains information on the jobs to be scheduled on the manufacturing lines. This information includes the product diameter, the production time, the finish time and the due dates, which are required for the program. All permutations of job arrangements for the order of jobs is generated. These different vectors combinations of order of jobs are input into a sub-routine that checks whether that arrangement allows the due dates to be met. However, only the valid vectors, that is the vector combinations that pass the due-date check are stored. The change in throughput or (|throughput(i+1)-throughput(i)|) is computed for all the valid vector combinations as change in diameter of two consecutive products and the minimum throughput-change vector of job arrangements is recommended. The simulation program is given in Appendix 2 and the flowchart of the simulated system is shown in Figure 4.

3.4 LP/ SIMULATION MODEL

The LP/Simulation scheduling system generates the final result in the following order: An optimal production schedule ignoring machine utilization and throughput change is obtained from the LP. EDD and uniform distribution heuristic further improve the solution by minimizing idling, however this solution ignores throughput change. The simulation model generates a schedule with minimum throughput ignoring the due dates. Finally a sequence with near optimal production, maximum machine utilization, no tardy jobs, and minimum throughput change is obtained. This final solution can be compared to the other intermediate options and the cost of adding various constraints can be recognized.

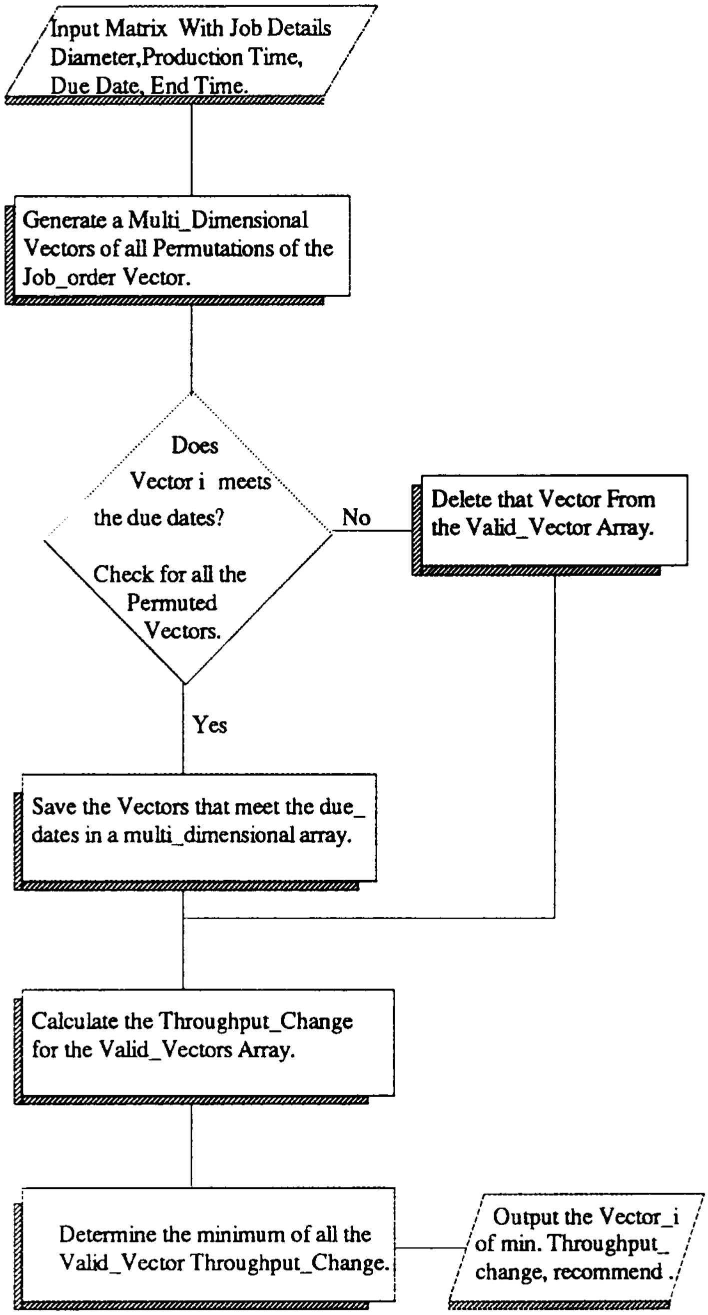

Figure 4. Flowchart for the simulation model

CHAPTER 4

RESULTS

In this chapter the use of the methodology is illustrated. Also a comparison of the system currently in use with the LP/Simulation modeling system is given in order to evaluate the proposed methodology.

4.1 INPUT REQUIREMENTS FOR SUB-MODEL 1:

The solution methodology for the optimization (LP) procedure is illustrated here using a representative set of products and operating conditions (Tables 4.1 and 4.2). The products to be manufactured on the line 1 use the raw material glass of "M" type chemistry while those on the line 2 use the glass of "B" type chemistry. This results in a simplification of the model, since no setup time is required for switching from one chemistry product to another. However, the setup time can be easily incorporated in the simulation model by adding a subroutine that adds the setup time to the start time of a job which follows a switch in glass chemistry. The following information is required for each product to be manufactured; i) product diameter, ii) Safety stock for that product, iii) positions it can be manufactured, iv) a program defined product number, v) maximum allowable inventory and vi) the pull rate. Demand for four weeks is considered, and scheduling is subsequently done for these 28 days or the system defined 7 time units, where each time unit is equal to 4 days.

Input Table.

The input to the linear program for the month of January is shown in Tables 4.1 to 4.3.

Tables 4.1 and 4.2 specify order arrival dates and demand due dates in terms of program defined time units, pull rate is given per time unit in thousands of pounds , and the quantity required and the initial inventory are also in thousands of pounds.

TABLE 4.1

Products to be manufactured on Line 1.

Diameter	Product Number	Safety Stock	Maximum Inventory	Positions	Pull Rate
0.8	22	20.2	80.6	1	3.84
3.0	39	900	4000	2,3,4,5,6,7	24
2.6	36	50	2000	2,3,4,5,6,7	13.4
2.6	37	5	25	2,3,4,5,6,7	12
3.7	41	2.5	50	2,3,4,5,6,7	15.84

TABLE 4.2

Products to be manufactured on Line 2.

Diameter	Product Number	Safety Stock	Maximum Inventory	Positions	Pull Rate
0.5	5	20	160	10,11,12	2.112
0.6	14	3	300	10,11,12	3.072
5.2	46	100	1100	13,14,15	14.4
0.5	7	30	30	10,11,12	3.84
Marble	P2	-	-	9	40.8

Table 4.3

Input Table for Product Demand

Product Number	Order Number	Order Arrival Date	Demand Due Date	Quantity Required	Initial Inventory
5	1	4	4.5	1	9.15
5	2	5	7	0.14	9.15
14	3	1	2	5.6	3
14	4	4	5	8	3
22	5	2	2.5	0.5	0.7
22	6	4	4.5	0.2	0.7
22	7	6	12.5	12	0.7
36	8	3	6	24.031	279.031
36	9	7	7.5	72.155	279.031
39	10	1	2	120.56	938.127
39	11	2	4	97.13	938.127
46	12	1	2	22.4	58
46	13	2	3	25	58

4.2 RESULTS OF SUB-MODEL 1:

The results obtained from the LP and a brief discussion of the analysis follows:

4.2.1 Production on Line 1

Line 1 produces product 22 to meet the demands that arrive at times 2, 4 and 6 and is required at times 2.5, 4.5 and 12.5, respectively. The demand that arrived at time 6 is required at time 12.5 which is beyond the time span of 7 time units considered for the model; however this demand will also be scheduled. Product 22 requires a drum collection mode and can be produced on the position 1. The allowable positions for each product, buffer inventory or the safety stock, the maximum allowable inventory of the product as well as its pull rate are known a-priori, based on the information available from the fiber glass corporation, as summarized in the Table 4.1.

The system utilizes position 1 on line 1 and manufactures 32.2 units of product 22 in 8.39 time units. Production time is determined by taking the difference between the start and the end times. The ending inventory at time 8.39 (or the 3rd of February) is 20.2 units which is the minimum required level. However, the inventory at the end of time 7 is 14.88. The production takes place in order to compensate for the depleted inventory. The variation in inventory over time is shown in Figure 5.

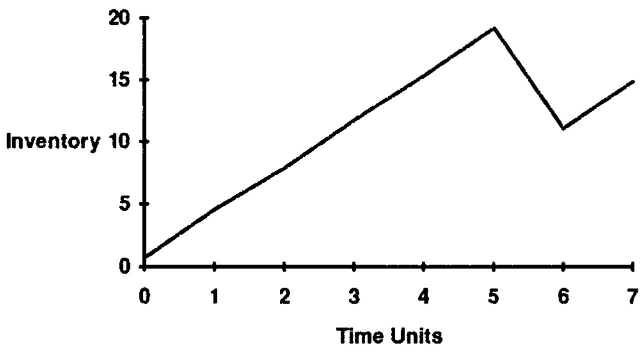

Figure 5. Inventory for product 22.

The production for product 22 is shown in Table 4.4. Position 1 manufactures 32.2 units so that the inventory can come close to the safety stock. The demands of 0.5 and 0.2 units that arrived at time units 2 and 4 were met by on-hand inventory (Table 4.4).

Table 4.4. Production time for product 22.

Time	Order Quantity	Amount on Posn_1	Start Time	End Time	Inv. on Hand	Due Date
1		0	-	-	4.54	
2	0.5	0	-	-	7.88	2.5
3		0	-	-	11.72	
4	0.2	0	-	-	15.36	4.5
5		0	-	-	19.2	
6	12	32.2	1	8	14.88	12.5
7		0	-	-	14.88	

Products 36 and 39

Products 39 and 36 are scheduled to be manufactured on positions 2, 3, 4, 5, 6, and 7 of line 1, since they need a conveyor collection mode. Since these products require the same positions, overlap of production has to be avoided. The heuristic decision rule of giving priority to 'earliest due date' is used to deal with this conflict.

Production of 39 is shown in Table 4.5a as per the solution from the LP. However, the objective of the LP is to maximize production and it does not consider the positions used. In order to maximize utilization of all positions the production is uniformly divided between the positions that can manufacture that product and are connected to a common collection mode (Table 4.5b). In order to minimize idle positions, heuristic rule of uniform distribution of production is used. If the LP solution suggests 'X' amount of product i to be made on position j, then X is uniformly divided among all the positions that can manufacture product i and are connected to a common collection system.

The orders of 120.56 and 97.13 units (refer to Table 4.3) for product 39 arrive at time 1 and 2, the due dates for these demands are at times 2 and 4, respectively. The production for product 39 is scheduled on position 3 to manufacture 36.5 units in 1.5 days, which makes this job tardy by 0.522 time units. However, when the job is distributed on all available six positions, then the production time would be reduced to time 0.2536 and the job would be completed on time. The solution from LP tries to meet the buffer inventory and maximize the production, on positions 2, 3, 4, 5, 6 and 7. In order to minimize idle times of all positions, this production of total units to be manufactured is scheduled evenly on positions 2 through 7. The production time is recomputed based on the number of positions in use and pull rate of the product. The available inventory at the end of the 7th time unit is 900 units.

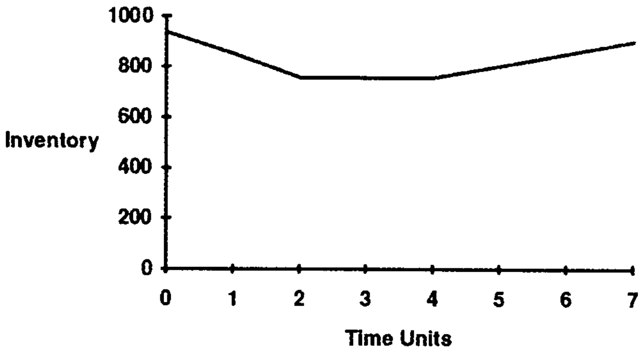

Figure 6. Inventory for product 39.

The production levels determined by the LP is shown in Table 4.5a, and the modified uniform production of 39 is shown in Table 4.5b.

Table 4.5a. Production time for product 39.

		Amount						Start Time						End Time							
Time	Order Quantity	2	3	4 Posn	5	6	7	2	3	4	5	6	7	2	3	4	5	6	7	Inv.	Due Date.
1	120.56	0	36.52	0	0	0	0	-	1	-	-	-	-	-	2.5	-	-	-	-	853	2
2	97.13	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	756	4
3		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	756	
4		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	756	
5		48	0	0	0	0	0	1	-	-	-	-	-	5	-	-	-	-	-	804	
6		48	0	0	0	0	0	1	-	-	-	-	-	5	-	-	-	-	-	852	
7		9.6	0	9.6	9.6	9.6	9.6	1	-	1	1	1	1	1.4	-	1.4	1.4	1.4	1.4	900	

Table 4.5b. Uniform production time for product 39.

