Reduction of Average Cycle Time at a Wafer Fabrication Facility

by

Sameer T. Shikalgar

Thesis submitted to the Faculty of Virginia State Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Industrial and Systems Engineering

Subhash Sarin, Chairman

Patrick Koelling

Dean Teague

August 25th, 2000

Blacksburg, Virginia

Keywords: Scheduling, Cycle time, Simulation

Reduction of Average Cycle Time at a Wafer Fabrication Facility

by

Sameer T. Shikalgar

Dr. Subhash C. Sarin, Chairman
Industrial and Systems Engineering

ABSTRACT

This research is concerned with the development of effective solutions for the reduction of average cycle time at a wafer fabrication facility. The wafer fabrication environment is quite different from the usual flow shop or job shop environments, with a distinguishing feature being the reentrant flow of the lots through the system. Lots at different stages of their manufacturing cycle may revisit the machines. This gives rise to the need of effective policies to sequence lots through the system.

The study is being conducted on a M/A-COM's wafer fabrication system. The facility on which the study is being conducted is based in Roanoke, VA. The facility consists of 92 machines and its products can be classified into six different types. The data required for each product such as routing, processing times, yield at each operation etc. have been acquired from the facility. Two methodologies have been developed to effect a reduction in the cycle time of the products at M/A-COM's facility. The first methodology is heuristic procedure based on the idea of reducing idle time on the bottleneck machine. The second methodology is based on mathematical programming.

For the first methodology, the manufacturing system is simulated using AutoSchedAP, which is part of the AutoSimulations Inc. software package. The software enables the accurate modeling of the existing system using actual part routes, station definitions, operator definitions, shift calendars, input orders, machine breakdowns and processing

and setup time distributions. The proposed approach, to reduce cycle time, is based on the principle of reduction of idle time at the bottleneck machine. The bottleneck machine controls the throughput of the system and any unnecessary idle time at the bottleneck leads to an increase in the average cycle time. The AutoSchedAP software enables the user to write custom scheduling rules using C++ and integrate it with the simulation model. The performance of the proposed procedure is compared with those of various other scheduling rules in the software.

The second proposed methodology models the system as an integer program. The integer program reads the various machine and product data and establishes the optimal flow of the lots through the system. The integer program uses the start time of lots, at various operations, as variables and outputs the time at which each lot should be started at each operation. The integer program is solved using CPLEX, which is a linear and integer programming software. Presently, various methods are being analyzed to relax the integer program into an equivalent linear/nonlinear program, since solving an integer program consumes a lot of time, even for small problems.

A third methodology has also been proposed. This methodology concerns the modeling of the system on the basis of a conjunctive-disjunctive graph. The main idea in this methodology is that the minimization of maximum lateness at each operation would result in the minimization of maximum cycle time of the overall system.

Some preliminary results obtained are presented. Also, the work in progress is described.

Table of Contents

List of Tables

List of Figures

Chapter 1: Motivation and Problem Statement

1.1 Introduction

The importance of semiconductor wafer fabrication scheduling has been increasingly steadily over the past decade. Wafer fabrication is the most technologically complex and capital intensive phase in semiconductor manufacturing. It involves the processing of wafers of silicon and gallium arsenide in order to build up the layers and patterns of metal and wafer material. Many operations have to be performed in a clean room environment to prevent particulate contamination of wafers. Also, since the machines on which the wafers are processed are expensive, service contention is an important concern. All these factors underline the importance of seeking policies to optimize the performance with respect to some objective. M/A-COM is one such group of companies which has various wafer fabrication facilities in the U.S.

M/A-COM Inc., at its Virginia based Roanoke facility, provides Gallium Arsenide (GaAs) integrated circuit solutions (IC) to the commercial and military wireless and radar markets. M/A-COM designs and manufactures a broad line of GaAs microwave and millimeter wave products using wafer fabrication techniques. Depending on the type of circuit design, a wafer goes through 100 to 200 process steps over a period of a few weeks. Another unique property of wafer fabrication is that the fabrication is not linear through the facility but wafers may revisit machines during different stages of their manufacturing. Wafer fabrication planning and control is complicated by random disturbances, namely, the reentrant flow of operations, random yields due to the special nature of the operations, random machine breakdowns and random repair times.