		Amount						Start Time						End Time							
Time	Order Quantity	2	3	4 Posn	5	6	7	2	3	4	5	6	7	2	3	4	5	6	7	Due Date	Inv.
1	120.56	6.09	6.09	6.09	6.09	6.09	6.09	1	1	1	1	1	1	1.25	1.25	1.25	1.25	1.25	1.25	2	853
2	97.13	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	4	756
3		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-		756
4		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-		756
5		8	8	8	8	8	8	1	1	1	1	1	1	1.33	1.33	1.33	1.33	1.33	1.33		804
6		8	8	8	8	8	8	1	1	1	1	1	1	1.33	1.33	1.33	1.33	1.33	1.33		852
7		8	8	8	8	8	8	1	1	1	1	1	1	1.33	1.33	1.33	1.33	1.33	1.33		900

The orders of 24.031 and 72.155 for product 36 arrived at time 3 and 7, and the due dates for these demands were times 6 and 7.5, respectively (Table 4.3). Production of 48 units was scheduled by the LP, on position 3 for a time period of 3.6 units. Again to minimize idling, production is spread uniformly on all available positions, that is, 8 units are to be manufactured are distributed on the six positions for 0.6 time units. The available inventory at the end if the 7th time unit is 230.85 units.

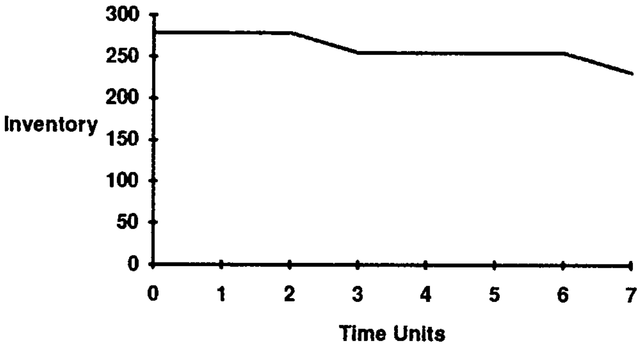

Figure 7. Inventory for product 36.

The production of 36 by LP is shown in Table 4.6a and uniform production in Table 4.6b.

Table 4.6a. Production time for product 36.

		Amount Produced						Start Time						End time							
Time	Order Quantity	2	3 Posns	4	5	6	7	2	3	4	5	6	7	2	3	4	5	6	7	Inv	Due Date
1		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	279	
2		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	279	
3	24.031	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	6
4		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	
5		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	
6		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	
7	72.155	0	48	0	0	0	0	-	1	-	-	-	-	-	4.57	-	-	-	-	231	7.5

Table 4.6b. Uniform production time for product 36.

		Amount Produced						Start Time						End time							
Time	Order Quantity	2	3	4 Posn	5	6	7	2	3	4	5	6	7	2	3	4	5	6	7	Inv.	Due Date
1		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	279	
2		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	279	
3	24.031	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	6
4		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	
5		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	
6		0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	255	
7	72.155	6	6	6	6	6	6	1	1	1	1	1	1	1.6	1.6	1.6	1.6	1.6	1.6	231	7.5

In order to generate an initial sequence of jobs, as an input to the simulation model, due dates of jobs to be scheduled on common positions are considered. Since the due dates for product 39 are less than the due dates for product 36, product 39 is scheduled first on the line 1 and then product 36. The total production time required for products 36 and 39, is only 1.84 time units, out of the available 7 time units. To utilize the available time on line 1, products 37 and 41 are scheduled for 3 and 2.1593 time units. These two products are selected because their initial inventory is below the safety stock, even though no demand orders arrived for them. The production for line 1, positions 2 through 7, is shown in Table 4.7 and graphically in Figure 8. The total production time required for each of the products are considered. A due date of large implies that the job is not critical, it is required for inventory of the next time span.

Table 4.7. Production on Line 1 conveyor positions.

Product No.	Start Time	End Time	Prodn. Time	Due Date
39	0	1.2436	1.2436	2
36	1.2437	1.8406	.597	7
37	1.8407	4.8407	3	Large
41	4.8408	7	2.1592	Large

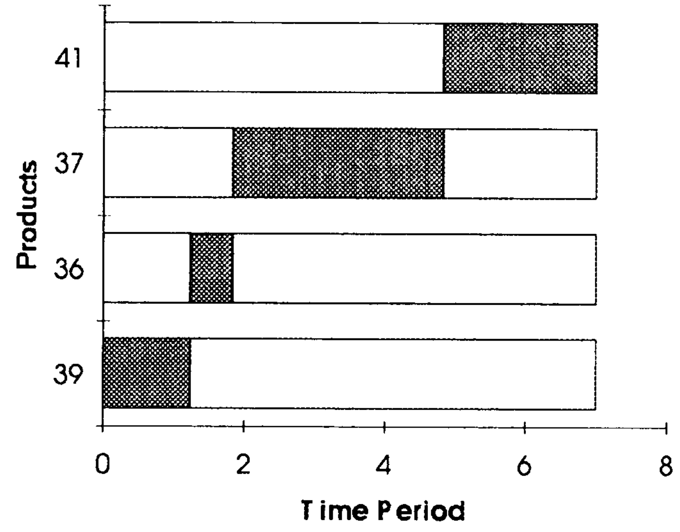

Figure 8. Production on positions 2,3,4,5,6,7 of Line 1.

4.2.2 Production on Line 2

Line 2 produces products 5, 14, and 46. Products 5 and 14 are produced on the positions 10, 11 and 12, since both are fine fibers that need the drum collection mode. Here also scheduling of jobs is modified so that an overlap of production does not occur, since product 5 and 14 need the same positions. Product 46 is a coarse fiber that needs the conveyor collection mode and is therefore manufactured on positions 13, 14 and 15. Position 9 of line 2 is a patty (or by-product P2) position for the time period considered, the demand of the patty is met from the available inventory, and no production is necessary. However, if the available inventory were to fall below a buffer quantity, production can be scheduled since melter capacity of line 2 is not fully utilized.

Product 46

For this product, demands of 22.4 and 25 units arrives at times 1 and 2, with the due dates of 2 and 3, respectively. These demands are met using the available inventory of 58 units. However, 86.4 units are manufactured on positions 13, 14, and 15 for 6 time units. The available inventory at the end of the 7th time unit is 269.8 units.

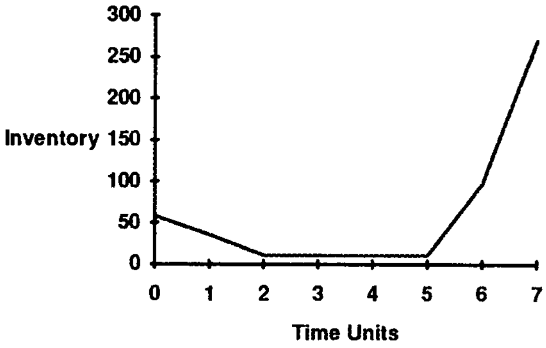

Figure 9. Inventory for product 46.

The production of 46 is shown in Table 4.8a and uniform production in Table 4.8b.

Table 4.8a. Production time for product 46.

		Amount			Start Time			End Time				
Time	Order Quantity	10	11 Posns.	12	10	11	12	10	11	12	Inv	Due Date
1	22.4	0	0	0	-	-	-	-	-	-	35.6	2
2	25	0	0	0	-	-	-	-	-	-	10.6	3
3		0	0	0	-	-	-	-	-	-	10.6	
4		0	0	0	-	-	-	-	-	-	10.6	
5		0	0	0	-	-	-	-	-	-	10.6	
6		86.4	0	0	1	-	-	7	-	-	97	
7		0	86.4	86.4	-	1	1	-	7	7	269.8	

Table 4.8b. Uniform production time for product 46.

Time	Order Quantity	Amount 10	11 Posns.	12	Start Time 10	11	12	End Time 10	11	12	Inv.	Due Date
1	22.4	0	0	0	-	-	-	-	-	-	35.6	2
2	25	0	0	0	-	-	-	-	-	-	10.6	3
3		0	0	0	-	-	-	-	-	-	10.6	
4		0	0	0	-	-	-	-	-	-	10.6	
5		0	0	0	-	-	-	-	-	-	10.6	
6		28.8	28.8	28.8	1	1	1	3	3	3	97	
7		57.6	57.6	57.6	1	1	1	5	5	5	269.8	

Product 14

For product 14, orders of 5.6 and 8 units, arrive at times 1 and 4, and the due dates are 2 and 5, respectively (Table 4.3). In order to meet the demand and the buffer inventory, production of corresponding to the sum of the amounts under each position in Table 4.9a is required. The due dates for the demands are met and the inventory for the product at the end of the time unit 7 is 3.072 units. However, product 5 also is also scheduled on these positions, hence the EDD heuristic is used to determine the sequence of job orders. The state of inventory of product 14 is shown in Figure 10.

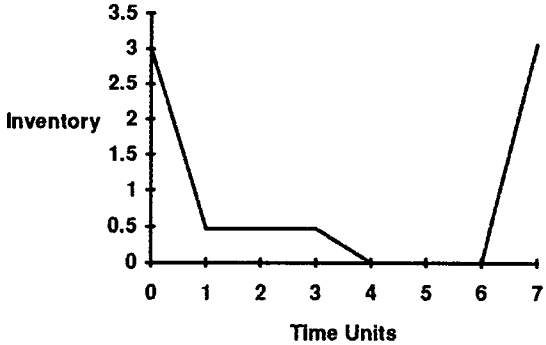

Figure 10. Inventory for product 14.

The production of 14 by LP is shown in Table 4.9a and uniform production in Table 4.9b.

Table 4.9a. Production time for product 14.

Time	Order Quantity	Amount 10	Amount 11 (Posns)	Amount 12	Start 10	Start 11	Start 12	End 10	End 11	End 12	Inv	Due Date
1	5.6	3.072	0	0	1	-	-	2	-	-	.472	2
2		0	0	0	-	-	-	-	-	-	.472	
3		0	0	0	-	-	-	-	-	-	.472	
4		2.51	2.51	2.51	1	1	1	1.82	1.82	1.82	0	
5	8	0	0	0	-	-	-	-	-	-	0	5
6		0	0	0	-	-	-	-	-	-	0	
7		1.024	1.024	1.024	1	1	1	2	2	2	3.072	

Table 4.9b. Uniform production time for product 14.

Time	Order Quantity	Amount 10	Amount 11 (Posns)	Amount 12	Start 10	Start 11	Start 12	End 10	End 11	End 12	Inv	Due Date
1	5.6	1.024	1.024	1.024	1	1	1	1.33	1.33	1.33	.472	2
2		0	0	0	-	-	-	-	-	-	.472	
3		0	0	0	-	-	-	-	-	-	.472	
4		2.51	2.51	2.51	1	1	1	1.82	1.82	1.82	0	
5	8	0	0	0	-	-	-	-	-	-	0	5
6		0	0	0	-	-	-	-	-	-	0	
7		1.024	1.024	1.024	1	1	1	2	2	2	3.072	

Product 5

For product 5, the orders of 1 and 0.14 units arrive at times 4 and 5, with due dates of 4.5 and 7 time units, respectively. Production is scheduled on positions 10, and 11 to manufacture 12.672 units each. Uniform scheduling on positions 10, 11 and 12 results in lower production time of 2 time units. The demands are met on time and an inventory of 33.35 units is available by the end of time unit 7 .

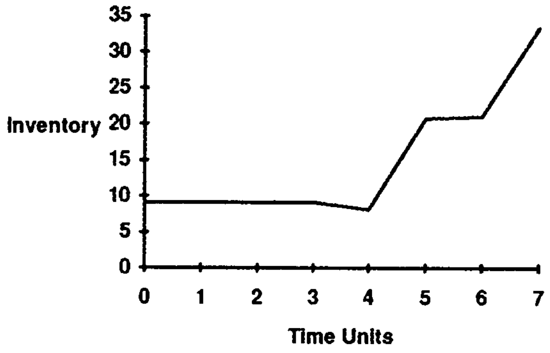

Figure 11. Inventory for product 5.

The production of 5 is shown in Table 4.10a and the uniform production in Table 4.10b.

Table 4.10a. Production time for product 5.

Time	Order Quantity	Amount 10	Amount 11 (Posns)	Amount 12	Start Time 10	Start Time 11	Start Time 12	End Time 10	End Time 11	End Time 12	Inv.	Due Date
1		0	0	0	-	-	-	-	-	-	9.15	
2		0	0	0	-	-	-	-	-	-	9.15	
3		0	0	0	-	-	-	-	-	-	9.15	
4	1	0	0	0	-	-	-	-	-	-	8.15	4.5
5	0.14	12.67	0	0	1	-	-	7	-	-	20.68	7
6		0	0	0	-	-	-	-	-	-	20.68	
7		0	12.672	0	-	1	-	-	7	-	33.35	

Table 4.10b Uniform production time for product 5.

Time	Order Quantity	Amount 10	Amount 11 (Posns)	Amount 12	Start Time 10	Start Time 11	Start Time 12	End Time 10	End Time 11	End Time 12	Inv.	Due Date
1		0	0	0	-	-	-	-	-	-	9.15	
2		0	0	0	-	-	-	-	-	-	9.15	
3		0	0	0	-	-	-	-	-	-	9.15	
4	1	0	0	0	-	-	-	-	-	-	8.15	4.5
5	0.14	4.22	4.22	4.22	1	1	1	3	3	3	20.68	7
6		0	0	0	-	-	-	-	-	-	20.68	
7		4.22	4.22	4.22	1	1	1	3	3	3	33.35	

The final production on line 2, positions 10,11 and 12, is based on the earliest due date rule, (refer to Table 3) and hence order 3 with due date 2 of product 14 is given first priority for production over order 1 of product 5 with due date of 4.5. However, order 1 of product 5 is given higher priority over order 4 of product 14, with due date 5. Also to utilize the production capacity available during 7 time units, product 7 is also scheduled, to increase its safety stock. The production for the positions 10,11, and 12, is shown in Table 4.11 and Figure 12.

Table 4.11. Production on Line 2 drum positions.

Product No.	Start Time	End Time	Prodn. Time	Due Date
14	0	1.15	1.15	2
5	1.16	3.16	2	4.5
14	3.17	4.17	1	7
5	4.18	6.18	2	7
7	6.19	7	0.81	Large

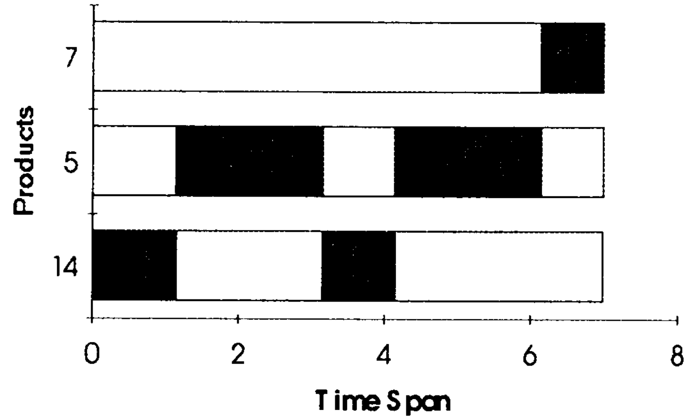

Figure 12. Production time on positions 10,11,12 of Line 2.

<u>4.3 RESULTS FROM SUB-MODEL 2</u> :

The sequenced result of sub-model 1 using the heuristics of EDD and Uniform distribution (Tables 4.7 and 4.11) are reiterated in Tables 4.12 and 4.14 along with some additional information (production times and product diameter). This is the input to sub-model 2 . Positions 2 through 7 on line 1 have a sequence of jobs in the queue and also positions 10,11, and 12 on line 2 have a queue of jobs. Hence, only these position are input to sub-model 2 because these go towards minimizing the change in throughput; single jobs on other positions do not effect the throughput. The input/output file is shown in Appendix 4a for line 1 and Appendix 4b for line 2. The output of sub-model 2 is shown in Tables 4.13 and 4.15. The simulation model takes the input matrix of information regarding job orders, their production time (calculated as the difference of end time and start time) and due dates, and product diameters. If the queue has 5 jobs in a queue then a total permutation of factorial 5 (120) sequences of job orders can be generated. The production time is adjusted accordingly if the order is shifted. Due to the shifting of production times, end times of production also changes. This may lead to a sequence of jobs that have orders which are tardy. Therefore, the program stores only the valid sequences or the sequence of jobs in which the due dates are met. A function of throughput quantity, Throughput-change = | Diameter(i+1) - Diameter(i) |, is calculated for all the valid vectors. Throughput-change is calculated as change in diameter since change in diameter is directly proportional to the pull rate and therefore to the throughput. The vector that has the minimum value for Throughput-change is the best possible sequence of job orders with no tardy jobs. However, if the jobs had no due dates, then the sorted sequence (either ascending or descending) would give the best sequence. In the tables below large due-date implies that this production is carried for building inventory for the next planning horizon.

Table 4.12: Input to simulation model for Line 1.

Product	Diameter	Start time	End Time	Prodn.Time	Due_Date
39	3	0	1.2436	1.2436	2
36	2.6	1.2436	1.8406	0.597	7
37	2.6	1.8407	4.8407	3	Large
41	3.7	4.8408	7	2.1592	Large

Table 4.13: Output of simulation model for Line 1.

Product	Diameter	Start time	End Time	Prodn.Time	Due_Date
39	3	0	1.2436	1.2436	2
36	2.6	1.2437	1.8406	0.597	7
37	2.6	1.8407	4.8407	3	Large
41	3.7	4.8402	7	2.1592	Large

As an output of the sub-model 2 the valid sequence with minimum Throughput-change remains the same as the input sequence with value of Throughput-change equal to 1.50. However, the minimum value ignoring the due dates was determined to be is 1.10 (input/output in appendix 4A). The final production sequence, with no tardy jobs and minimum throughput change is shown in Fig. 13.

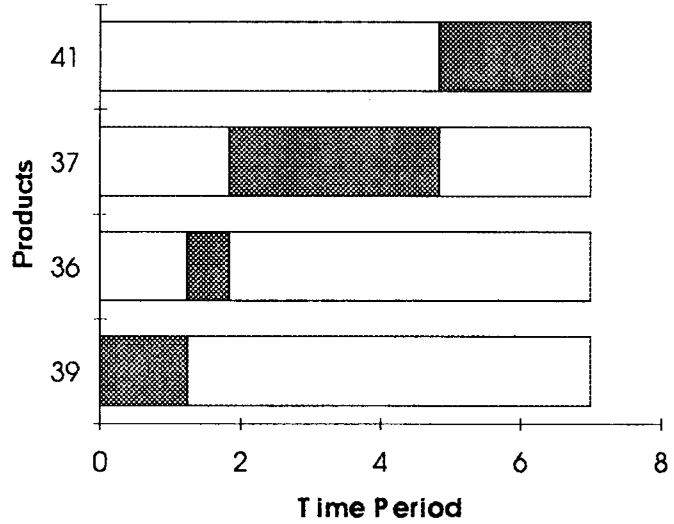

Figure 13. Final production on positions 2,3,4,5,6,7 of Line 1.

Table 4.14: Input to simulation model for Line 2.

Product	Diameter	Start Time	End Time	Prodn.Time	Due_Date
14	0.6	0	1.15	1.15	2
5	0.5	1.16	3.16	2	4.5
14	0.6	3.17	4.17	1	7
5	0.5	4.18	6.18	2	7
7	0.5	6.19	7	0.81	Large

Table 4.15: Output of simulation model for Line 2.

Product	Diameter	Start time	End Time	Prodn.Time	Due_Date
14	0.6	0	1.15	1.15	2
14	0.6	1.16	2.16	1	7
5	0.5	2.17	4.17	2	4.5
5	0.5	4.18	6.18	2	7
7	0.5	6.19	7	0.81	Large

For line 2 the sequence of the jobs varies in the output with the total absolute rate of change of throughput as 0.1 also the minimum total with no due date constraint is 0.1.

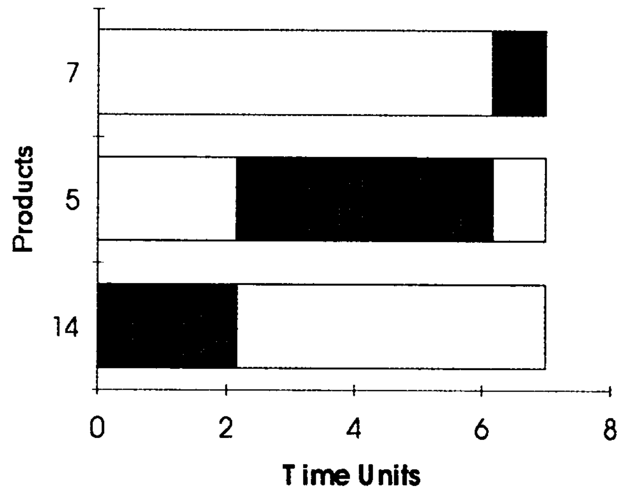

Figure 14. Final production on positions 10,11,12 of Line 2.

4.4 Method Evaluation:

The model is evaluated by comparing the output from the optimization and simulation models to the results of current scheduling method using spreadsheet. The outputs are compared using inventory levels, average throughput change and machine utilization. Tables 4.16 and 4.17 show the results from the spreadsheet model; the results from the proposed model are shown in Tables 4.18 and 4.19.

Table 4.16: Actual scheduling used for Line 1.

Positions on Line 1

Time Unit	1	2	3	4	5	6	7
1	22	Down	Down	39	39	39	39
2	22	Down	Down	39	39	39	39
3	22	Down	Down	39	39	39	39
4.5	22	Down	Down	39	39	39	39
5	Down	Down	Down	39	39	39	39
5.5	Down	Down	Down	39	39	39	39
6	Down	36	36	36	36	36	36
7	Down	36	36	36	36	36	36

Table 4.17: Actual scheduling used for Line 2.

Positions on Line 2

Time Unit	9	10	11	12	13	14	15
1	P2	5	5	5	Down	46	46
2	P2	5	5	5	Down	46	46
3	P2	5	5	5	Down	46	46
3.75	P2	5	5	5	Down	46	46
3.76	P2	14	14	14	14	46	46
4	P2	14	14	14	14	46	46
5	P2	14	14	14	14	46	46
6	P2	14	14	14	14	46	46
7	P2	14	14	14	14	46	46

Table 4.18: Scheduling done using LP and simulation models on Line 1.

Positions on Line 1

Time Unit	1	2	3	4	5	6	7
1	22	39	39	39	39	39	39
2	22	39	39	39	39	39	39
3	22	36	36	36	36	36	36
4	22	37	37	37	37	37	37
5	22	37	37	37	37	37	37
6	22	37	37	37	37	37	37
7	22	41	41	41	41	41	41

Table 4.19: Scheduling done using the LP and simulation models on Line 2.

	Positions on Line 2						
Time Unit	9	10	11	12	13	14	15
1	P2	14	14	14	46	46	46
2	P2	14	14	14	46	46	46
3	P2	5	5	5	46	46	46
4	P2	5	5	5	46	46	46
5	P2	5	5	5	46	46	46
6	P2	5	5	5	46	46	46
7	P2	7	7	7	46	46	46

Based on the above information the ending inventory for each product are determined for both the systems. Table 4.20 shows the ending, the maximum and the minimum inventory for each of the products. Also shown is the deviation from the minimum inventory level for both methodologies. Smaller deviation is preferred since storage and management of excess inventory is undesirable and costly.

Table 4.20: Inventory status for the two scheduling systems.

Product	LP/SIM Model	Current Model	Max. Inv.	Min. Inv.	Devn. from Min. LP/SIM	Devn. from Min. Current.
22	14.88	5.28	80.6	20.2	-5.32	-14.92
39	900	1200.437	4000	900	0	300.437
36	231	383.845	2000	50	181	333.845
5	33.35	31.77	160	20	13.35	11.77
46	269.80	212.2	1100	100	169.8	112.2
14	3.072	29.336	300	3	0.072	26.336

For product 22 the scheduling done by the current system ignores the safety stock or the minimum level of inventory required in case of unwarranted demands. Since the scheduling was done for 7 time units, the LP/Simulation model utilizes the entire 7 time units and tries to build the inventory. The average deviation from minimum inventory for the current system is 128.278, while it is only 61.59 for the proposed LP/Simulation system.

Table 4.3 showed the arrival and due dates of different job orders. Product 14 is required at time units 2 and 5, however production for this product by the current system does not start by time unit 4.5 from Table 4.17 and on-hand inventory is not sufficient to meet demand. It is clear from Table 4.17 and Table 4.3 that job number 3 and 4 are tardy using current scheduling. The optimization model results in a valid sequence of on time jobs.

As a result of the LP/Simulation model no tardy jobs are obtained. The melter capacity utilized by the LP/Simulation system is 930.5082 pounds as compared to 900 pounds using the current method. The difference is not significant due to the fact that the more of thicker fibers are manufactured by the current system, instead of making thinner fibers. In terms of efficient usage of the melter capacity the proposed system is better since it achieves lower inventory while maintaining the safety stock, and meets all due dates.

CHAPTER 5

CONCLUSIONS

The purpose of this research was to develop an efficient production scheduling technique for a glass fiber operation. The decision system used is an integrated algorithm involving optimization and heuristics. The optimization model is a product mix linear programming model and the heuristic based model involves simulation of the discrete event system.

Production scheduling was required in order to minimize the deviation from the target inventory, stabilizing the throughput from the melter, and meeting the demand. Due to the complexity of the system and limitation of the software (LINDO for IBM PC restricts the number of constraints to 699 and subsequently the number of variables) used for the optimization module, the problem was dealt in two phases. The first phase determines the amount of products to be scheduled for production on each machine. The important factors in consideration were the incoming inventory of each product, the capacity of the collection system and the melters, and the forecasted arrival of demands.

The second phase takes the solution from the optimization module and further improves the solution set by minimizing the rate of change of throughput from the melters into the manufacturing lines. The solution set from the LP model was ordered based on the EDD (earliest due date) rule, so that any overlap of assigned tasks on a machine can be resolved into a queue of prioritized jobs waiting to be scheduled.

The aim of the two subsystems was to generate a feasible near optimal solution within the range of practical limitations so as to reduce the manufacturing costs involved and save time by efficient allocation of jobs.

The output from the LP model give production levels required to meet demand and restore depleted inventory, so as to maintain the safety stock and still be below the maximum allowable inventory. It is observed that for several products the incoming inventory was less than the safety stock required, so production continued over and above the planning horizon. Again due to the limitation of the LINDO software, the mathematical equations did not rule out the solution set where a machine could process multiple jobs at the same time. This problem was resolved by simply ordering the jobs based on EDD. It was observed that only two production systems had a queue of jobs waiting to be processed, the conveyor production system on Line 1 and the drum collection system on Line 2. Hence the partial solution from the LP was fed into the simulation system, and a solution that met the due dates and minimized the change in throughput was obtained. The minimization in throughput meant stabilizing the melters, which in turn met minimizing the operational costs of the system. The repetitive use of this integrated system over several planning horizons would aid in generating better solutions.

The present study therefore provides:

1) A closer look at the deviation from the near optimal solution, since the LP model suggests the amount of each product to be manufactured under all the manufacturing system constraints. Also the real life situations suggest that a near optimal solution is more desirable than a ideal optimal solution, due to its practicality and feasibility.

2) The simulation model allows to experiment with 'what if?' kind of situations. The analyst is given the freedom of trying out various possibilities of allocating jobs that were not considered due to absence of demands. This makes the system flexible for future use.

5.1 Recommendations for future research

The system developed in this research provides a methodology that focuses on combining the strengths of two solution approaches: optimization and heuristics. There are two areas for future research that follow from this study.

1) Evaluation of the proposed methodology. The system was evaluated for a limited set of products. Additional evaluation covering a larger set of products would be desirable. Furthermore, the usefulness of the model for other fiber glass manufacturing facilities should be explored.

2) Limiting the characteristics of the problem handled in sub-model 1 helped in reducing the model size and improved the computational efficiency of the optimization system. These features and more were then included in the simulation system. It would be interesting to model some of the non-linear features as an embedded expert system. However, this development would require observation and interaction of manufacturing operations and personnel at a production facility.

References

1. Day, James E. and Michael P. Hottenstein, "Review of Sequencing Research", Naval Research Logistics Quarterly, 25, 3, 118-146,1970.

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

3. Johnson L. and Montgomery D., Operations Research in Production Planning: Scheduling and Inventory Control, John Wiley& Sons, New York, NY, 1979.

4. Conway, R. W., "Priority Dispatching and Work-in-Process Inventory in a Job Shop", The Journal of Industrial Engineering, 16, 2, 123-130. 1965.

5. F. V. Tooley , Handbook of Glass Manufacture, Vols 1, and 2. 3rd Edition, Ashlee Publishing Co., New York, NY, 1992.

6. Gnann. A. "Speciality Glass Fibers, Technical Aspects of their use in Nonwovens and Papermaking Applications", 1983, Marketing Technology Service Inc., Kalamazoo, MI.

7. Sircar A., "Glass Compositions and Workability", Glass Industry, 26, 7, 13-14, 1991.

8. Caroselli R.F. "Glass Textile Fibers", Eds. H. F. Marks, S. M. Atlas, and E. Cernia, Man Made Fibers , 425-454, Wiley-Interscience Publishers, Inc., New York 1968.

9. Schaeffel, N. " Glass, Ceramic, and Quartz Fibers for the aper Industry", TAPPI Journal, 58, 5, 56-60, 1975.

10. Rogers P. and M. T. Flanagan. "On-Line Simulation for Real Time Scheduling" , Industrial Engineering, 23, 12, 37-40, 1991.

11. Lowenstein, K. L. The Manufacturing Technology of Continuos Glass Fibers, Elsevier, Science Publishing Co., New York, NY, 1983.

12. Volf M.B., Chemical Approach to Glass, Elsevier Science Publishing Co. New York, NY, 1984.

13. Bryant M. "Glass Fabrics Production - An overview", Proc. National Electronic and Production Conference. Industrial and Scientific Conference Management.

1,16594, 245-247, 1992.

14. Sztrimbely W. and P. J. Weymouth, " Dynamic process plant simulation and scheduling: an expert systems approach", Simulation, 56,3, 175-178, 1991.

15. Bruno G., A. Elia, and P. Laface " A Rule based system to schedule production". Computer, 19, 7, 32-39, 1986.

16. Zhou D. N., V. Cherkassky, T.R. Bladwin, and D. E. Olson, " A Neural Network Approach to Job Shop Scheduling'. IEEE Trans. on Neural Networks, 2, 1, 175-179, 1991.

17. Hong K. S. and J. Y-T Leung, "On-Line Scheduling of Real-Time Tasks", IEEE Trans. on Computers, 41,10, 1326-1331,1992.

18. Russel A. D. , and W. F. Casselton "Extension to Linear Scheduling Optimization" Journal of Construction Engineering and Management, 114, 1, 36-52,1988.

19. Lund J. R. , "Least Cost Scheduling of Solid Waste Recycling", Journal of Environmental Engineering, 116, 1, 182-196, 1990.

20. E. T. Bjarnason, " A Decision Support System for Integrating Fishing and Fish Processing", M.S. Thesis, Industrial and Manufacturing Engineering, Oregon State University, 1992.

21. Schrage W. LINDO: Linear, Interactive Device for Optimization, Scientific Press, Palo Alto, CA, 1982.

22. Borland C, Borland International Inc.

APPENDICES

APPENDIX 1

Formulation of the LP.

Given below is a listing of the input file for LINDO.

MIN - X1 - X2 - ΣUx_{i7}

SUBJECT TO
/* Equations 2 through 50 represents the demand constraints */

```
 2) I221 - X2211 =  .7
 3) I221 + X2212 - I222 =  .5
 4) I222 + X2213 - I223 =  0
 5) I223 + X2214 - I224 =  .2
 6) I224 + X2215 + I225 =  0
 7) I225 + X2216 - I226 =  12
 8) I226 + X2217 - I227 =  0

 9) I391 - X3921 - X3931 - X3941 - X3951 - X3961 - X3971
    =  816.611
10) I391 + X3922 + X3932 + X3942 + X3952 + X3962 + X3972 - I392
    =  97.13
11) I392 + X3923 + X3933 + X3943 + X3953 + X3963 + X3973 - I393
    =  0
12) I393 + X3924 + X3934 + X3944 + X3954 + X3964 + X3974 - I394
    =  0
13) I394 + X3925 + X3935 + X3945 + X3955 + X3965 + X3975 - I395
    =  0
15) I396 + X3927 + X3937 + X3947 + X3957 + X3967 + X3977 - I397
    =  0

16) I361 - X3621 - X3631 - X3641 - X3651 - X3661 - X3671
    =  279.031
17) I361 + X3622 + X3632 + X3642 + X3652 + X3662 + X3672 - I362
    =  0
18) I362 + X3623 + X3633 + X3643 + X3653 + X3663 + X3673 - I363
    =  24.031
19) I363 + X3624 + X3634 + X3644 + X3654 + X3664 + X3674 - I364
    =  0
20) I364 + X3625 + X3635 + X3645 + X3655 + X3665 + X3675 - I365
    =  0
21) I365 + X3626 + X3636 + X3646 + X3656 + X3666 + X3676 - I366
    =  0
22) I366 + X3627 + X3637 + X3647 + X3657 + X3667 + X3677 - I367
    =  72.155

23) IP21 - XP291 =  170.505
24) IP21 + XP292 - IP22 =  0
25) IP22 + XP293 - IP23 =  0
26) IP23 + XP294 - IP24 =  0
```

27) IP24 + XP295 - IP25 = 0
28) IP25 + XP296 - IP26 = 0
29) IP26 + XP297 - IP27 = 5

30) I51 - X5101 - X5111 - X5121 = 9.15
31) I51 + X5102 + X5112 + X5122 - I52 = 0
32) I52 + X5103 + X5113 + X5123 - I53 = 0
33) I53 + X5104 + X5114 + X5124 - I54 = 1
34) I54 + X5105 + X5115 + X5125 - I55 = .14
35) I55 + X5106 + X5116 + X5126 - I56 = 0
36) I56 + X5107 + X5117 + X5127 - I57 = 0

37) X14101 + X14111 + X14121 - I141 = 2.6
38) I141 + X14102 + X14112 + X14122 - I142 = 0
39) I142 + X14103 + X14113 + X14123 - I143 = 0
40) I143 + X14104 + X14114 + X14124 - I144 = 8
41) I144 + X14105 + X14115 + X14125 - I145 = 0
42) I145 + X14106 + X14116 + X14126 - I146 = 0
43) I146 + X14107 + X14117 + X14127 - I147 = 0

44) I461 - X46131 - X46141 - X46151 = 35.6
45) I461 + X46132 + X46142 + X46152 - I462 = 25
46) I462 + X46133 + X46143 + X46153 - I463 = 0
47) I463 + X46134 + X46144 + X46154 - I464 = 0
48) I464 + X46135 + X46145 + X46155 - I465 = 0
49) I465 + X46136 + X46146 + X46156 - I466 = 0
50) I466 + X46137 + X46147 + X46157 - I467 = 0

/* Equations 51 through 132 represent the start and end times within the planning horizon*/

51) S391 >= 1
52) E391 <= 7
53) S392 >= 1
54) E392 <= 7
55) S393 >= 1
56) E393 <= 7
57) S394 >= 1
58) E394 <= 7
59) S395 >= 1
60) E395 <= 7
61) S396 >= 1
62) E396 <= 7
63) S397 >= 1
64) E397 <= 7

65) S221 >= 1
66) E221 <= 7
67) S222 >= 1
68) E222 <= 7
69) S223 >= 1
70) E223 <= 7
71) S224 >= 1
72) E224 <= 7

73) S225 >= 1
74) E225 <= 7
75) S226 >= 0
76) S227 >= 0

77) S361 >= 1
78) E361 <= 7
79) S362 >= 1
80) E362 <= 7
81) S363 >= 1
82) E363 <= 7
83) S364 >= 1
84) E364 <= 7
85) S365 >= 1
86) E365 <= 7
87) S366 >= 1
88) E366 <= 7
89) S367 >= 1
90) E367 <= 7

91) S51 >= 1
92) E51 <= 7
93) S52 >= 1
94) E52 <= 7
95) S53 >= 1
96) E53 <= 7
97) S54 >= 1
98) E54 <= 7
99) S55 >= 1
100) E55 <= 7
101) S56 >= 1
102) E56 <= 7
103) S57 >= 1
104) E57 <= 7

105) S141 >= 1
106) E141 <= 7
107) S142 >= 1
108) E142 <= 7
109) S143 >= 1
110) E143 <= 7
111) S144 >= 1
112) E144 <= 7
113) S145 >= 1
114) E145 <= 7
115) S146 >= 1
116) E146 <= 7
117) S147 >= 1
118) E147 <= 7

119) S461 >= 1
120) E461 <= 7

121) S462 >= 1
122) E462 <= 7
123) S463 >= 1
124) E463 <= 7
125) S464 >= 1
126) E464 <= 7
127) S465 >= 1
128) E465 <= 7
129) S466 >= 1
130) E466 <= 7
131) S467 >= 1
132) E467 <= 7

/* Equations 133 through 440 define the minimum start times and maximum end times */
133) E221 - E2211 >= 0
134) E222 - E2212 >= 0
135) E223 - E2213 >= 0
136) E224 - E2214 >= 0
137) E225 - E2215 >= 0
138) E391 - E3921 >= 0

139) E391 - E3931 >= 0
140) E391 - E3941 >= 0
141) E391 - E3951 >= 0
142) E391 - E3961 >= 0
143) E391 - E3971 >= 0
144) E392 - E3922 >= 0
145) E392 - E3932 >= 0
146) E392 - E3942 >= 0
147) E392 - E3952 >= 0
148) E392 - E3962 >= 0
149) E392 - E3972 >= 0
150) E393 - E3923 >= 0
151) E393 - E3933 >= 0
152) E393 - E3943 >= 0
153) E393 - E3953 >= 0
154) E393 - E3963 >= 0
155) E393 - E3973 >= 0
156) E394 - E3924 >= 0
157) E394 - E3934 >= 0
158) E394 - E3944 >= 0
159) E394 - E3954 >= 0
160) E394 - E3964 >= 0
161) E394 - E3974 >= 0
162) E395 - E3925 >= 0
163) E395 - E3935 >= 0
164) E395 - E3945 >= 0
165) E395 - E3955 >= 0
166) E395 - E3965 >= 0
167) E395 - E3975 >= 0
168) E396 - E3926 >= 0
169) E396 - E3936 >= 0

170) E396 - E3946 >= 0
171) E396 - E3956 >= 0
172) E396 - E3966 >= 0
173) E396 - E3976 >= 0
174) E397 - E3927 >= 0
175) E397 - E3937 >= 0
176) E397 - E3947 >= 0
177) E397 - E3957 >= 0
178) E397 - E3967 >= 0
179) E397 - E3977 >= 0

180) E361 - E3621 >= 0
181) E361 - E3631 >= 0
182) E361 - E3641 >= 0
183) E361 - E3651 >= 0
184) E361 - E3661 >= 0
185) E361 - E3671 >= 0
186) E362 - E3622 >= 0
187) E362 - E3632 >= 0
188) E362 - E3642 >= 0
189) E362 - E3652 >= 0
190) E362 - E3662 >= 0
191) E362 - E3672 >= 0
192) E363 - E3623 >= 0
193) E363 - E3633 >= 0
194) E363 - E3643 >= 0
195) E363 - E3653 >= 0
196) E363 - E3663 >= 0
197) E363 - E3673 >= 0
198) E364 - E3624 >= 0
199) E364 - E3634 >= 0
200) E364 - E3644 >= 0
201) E364 - E3654 >= 0
202) E364 - E3664 >= 0
203) E364 - E3674 >= 0
204) E365 - E3625 >= 0
205) E365 - E3635 >= 0
206) E365 - E3645 >= 0
207) E365 - E3655 >= 0
208) E365 - E3665 >= 0
209) E365 - E3675 >= 0
210) E366 - E3626 >= 0
211) E366 - E3636 >= 0
212) E366 - E3646 >= 0
213) E366 - E3656 >= 0
214) E366 - E3666 >= 0
215) E366 - E3676 >= 0
216) E367 - E3627 >= 0
217) E367 - E3637 >= 0
218) E367 - E3647 >= 0
219) E367 - E3657 >= 0
220) E367 - E3667 >= 0

221) E367 - E3677 >= 0

222) E51 - E5101 >= 0
223) E51 - E5111 >= 0
224) E51 - E5121 >= 0
225) E52 - E5102 >= 0
226) E52 - E5112 >= 0
227) E52 - E5122 >= 0
228) E53 - E5103 >= 0
229) E53 - E5113 >= 0
230) E53 - E5123 >= 0
231) E54 - E5104 >= 0
232) E54 - E5114 >= 0
233) E54 - E5124 >= 0
234) E55 - E5105 >= 0
235) E55 - E5115 >= 0
236) E55 - E5125 >= 0
237) E56 - E5106 >= 0
238) E56 - E5116 >= 0
239) E56 - E5126 >= 0
240) E57 - E5107 >= 0
241) E57 - E5117 >= 0
242) E57 - E5127 >= 0

243) E141 - E14101 >= 0
244) E141 - E14111 >= 0
245) E141 - E14121 >= 0
246) E142 - E14102 >= 0
247) E142 - E14112 >= 0
248) E142 - E14122 >= 0
249) E143 - E14103 >= 0
250) E143 - E14113 >= 0
251) E143 - E14123 >= 0
252) E144 - E14104 >= 0
253) E144 - E14114 >= 0
254) E144 - E14124 >= 0
255) E145 - E14105 >= 0
256) E145 - E14115 >= 0
257) E145 - E14125 >= 0
258) E146 - E14106 >= 0
259) E146 - E14116 >= 0
260) E146 - E14126 >= 0
261) E147 - E14107 >= 0
262) E147 - E14117 >= 0
263) E147 - E14127 >= 0

264) E461 - E46131 >= 0
265) E461 - E46141 >= 0
266) E461 - E46151 >= 0
267) E462 - E46132 >= 0
268) E462 - E46142 >= 0
269) E462 - E46152 >= 0

270) E463 - E46133 >= 0
271) E463 - E46143 >= 0
272) E463 - E46153 >= 0
273) E464 - E46134 >= 0
274) E464 - E46144 >= 0
275) E464 - E46154 >= 0
276) E465 - E46135 >= 0
277) E465 - E46145 >= 0
278) E465 - E46155 >= 0
279) E466 - E46136 >= 0
280) E466 - E46146 >= 0
281) E466 - E46156 >= 0
282) E467 - E46137 >= 0
283) E467 - E46147 >= 0
284) E467 - E46157 >= 0

285) EP21 - EP291 >= 0

286) S221 - S2211 <= 0
287) S222 - S2212 <= 0
288) S223 - S2213 <= 0
289) S224 - S2214 <= 0
290) S225 - S2215 <= 0
291) S226 - S2216 <= 0
292) S227 - S2217 <= 0

293) S391 - S3921 <= 0
294) S391 - S3931 <= 0
295) S391 - S3941 <= 0
296) S391 - S3951 <= 0
297) S391 - S3961 <= 0
298) S391 - S3971 <= 0
299) S392 - S3922 <= 0
300) S392 - S3932 <= 0
301) S392 - S3942 <= 0
302) S392 - S3952 <= 0
303) S392 - S3962 <= 0
304) S392 - S3972 <= 0
305) S393 - S3923 <= 0
306) S393 - S3933 <= 0
307) S393 - S3943 <= 0
308) S393 - S3953 <= 0
309) S393 - S3963 <= 0
310) S393 - S3973 <= 0
311) S394 - S3924 <= 0
312) S394 - S3934 <= 0
313) S394 - S3944 <= 0
314) S394 - S3954 <= 0
315) S394 - S3964 <= 0
316) S394 - S3974 <= 0
317) S395 - S3925 <= 0
318) S395 - S3935 <= 0

319) S395 - S3945 <= 0
320) S395 - S3955 <= 0
321) S395 - S3965 <= 0
322) S395 - S3975 <= 0
323) S396 - S3926 <= 0
324) S396 - S3936 <= 0
325) S396 - S3946 <= 0
326) S396 - S3956 <= 0
327) S396 - S3966 <= 0
328) S396 - S3976 <= 0
329) S397 - S3927 <= 0
330) S397 - S3937 <= 0
331) S397 - S3947 <= 0
332) S397 - S3957 <= 0
333) S397 - S3967 <= 0
334) S397 - S3977 <= 0

335) S361 - S3621 <= 0
336) S361 - S3631 <= 0
337) S361 - S3641 <= 0
338) S361 - S3651 <= 0
339) S361 - S3661 <= 0
340) S361 - S3671 <= 0
341) S362 - S3622 <= 0
342) S362 - S3632 <= 0
343) S362 - S3642 <= 0
344) S362 - S3652 <= 0
345) S362 - S3662 <= 0
346) S362 - S3672 <= 0
347) S363 - S3623 <= 0
348) S363 - S3633 <= 0
349) S363 - S3643 <= 0
350) S363 - S3653 <= 0
351) S363 - S3663 <= 0
352) S363 - S3673 <= 0
353) S364 - S3624 <= 0
354) S364 - S3634 <= 0
355) S364 - S3644 <= 0
356) S364 - S3654 <= 0
357) S364 - S3664 <= 0
358) S364 - S3674 <= 0
359) S365 - S3625 <= 0
360) S365 - S3635 <= 0
361) S365 - S3645 <= 0
362) S365 - S3655 <= 0
363) S365 - S3665 <= 0
364) S365 - S3675 <= 0
365) S366 - S3626 <= 0
366) S366 - S3636 <= 0
367) S366 - S3646 <= 0
368) S366 - S3656 <= 0
369) S366 - S3666 <= 0

370) S366 - S3676 <= 0
371) S367 - S3627 <= 0
372) S367 - S3637 <= 0
373) S367 - S3647 <= 0
374) S367 - S3657 <= 0
375) S367 - S3667 <= 0
376) S367 - S3677 <= 0

377) S51 - S5101 <= 0
378) S51 - S5111 <= 0
379) S51 - S5121 <= 0
380) S52 - S5102 <= 0
381) S52 - S5112 <= 0
382) S52 - S5122 <= 0
383) S53 - S5103 <= 0
384) S53 - S5113 <= 0
385) S53 - S5123 <= 0
386) S54 - S5104 <= 0
387) S54 - S5114 <= 0
388) S54 - S5124 <= 0
389) S55 - S5105 <= 0
390) S55 - S5115 <= 0
391) S55 - S5125 <= 0
392) S56 - S5106 <= 0
393) S56 - S5116 <= 0
394) S56 - S5126 <= 0
395) S57 - S5107 <= 0
396) S57 - S5117 <= 0
397) S57 - S5127 <= 0

398) S141 - S14101 <= 0
399) S141 - S14111 <= 0
400) S141 - S14121 <= 0
401) S142 - S14102 <= 0
402) S142 - S14112 <= 0
403) S142 - S14122 <= 0
404) S143 - S14103 <= 0
405) S143 - S14113 <= 0
406) S143 - S14123 <= 0
407) S144 - S14104 <= 0
408) S144 - S14114 <= 0
409) S144 - S14124 <= 0
410) S145 - S14105 <= 0
411) S145 - S14115 <= 0
412) S145 - S14125 <= 0
413) S146 - S14106 <= 0
414) S146 - S14116 <= 0
415) S146 - S14126 <= 0
416) S147 - S14107 <= 0
417) S147 - S14117 <= 0
418) S147 - S14127 <= 0

419) S461 - S46131 <= 0
420) S461 - S46141 <= 0
421) S461 - S46151 <= 0
422) S462 - S46132 <= 0
423) S462 - S46142 <= 0
424) S462 - S46152 <= 0
425) S463 - S46133 <= 0
426) S463 - S46143 <= 0
427) S463 - S46153 <= 0
428) S464 - S46134 <= 0
429) S464 - S46144 <= 0
430) S464 - S46154 <= 0
431) S465 - S46135 <= 0
432) S465 - S46145 <= 0
433) S465 - S46155 <= 0
434) S466 - S46136 <= 0
435) S466 - S46146 <= 0
436) S466 - S46156 <= 0
437) S467 - S46137 <= 0
438) S467 - S46147 <= 0
439) S467 - S46157 <= 0
440) SP21 - SP291 <= 0

/* Due dates constraint, 441 through 454 */
441) E222 + SN222 - SP222 = 2.5
442) E224 + SN224 - SP224 = 4.5
443) E2216 + SN226 - SP226 = 12.5

444) E391 + SN391 - SP391 = 2
445) E392 + SN392 - SP392 = 4

446) E363 + SN363 - SP363 = 6
447) E367 + SN367 - SP367 = 7.5

448) E54 + SN54 - SP54 = 4.5
449) E55 + SN55 - SP55 = 7

450) E141 + SN141 - SP141 = 2
451) E144 + SN144 - SP144 = 5

452) E461 + SN461 - SP461 = 2
453) E462 + SN462 - SP462 = 3

454) EP21 + SNP21 - SPP21 = 7

/*Individual product pull rate and production constraints, 455 through 622 */
455) X2211 - 3.84 E2211 + 3.84 S2211 <= 0
456) X2212 - 3.84 E2212 + 3.84 S2212 <= 0
457) X2213 - 3.84 E2213 + 3.84 S2213 <= 0
458) X2214 - 3.84 E2214 + 3.84 S2214 <= 0
459) X2215 - 3.84 E2215 + 3.84 S2215 <= 0
460) X2216 + 3.84 S2216 - 3.84 E2216 <= 0

461) X2217 + 3.84 S2217 - 3.84 E2217 <= 0

462) X3921 - 24 E3921 + 24 S3921 <= 0
463) X3922 - 24 E3922 + 24 S3922 <= 0
464) X3923 - 24 E3923 + 24 S3923 <= 0
465) X3924 - 24 E3924 + 24 S3924 <= 0
466) X3925 - 24 E3925 + 24 S3925 <= 0
467) X3926 - 24 E3926 + 24 S3926 <= 0
468) X3927 - 24 E3927 + 24 S3927 <= 0
469) X3931 - 24 E3931 + 24 S3931 <= 0
470) X3932 - 24 E3932 + 24 S3932 <= 0
471) X3933 - 24 E3933 + 24 S3933 <= 0
472) X3934 - 24 E3934 + 24 S3934 <= 0
473) X3935 - 24 E3935 + 24 S3935 <= 0
474) X3936 - 24 E3936 + 24 S3936 <= 0
475) X3937 - 24 E3937 + 24 S3937 <= 0
476) X3941 - 24 E3941 + 24 S3941 <= 0
477) X3942 - 24 E3942 + 24 S3942 <= 0
478) X3943 - 24 E3943 + 24 S3943 <= 0
479) X3944 - 24 E3944 + 24 S3944 <= 0
480) X3945 - 24 E3945 + 24 S3945 <= 0
481) X3946 - 24 E3946 + 24 S3946 <= 0
482) X3947 - 24 E3947 + 24 S3947 <= 0
483) X3951 - 24 E3951 + 24 S3951 <= 0
484) X3952 - 24 E3952 + 24 S3952 <= 0
485) X3953 - 24 E3953 + 24 S3953 <= 0
486) X3954 - 24 E3954 + 24 S3954 <= 0
487) X3955 - 24 E3955 + 24 S3955 <= 0
488) X3956 - 24 E3956 + 24 S3956 <= 0
489) X3957 - 24 E3957 + 24 S3957 <= 0
490) X3961 - 24 E3961 + 24 S3961 <= 0
491) X3962 - 24 E3962 + 24 S3962 <= 0
492) X3963 - 24 E3963 + 24 S3963 <= 0
493) X3964 - 24 E3964 + 24 S3964 <= 0
494) X3965 - 24 E3965 + 24 S3965 <= 0
495) X3966 - 24 E3966 + 24 S3966 <= 0
496) X3967 - 24 E3967 + 24 S3967 <= 0
497) X3971 - 24 E3971 + 24 S3971 <= 0
498) X3972 - 24 E3972 + 24 S3972 <= 0
499) X3973 - 24 E3973 + 24 S3973 <= 0
500) X3974 - 24 E3974 + 24 S3974 <= 0
501) X3975 - 24 E3975 + 24 S3975 <= 0
502) X3976 - 24 E3976 + 24 S3976 <= 0
503) X3977 - 24 E3977 + 24 S3977 <= 0

504) X3621 - 13.44 E3621 + 13.44 S3621 <= 0
505) X3622 - 13.44 E3622 + 13.44 S3622 <= 0
506) X3623 - 13.44 E3623 + 13.44 S3623 <= 0
507) X3624 - 13.44 E3624 + 13.44 S3624 <= 0
508) X3625 - 13.44 E3625 + 13.44 S3625 <= 0
509) X3626 - 13.44 E3626 + 13.44 S3626 <= 0
510) X3627 - 13.44 E3627 + 13.44 S3627 <= 0

511) X3631 - 13.44 E3631 + 13.44 S3631 <= 0
512) X3632 - 13.44 E3632 + 13.44 S3632 <= 0
513) X3633 - 13.44 E3633 + 13.44 S3633 <= 0
514) X3634 - 13.44 E3634 + 13.44 S3634 <= 0
515) X3635 - 13.44 E3635 + 13.44 S3635 <= 0
516) X3636 - 13.44 E3636 + 13.44 S3636 <= 0
517) X3637 - 13.44 E3637 + 13.44 S3637 <= 0
518) X3641 - 13.44 E3641 + 13.44 S3641 <= 0
519) X3642 - 13.44 E3642 + 13.44 S3642 <= 0
520) X3643 - 13.44 E3643 + 13.44 S3643 <= 0
521) X3644 - 13.44 E3644 + 13.44 S3644 <= 0
522) X3645 - 13.44 E3645 + 13.44 S3645 <= 0
523) X3646 - 13.44 E3646 + 13.44 S3646 <= 0
524) X3647 - 13.44 E3647 + 13.44 S3647 <= 0
525) X3651 - 13.44 E3651 + 13.44 S3651 <= 0
526) X3652 - 13.44 E3652 + 13.44 S3652 <= 0
527) X3653 - 13.44 E3653 + 13.44 S3653 <= 0
528) X3654 - 13.44 E3654 + 13.44 S3654 <= 0
529) X3655 - 13.44 E3655 + 13.44 S3655 <= 0
530) X3656 - 13.44 E3656 + 13.44 S3656 <= 0
531) X3657 - 13.44 E3657 + 13.44 S3657 <= 0
532) X3661 - 13.44 E3661 + 13.44 S3661 <= 0
533) X3662 - 13.44 E3662 + 13.44 S3662 <= 0
534) X3663 - 13.44 E3663 + 13.44 S3663 <= 0
535) X3664 - 13.44 E3664 + 13.44 S3664 <= 0
536) X3665 - 13.44 E3665 + 13.44 S3665 <= 0
537) X3666 - 13.44 E3666 + 13.44 S3666 <= 0
538) X3667 - 13.44 E3667 + 13.44 S3667 <= 0
539) X3671 - 13.44 E3671 + 13.44 S3671 <= 0
540) X3672 - 13.44 E3672 + 13.44 S3672 <= 0
541) X3673 - 13.44 E3673 + 13.44 S3673 <= 0
542) X3674 - 13.44 E3674 + 13.44 S3674 <= 0
543) X3675 - 13.44 E3675 + 13.44 S3675 <= 0
544) X3676 - 13.44 E3676 + 13.44 S3676 <= 0
545) X3677 - 13.44 E3677 + 13.44 S3677 <= 0

546) X5101 - 2.112 E5101 + 2.112 S5101 <= 0
547) X5102 - 2.112 E5102 + 2.112 S5102 <= 0
548) X5103 - 2.112 E5103 + 2.112 S5103 <= 0
549) X5104 - 2.112 E5104 + 2.112 S5104 <= 0
550) X5105 - 2.112 E5105 + 2.112 S5105 <= 0
551) X5106 - 2.112 E5106 + 2.112 S5106 <= 0
552) X5107 - 2.112 E5107 + 2.112 S5107 <= 0
553) X5111 - 2.112 E5111 + 2.112 S5111 <= 0
554) X5112 - 2.112 E5112 + 2.112 S5112 <= 0
555) X5113 - 2.112 E5113 + 2.112 S5113 <= 0
556) X5114 - 2.112 E5114 + 2.112 S5114 <= 0
557) X5115 - 2.112 E5115 + 2.112 S5115 <= 0
558) X5116 - 2.112 E5116 + 2.112 S5116 <= 0
559) X5117 - 2.112 E5117 + 2.112 S5117 <= 0
560) X5121 - 2.112 E5121 + 2.112 S5121 <= 0
561) X5122 - 2.112 E5122 + 2.112 S5122 <= 0

562) X5123 - 2.112 E5123 + 2.112 S5123 <= 0
563) X5124 - 2.112 E5124 + 2.112 S5124 <= 0
564) X5125 - 2.112 E5125 + 2.112 S5125 <= 0
565) X5126 - 2.112 E5126 + 2.112 S5126 <= 0
566) X5127 - 2.112 E5127 + 2.112 S5127 <= 0

567) X14101 - 3.072 E14101 + 3.072 S14101 <= 0
568) X14102 - 3.072 E14102 + 3.072 S14102 <= 0
569) X14103 - 3.072 E14103 + 3.072 S14103 <= 0
570) X14104 - 3.072 E14104 + 3.072 S14104 <= 0
571) X14105 - 3.072 E14105 + 3.072 S14105 <= 0
572) X14106 - 3.072 E14106 + 3.072 S14106 <= 0
573) X14107 - 3.072 E14107 + 3.072 S14107 <= 0
574) X14111 - 3.072 E14111 + 3.072 S14111 <= 0
575) X14112 - 3.072 E14112 + 3.072 S14112 <= 0
576) X14113 - 3.072 E14113 + 3.072 S14113 <= 0
577) X14114 - 3.072 E14114 + 3.072 S14114 <= 0
578) X14115 - 3.072 E14115 + 3.072 S14115 <= 0
579) X14116 - 3.072 E14116 + 3.072 S14116 <= 0
580) X14117 - 3.072 E14117 + 3.072 S14117 <= 0
581) X14121 - 3.072 E14121 + 3.072 S14121 <= 0
582) X14122 - 3.072 E14122 + 3.072 S14122 <= 0
583) X14123 - 3.072 E14123 + 3.072 S14123 <= 0
584) X14124 - 3.072 E14124 + 3.072 S14124 <= 0
585) X14125 - 3.072 E14125 + 3.072 S14125 <= 0
586) X14126 - 3.072 E14126 + 3.072 S14126 <= 0
587) X14127 - 3.072 E14127 + 3.072 S14127 <= 0

588) X46131 - 14.4 E46131 + 14.4 S46131 <= 0
589) X46132 - 14.4 E46132 + 14.4 S46132 <= 0
590) X46133 - 14.4 E46133 + 14.4 S46133 <= 0
591) X46134 - 14.4 E46134 + 14.4 S46134 <= 0
592) X46135 - 14.4 E46135 + 14.4 S46135 <= 0
593) X46136 - 14.4 E46136 + 14.4 S46136 <= 0
594) X46137 - 14.4 E46137 + 14.4 S46137 <= 0
595) X46141 - 14.4 E46141 + 14.4 S46141 <= 0
596) X46142 - 14.4 E46142 + 14.4 S46142 <= 0
597) X46143 - 14.4 E46143 + 14.4 S46143 <= 0
598) X46144 - 14.4 E46144 + 14.4 S46144 <= 0
599) X46145 - 14.4 E46145 + 14.4 S46145 <= 0
600) X46146 - 14.4 E46146 + 14.4 S46146 <= 0
601) X46147 - 14.4 E46147 + 14.4 S46147 <= 0
602) X46151 - 14.4 E46151 + 14.4 S46151 <= 0
603) X46152 - 14.4 E46152 + 14.4 S46152 <= 0
604) X46153 - 14.4 E46153 + 14.4 S46153 <= 0
605) X46154 - 14.4 E46154 + 14.4 S46154 <= 0
606) X46155 - 14.4 E46155 + 14.4 S46155 <= 0
607) X46156 - 14.4 E46156 + 14.4 S46156 <= 0
608) X46157 - 14.4 E46157 + 14.4 S46157 <= 0

609) XP291 - 40.8 EP291 + 40.8 SP291 <= 0
610) XP292 - 40.8 EP292 + 40.8 SP292 <= 0

611) XP293 - 40.8 EP293 + 40.8 SP293 <= 0
612) XP294 - 40.8 EP294 + 40.8 SP294 <= 0
613) XP295 - 40.8 EP295 + 40.8 SP295 <= 0
614) XP296 - 40.8 EP296 + 40.8 SP296 <= 0
615) XP297 - 40.8 EP297 + 40.8 SP297 <= 0
616) XP291 - 14.4 EP291 + 14.4 SP291 >= 0
617) XP292 - 14.4 EP292 + 14.4 SP292 >= 0
618) XP293 - 14.4 EP293 + 14.4 SP293 >= 0
619) XP294 - 14.4 EP294 + 14.4 SP294 >= 0
620) XP295 - 14.4 EP295 + 14.4 SP295 >= 0
621) XP296 - 14.4 EP296 + 14.4 SP296 >= 0
622) XP297 - 14.4 EP297 + 14.4 SP297 >= 0

/*Melter capacity for line 1 */
623) - X1 + X2211 + X2212 + X2213 + X2214 + X2215 + X2216 + X2217
+ X3921 + X3931 + X3941 + X3951 + X3961 + X3971 + X3922 + X3932
+ X3942 + X3952 + X3962 + X3972 + X3923 + X3933 + X3943 + X3953
+ X3963 + X3973 + X3924 + X3934 + X3944 + X3954 + X3964 + X3974
+ X3925 + X3935 + X3945 + X3955 + X3965 + X3975 + X3926 + X3936
+ X3946 + X3956 + X3966 + X3976 + X3927 + X3937 + X3947 + X3957
+ X3967 + X3977 + X3621 + X3631 + X3641 + X3651 + X3661 + X3671
+ X3622 + X3632 + X3642 + X3652 + X3662 + X3672 + X3623 + X3633
+ X3643 + X3653 + X3663 + X3673 + X3624 + X3634 + X3644 + X3654
+ X3664 + X3674 + X3625 + X3635 + X3645 + X3655 + X3665 + X3675
+ X3626 + X3636 + X3646 + X3656 + X3666 + X3676 + X3627 + X3647
+ X3657 + X3667 + X3677 <= 0
624) X1 <= 11760

/* Melter capacity for line 2 */
625) - X2 + XP291 + XP292 + XP293 + XP294 + XP295 + XP296 + XP297
+ X5101 + X5111 + X5121 + X5102 + X5112 + X5122 + X5103 + X5113
+ X5123 + X5104 + X5114 + X5124 + X5105 + X5115 + X5125 + X5106
+ X5116 + X5126 + X5107 + X5117 + X5127 + X14101 + X14111 + X14121
+ X14126 + X14107 + X14117 + X14127 + X46131 + X46141 + X46151
+ X46132 + X46142 + X46152 + X46133 + X46143 + X46153 + X46134
+ X46144 + X46154 + X46135 + X46145 + X46155 + X46136 + X46146
+ X46156 + X46137 + X46147 + X46157 <= 0
626) X2 <= 840

/* Conveyor capacity constraints 627 through 640 */
627) X3921 + X3931 + X3941 + X3951 + X3961 + X3971 <= 48
628) X3621 + X3631 + X3641 + X3651 + X3661 + X3671 <= 48
629) X3922 + X3932 + X3942 + X3952 + X3962 + X3972 <= 48
630) X3622 + X3632 + X3642 + X3652 + X3662 + X3672 <= 48
631) X3923 + X3933 + X3943 + X3953 + X3963 + X3973 <= 48
632) X3623 + X3633 + X3643 + X3653 + X3663 + X3673 <= 48
633) X3924 + X3934 + X3944 + X3954 + X3964 + X3974 <= 48
634) X3624 + X3634 + X3644 + X3654 + X3664 + X3674 <= 48
635) X3925 + X3935 + X3945 + X3955 + X3965 + X3975 <= 48
636) X3625 + X3635 + X3645 + X3655 + X3665 + X3675 <= 48
637) X3926 + X3936 + X3946 + X3956 + X3966 + X3976 <= 48
638) X3626 + X3636 + X3646 + X3656 + X3666 + X3676 <= 48

```
   639) X3927 + X3937 + X3947 + X3957 + X3967 + X3977 <=  48
   640) X3627 + X3637 + X3647 + X3657 + X3667 + X3677 <=  48

/* Minimum and maximum allowable inventory constraints, 641 through 652*/
   641) I227 + UX227 =  80.6
   642) I227 - ON227 =  20.2

   643) I467 + UX467 =  1100
   644) I467 - ON467 =  100

   645) I147 + UX147 =  300
   646) I147 - ON147 =  3

   647) I367 + UX367 =  2000
   648) I367 - ON367 =  50

   649) I57 + UX57 =  160
   650) I57 - ON57 =  20

   651) I397 + UX397 =  4000
   652) I397 - ON397 =  900
 END
```

APPENDIX 2

Program for the simulation model.

```
/*      Below is the listing of the program which sequences the arriving jobs.
        The input to the program is a list of jobs witth additional information, such as due date,
        demand, production rate, and machine resources requirements.
        The output is a set of sequences which meet due dates, and the sequence with minimum
        throughput change.
*/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define SIZE 4
#define SIZEFACT 24
#define TRUE 1
#define FALSE 0

int column_no=0;


void permute(int order[SIZE][SIZEFACT], int que_size, int number_to_permute);
void copy_array(int sourcecol, int destcol, int matrix[SIZE][SIZEFACT]);
int in_array(int check_in_array[SIZE][SIZEFACT], int col_no, int no_to_check);
int in_vector(int check_in_vector[SIZEFACT], int no_to_check);
void initialize_array( int array_to_init[SIZE][SIZEFACT], int array_size, int columns);
int factorial( int input);
int check_due_date( int order_array[SIZE][SIZEFACT], float job_array[SIZE][4],
                        int order_no, int fin_time[SIZE][SIZEFACT]);
int check_all_due_dates (int order_arr[SIZE][SIZEFACT], float job_arr[SIZE][4],
                              int valid_order_nos[SIZEFACT], int fin_times[SIZE][SIZEFACT]);
int find_all_throughputs (int order_arr[SIZE][SIZEFACT],
                              float job_arr[SIZE][4], float thruputs[SIZEFACT]);
float minimum_throughput (float thruputs[SIZEFACT], int valid_order_nos[SIZEFACT],
                              float *min_of_all, int *min_order, int * min_valid_order);
void input_jobs ( float job_array[SIZE][4]);


main()
{    int vcount, rows, count;
        int size_of_que, no_permute;
        int order[SIZE][SIZEFACT];
        float min_valid_thruput, min_thruput;
        int min_thruput_order, min_valid_thruput_order;
        float job_array[SIZE][4];
        int valid_orders[SIZEFACT];
        int finish_times[SIZE][SIZEFACT];
```

```
        float throughputs[SIZEFACT];

        size_of_que=SIZE;
        no_permute=SIZE;
        input_jobs(job_array);
        initialize_array(order, SIZE, SIZEFACT);
        permute(order, size_of_que, no_permute);
        vcount = check_all_due_dates(order, job_array, valid_orders, finish_times);
        find_all_throughputs(order, job_array, throughputs);
        min_valid_thruput = minimum_throughput(throughputs, valid_orders,
                            &min_thruput, &min_thruput_order, &min_valid_thruput_order);
        printf("\n");
        printf("\n The number of valid orders is %d \n", vcount);
        printf("\n The valid orders are:");
        for (rows=0; rows< vcount; rows++)
        {
          printf("\nOrder No. %d \n", valid_orders[rows]);
          printf("The order of the jobs is \n");
          for (count=0; count<SIZE; count++)
          {
            printf("%d \t", order[count][valid_orders[rows]]);
          }
          printf("\nThe throughput is %f \n", throughputs[valid_orders[rows]]);
        }
        printf("\nThe minimum throughput is %f for %d order \n", min_thruput,
min_thruput_order);
        printf("\nThe minimum valid throughput is %f for order %d \n", min_valid_thruput,
min_valid_thruput_order);
}


/*      The function input_jobs  handles the inputting of information about the jobs      */

void input_jobs (float job_array[SIZE][4])
{
        char in_filename[12];
        FILE *in_fp;
        int job_row, q_size;

        printf("\n \nWhat is the name of the file you want to read with job_details ?");
        gets(in_filename);
        if((in_fp=fopen(in_filename,"r"))==NULL)
        {
        printf("\n\n %s does not exist.\n", in_filename);
        exit(1);
        }
        fscanf(in_fp, "%d \n", &q_size);
        if (q_size != SIZE)
        {
        printf("\n Size of the input job not valid");
        }
        for( job_row=0; job_row < SIZE; job_row++)
```

```
        {
        fscanf(in_fp, "%f %f %f %f\n", &job_array[job_row][0],
        &job_array[job_row][1], &job_array[job_row][2], &job_array[job_row][3]);
        }
        fclose(in_fp);
}

int factorial (int input)
{
  int fcount, prod;
  prod=1;
  for ( fcount=1; fcount<=input; fcount++)
        prod=prod*fcount;
  return prod;
}

void initialize_array( int array_to_init[SIZE][SIZEFACT], int array_size, int columns)
{
        int counter, colcount;
        for (colcount=0; colcount< columns; colcount++)
        {
          for (counter=0; counter< array_size; counter++)
            array_to_init[counter][colcount]=0;
        }
}

/* Function in_array checks whether a number to check is in an array */

int in_array(int check_in_array[SIZE][SIZEFACT], int col_no, int no_to_check)
{
        int count;
        for(count=0; count<SIZE; count++)
        {
          if (no_to_check==check_in_array[count][col_no])
                return TRUE;
        }
        return FALSE;

}

void copy_array(int sourcecol, int destcol, int matrix[SIZE][SIZEFACT])
{
        int copycount;
        for (copycount=0; copycount<SIZE; copycount++)
                matrix[copycount][destcol]=matrix[copycount][sourcecol];
        return ;
}

/*      Function permute generates all possible permutations of a specified number
         of elements from a vector.
*/
```

```
void permute(int order[SIZE][SIZEFACT], int que_size, int number_to_permute )
{
        int icount, li, lj, callperm;
        callperm=0;

        for(icount=1; icount<=SIZE; icount++)
        {
                if ( in_array(order,column_no, icount))
                  { continue; }
                else
                {
                callperm =  callperm + factorial(number_to_permute -1);
                if(callperm <= factorial(number_to_permute))
                {
                  if ( number_to_permute == 1)
                    {
                     order[que_size - number_to_permute][column_no] = icount;
                         if (column_no< SIZEFACT)
                                 column_no++;
                         return;
                    }
                  else
                    {
                     order[que_size - number_to_permute][column_no] = icount;
                     lj = (factorial(number_to_permute-1) - 1);
                     for ( li=1; li <= lj; li++)
                           {
                           copy_array(column_no, column_no+li, order);
                           }
                           permute(order, que_size, number_to_permute-1);
                          }
                  }       /* if callperm */
                }
        }
          return;
}

/*      This function checks whether a job meets its due date */

int check_due_date( int order_array[SIZE][SIZEFACT], float job_array[SIZE][4], int order_no, int
fin_time[SIZE][SIZEFACT])
{
  int job_row, order_row, valid;
  if (order_no > SIZEFACT)
        return FALSE;
  valid=TRUE;
  for (order_row=0; order_row < SIZE; order_row++ )
        {
         job_row = order_array[order_row][order_no] - 1;
         if (order_row == 0)
           fin_time[order_row][order_no]=job_array[job_row][1];
         else
```

```
                fin_time[order_row][order_no] = fin_time[order_row -1][order_no] +
job_array[job_row][1];
              if (fin_time[order_row][order_no] > job_array[job_row][2])
                    valid=FALSE;
            }
    return valid;
}

/*      This function checks due dates for all the jobs in a sequence of jobs */

int check_all_due_dates ( int order_arr[SIZE][SIZEFACT], float job_arr[SIZE][4], int
valid_order_nos[SIZEFACT], int fin_times[SIZE][SIZEFACT])
{
     int order_col, all, valid_count;
     valid_count=0;
     for (order_col=0; order_col<SIZEFACT; order_col++)
     {
            if ( check_due_date(order_arr, job_arr, order_col, fin_times) )
              {
              valid_order_nos[valid_count] = order_col;
              if (valid_count==0)
              {
               for(all=0; all< SIZEFACT; all++)
                    valid_order_nos[all] = order_col;
              }
              valid_count++;
              }
     }
    return valid_count;
}

/*      This function calculates the total throughput change of a job sequence */

int find_all_throughputs (int order_arr[SIZE][SIZEFACT],
                              float job_arr[SIZE][4], float thruputs[SIZEFACT])
{
  int this_job_no, next_job_no, order_row, order_col;
  float thruput_change;

  for (order_col=0; order_col< SIZEFACT; order_col++)
     {
     thruput_change=0;
     for (order_row=0; order_row < SIZE - 1; order_row++)
              {
              this_job_no = order_arr[order_row][order_col]-1;
              next_job_no = order_arr[order_row + 1][order_col]-1;
              thruput_change+= fabs(job_arr[this_job_no][3] - job_arr[next_job_no][3]);
              }
    thruputs[order_col] = thruput_change;
     }
  return 0;
}
```

```
int in_vector(int check_in_vector[SIZEFACT], int no_to_check)
{
        int count;
        for(count=0; count<SIZEFACT; count++)
        {
          if (no_to_check==check_in_vector[count])
                return TRUE;
        }
        return FALSE;

}

/*      This function calculates the minimum throughput computed for all sequences */
float minimum_throughput (float thruputs[SIZEFACT], int valid_order_nos[SIZEFACT],
                              float *min_of_all, int *min_order, int *min_valid_order)
{
  float temp_min_valid, temp_min_all, critical_thruput;
  int count;
  count=0;
  critical_thruput = 0.01;
  *min_order = 0;
  temp_min_all = thruputs[count];
  *min_valid_order = valid_order_nos[0];
  temp_min_valid = thruputs[*min_valid_order];

  for (count=0; count< SIZEFACT; count++)
     {
        if(temp_min_all <= critical_thruput) BREAK;
        if (temp_min_all > thruputs[count])
           { temp_min_all = thruputs[count];
             *min_order = count;
           }
        if ( in_vector(valid_order_nos, count) )
           {
           if (temp_min_valid > thruputs[count])
             { temp_min_valid = thruputs[count];
                 *min_valid_order = count;
             }
           }
        }
    *min_of_all = temp_min_all;
        return temp_min_valid;
}
```

APPENDIX 3A

Sensitivity of Starting Inventory.

Sensitivity Analysis: Sensitivity analysis allows insight into the behavior of the system in case the forecasted demand orders, quantity or due dates change. The increase and decrease columns in these tables specify the range in which the basis or the production results from the LP remains same, however values of production quantity, and production times might change. The sensitivity of results to inventory and demands for products 22 on line 1 and product 5 on line 2 is shown in Tables 1 and Table 2. The sensitivity for other products also follow in Tables 3 through 6.

Table 1: Sensitivity analysis for product 22.

Time period	Increase	Decrease	RHS
1	0	.2	.7
2	.2	0	.5
3	.2	0	0
4	13.44	0	.2
5	32.2	0	0
6	15.8	32.2	12
7	15.8	20.2	0

During time period 1 no demand arrived but the buffer had 0.7 inventory. Therefore, any production would mean that the inventory for time period 1 would increase by that amount over 0.7 units, since the constraint is of the format:

Inventory_Now - Production_Now = Inventory_Last Period.

The range analysis shows that taking into consideration the production capacity and demands for all time periods the starting inventory could have been reduced by as much as 0.2 units. For time periods 2 through 7 the format of the constraint is as follows:

Inventory_Last Period + Production_Now = Inventory_Now + Demand_Now.

For time periods 2 and 3, either the demands or the present inventory (implying production now) could have been more by 0.2 units, implying that the system could have handled an increase in demand during the time periods 2 and 3, maintaining the same basis but different values. Similarly the demands or the inventory now for the time periods 4 and 5 could handle an increase of 13.44 and 32.2 units, respectively whereas for the time periods 6 and 7 an increase by 15.8 units and a decrease by 32.2 and 20.2 units in demand or the inventory would be allowed by the system.

Table 2: Sensitivity analysis for product 5.

Time period	Increase	Decrease	RHS
1	126.64	8.15	9.15
2	8.15	126.65	0
3	8.15	126.65	0
4	8.15	126.65	1.0
5	13.354	126.65	0.14
6	13.354	126.65	0
7	13.354	126.65	0

For the product 5 since no demand arrived on the first time unit, the format of the constraint for the first time period is as follows:

Inventory_Now - Production_Now = Inventory_Last Period.

This implies that the starting or the initial inventory can be either increased or decreased by 126.64 or 8.15 units, respectively without changing the current basis. For time periods 2 through 7 the format of the constraint are as follows:

Inventory_Last Period + Production_Now = Inventory_Now + Demand_Now.

However, since there is no demand during the time periods 2 and 3 the inventory can be increased or decreased by 8.15 or 126.65 units. For time periods 4 and 5 since there were arrival of demands so either the inventory now or the demands can be increased or decreased by 8.15 or 126.65 units for the time period 4 and 13.354 or 126.65 for the time period 5. Again for the time periods 6 and 7 there are no demand arrivals so the inventory now can be increased or decreased by 13.354 or 126.65 units.

Table 3: Sensitivity analysis for product 39.

Time period	Increase	Decrease	RHS
1	12.51	11.48	816.6
2	11.48	12.52	97.13
3	11.48	12.52	0
4	11.48	12.52	0
5	11.48	12.52	0
6	11.48	12.52	0
7	11.48	12.52	0

Table 4: Sensitivity analysis for product 36.

Time period	Increase	Decrease	RHS
1	1769.16	180.84	279.03
2	180.84	1769.16	0
3	180.84	1769.16	24.03
4	180.84	1769.16	0
5	180.84	1769.16	0
6	180.84	1769.16	0
7	180.84	1769.16	72.15

Table 5: Sensitivity analysis for product 14.

Time period	Increase	Decrease	RHS
1	.472	7.53	2.6
2	.472	7.53	0
3	.472	7.53	0
4	29.34	7.53	8
5	0	0	0
6	0	0	0
7	52.3	244.704	0

Table 6: Sensitivity analysis for product 46.

Time period	Increase	Decrease	RHS
1	830.2	10.6	35
2	10.6	830.2	25
3	10.6	830.2	0
4	10.6	830.2	0
5	10.6	830.2	0
6	97	830.2	0
7	169.8	830.2	0

APPENDIX 3B

Sensitivity of Due Dates.

Due Date Sensitivity for the products:

Tables 7 and 8 for products 22 (of line 1) and product 5 (of line 2) suggest that the due dates for the demands that arrived on the 'demand arrival' days can be increased or decreased by respective time units for the solution to remain feasible. 'INF' suggests a large amount of time. Tables 9 through 12 show sensitivity of other products.

Table 7: Sensitivity analysis for Due Dates for product 22.

Due Date	Demand Arrival	Increase	Decrease
2.5	2	4.5	1.5
4.5	4	2.5	3.5
12.5	6	INF.	4.12

Table 8: Sensitivity analysis for Due Dates for product 5.

Due Date	Demand Arrival	Increase	Decrease
4.5	4	2.5	3.5
7	5	0	INF.

Table 9: Sensitivity analysis for Due Dates for product 39.

Due Date	Demand Arrival	Increase	Decrease
2	1	0.52	INF
4	2	INF	3.0

Table 10: Sensitivity analysis for Due Dates for product 36.

Due Date	Demand Arrival	Increase	Decrease
6	3	1	5
7.5	7	INF	0.5

Table 11: Sensitivity analysis for Due Dates for product 14.

Due Date	Demand Arrival	Increase	Decrease
2	1	2.45	0.15
5	4	INF	3.18

Table 12: Sensitivity analysis for Due Dates for product 46.

Due Date	Demand Arrival	Increase	Decrease
2	1	5	1
3	2	4	2

APPENDIX 3C

Sensitivity of Inventory Limits.

Inventory Limits: Ideally a maximum allowable inventory should be a very large amount and the minimum allowable inventory should be zero, so that no restrictions arise due to inventory. Tables 13 and 14 for products 22 and 5 suggest the allowable increase and decrease in the minimum and the maximum inventories.

Table 13: Sensitivity of Inventory Limits for product 22.

Max. Level	Increase	Decrease
80.6	INF.	60.4
Min. Level	Increase	Decrease
20.2	15.8	20.2

Table 14: Sensitivity of Inventory Limits for product 5.

Max. Level	Increase	Decrease
160	INF.	126.646
Min. Level	Increase	Decrease
20	13.354	INF.

Table 15: Sensitivity of Inventory Limits for product 46.

Max Level	Increase	Decrease
1100	INF	830.2
Min Level	Increase	Decrease
100	169.8	INF

Table 16: Sensitivity of Inventory Limits for product 14.

Max Level	Increase	Decrease
300	INF	244.74
Min Level	Increase	Decrease
3	52.3	INF

Table 17: Sensitivity of Inventory Limits for product 36.

Max Level	Increase	Decrease
2000	INF	1769.16
Min Level	Increase	Decrease
50	180.84	INF

Table 18: Sensitivity of Inventory Limits for product 39.

Max Level	Increase	Decrease
4000	INF	3100.0
Min Level	Increase	Decrease
900	11.48	12.51

APPENDIX 3D

Sensitivity of Melter Capacity.

Capacity: The constant capacity of each melter forces a restriction on the amount of products that can be produced. Tables 19 and 20 show the capacity sensitivity of the two manufacturing lines.

Table 19: Sensitivity analysis for Capacity of Line 1.

RHS	Increase	Decrease
11760	INF.	11547.28

The capacity of the line 1 is 11760 units. The current solution remains feasible and optimal over the range 212.72 to infinity that is, it can be increased to a large amount and can not be reduced more than 11547.28 units which is to say that total of (11760-11547.28=212.72 units) have to be manufactured.

Table 20: Sensitivity analysis for Capacity of Line 2.

RHS	Increase	Decrease
840	INF.	489.56

Similarly the capacity of the line 2 is 840 units, this can be increased to a large amount but cannot be reduced by more than 489.56 units.

APPENDIX 4A

Input/Output of Line 1 from simulation model.

```
/* The following is the required input file for Line 1 */

4
1.0 1.2436 2.0 3.0
2.0 0.597 7.0 2.6
3.0 3.0 15.0 2.6
4.0 2.1592 15.0 3.7

/* The output from the simulation model for Line 1 is given below */

What is the name of the file you want to read with job_details ?

{ user input e.g. Line1.ip }

/* The output for different orders is now given */

 The number of valid orders is 8

 The valid orders are:
Order No. 0
The order of the jobs is
1       2       3       4
The throughput is 1.500000

Order No. 1
The order of the jobs is
1       2       4       3
The throughput is 2.600000

Order No. 2
The order of the jobs is
1       3       2       4
The throughput is 1.500000

Order No. 3
The order of the jobs is
1       3       4       2
The throughput is 2.600000

Order No. 4
The order of the jobs is
1       4       2       3
The throughput is 1.800000
```

Order No. 5
The order of the jobs is
1 4 3 2
The throughput is 1.800000

Order No. 6
The order of the jobs is
2 1 3 4
The throughput is 1.900000

Order No. 7
The order of the jobs is
2 1 4 3
The throughput is 2.200000

The minimum throughput is 1.100000 for 8 order

The minimum valid throughput is 1.500000 for order 0

APPENDIX 4B

Input/Output of Line 2 from simulation model.

```
/* The following is the required input file for Line 2 */

5
1.0 1.15 2.0 0.6
2.0 2.0 4.5 0.5
3.0 1.0 7.0 0.6
4.0 2.0 7.0 0.5
5.0 .81 15.0 0.5

/* The output from the simulation model for Line 2 is given below */


What is the name of the file you want to read with job_details ?

{ user input e.g. Line 2.ip }

/* The output for different orders is now given */


 The number of valid orders is 20

 The valid orders are:
Order No. 0
The order of the jobs is
1       2       3       4       5
The throughput is 0.300000

Order No. 1
The order of the jobs is
1       2       3       5       4
The throughput is 0.300000

Order No. 2
The order of the jobs is
1       2       4       3       5
The throughput is 0.300000

Order No. 3
The order of the jobs is
1       2       4       5       3
The throughput is 0.200000

Order No. 4
```

The order of the jobs is

1 2 5 3 4

The throughput is 0.300000

Order No. 5

The order of the jobs is

1 2 5 4 3

The throughput is 0.200000

Order No. 6

The order of the jobs is

1 3 2 4 5

The throughput is 0.100000

Order No. 7

The order of the jobs is

1 3 2 5 4

The throughput is 0.100000

Order No. 10

The order of the jobs is

1 3 5 2 4

The throughput is 0.100000

Order No. 18

The order of the jobs is

1 5 2 3 4

The throughput is 0.300000

Order No. 19

The order of the jobs is

1 5 2 4 3

The throughput is 0.200000

Order No. 20

The order of the jobs is

1 5 3 2 4

The throughput is 0.300000

Order No. 48

The order of the jobs is

3 1 2 4 5

The throughput is 0.100000

Order No. 49

The order of the jobs is

3 1 2 5 4

The throughput is 0.100000

Order No. 52

The order of the jobs is

3 1 5 2 4

The throughput is 0.100000

Order No. 66
The order of the jobs is
3 5 1 2 4
The throughput is 0.300000

Order No. 96
The order of the jobs is
5 1 2 3 4
The throughput is 0.400000

Order No. 97
The order of the jobs is
5 1 2 4 3
The throughput is 0.300000

Order No. 98
The order of the jobs is
5 1 3 2 4
The throughput is 0.200000

Order No. 108
The order of the jobs is
5 3 1 2 4
The throughput is 0.200000

The minimum throughput is 0.100000 for 6 order

The minimum valid throughput is 0.100000 for order 6