M/A-COM is in the process of remodeling the operations of its wafer fabrication area, at its Roanoke facility. The motivation behind this remodeling effort is to reduce the cycle time and improve the overall efficiency. Presently it is observed that the lots spend a substantial portion of their cycle time waiting in queue for a machine. This leads to an increase in the overall Work-In-Process (WIP) and a reduction in the throughput.

1.2 Manufacturing Environment

Semiconductor wafer fabrication can be described as a multistage process, with reentrant flow, in which the equipment is arranged similar to a job shop. Some of the typical operations include photolithography, diffusion, metalization, ion implantation (doping) and etching. Depending on the type of circuit design, a wafer goes through 100 to 200 process steps over a period of a few weeks. Wafer fabrication planning and control is a complex job due to the large number of wafers and machines involved. It is further complicated by random disturbances, namely, the reentrant flow of operations, random yields due to the special nature of the operations, random machine breakdowns and random repair times. Improvements in this area make dramatic differences in the performance of the semiconductor manufacturing facility (fab).

M/A COM's clean room manufacturing environment consists of 5 different areas. These are Photolithography, Process I, Process II, Materials, and Implant, which are shown in Figure 1.1. Typically, wafers are prepared for manufacturing and introduced to manufacturing in the Materials area. Wafers are then transported to Photolithography Area to apply the desired circuit patterns with a substance called photoresist. The patterns are cleared from photoresist in the Process I Area. These patterns are then implanted with the desired ions to give the necessary properties to the pattern at the Implant Area. In the Process II area, metal is deposited to obtain the "Gate" metal. This is a typical sequence to form certain properties on a circuit. It is followed several times before the circuit is completed.

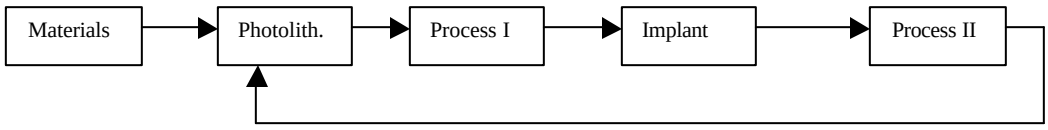


Figure 1.1: Manufacturing Areas at M/A COM

1.3 Objective

The aim of the research is to decrease the average amount of time that a lot spends in the system. The main areas for the analysis of this problem can be identified as follows :

- **Operations Bottlenecks**: Bottleneck machines govern the cycle time of the entire operation. Identification of bottleneck machines is essential so as to sequence the jobs in an optimal way at those machines. Optimal sequencing at the bottleneck machines reduces the cycle time considerably.
- **Work-in-Process (WIP)**: Excessive WIP represents low efficiency of the operation. It is necessary to keep WIP at a minimum to reduce locked up capital and to ensure the smooth flow of jobs through the system.
- **Equipment utilization**: Equipment utilization needs to be monitored to ensure effective utilization of all the machines. Under- utilization of machines can result in increased delays and cycle time. Equipment utilization needs to be maintained at some predetermined level to obtain increase in efficiency.

1.4 Problem Statement

The research conducted is mainly concerned with the following problem: given the number of lots in queue at a particular workstation, which lot needs to be processed next so that the average cycle time of the entire system is reduced. This problem is entirely a sequencing problem since the sequence for the processing of the lots at a particular machine is considered.

The problem considered is within a semiconductor manufacturing environment. Such plants feature a characteristic reentrant process flow where jobs, at different stages of production, compete for the same machine, for their processing. This additional feature of semiconductor manufacturing makes the problem dissimilar to the common job shop or flow shop problems.

A few unique issues arise when considering scheduling in a semiconductor manufacturing environment. First, which lots need to be processed first so as to reduce average cycle time? Lots at the same machine could either be at the start of their manufacturing route or towards the end or anywhere in between. Hence there is a need to determine if lots, in any one of the above-mentioned classes, dominate over the others in terms of reducing the objective. If lots in one class consistently outperform the others, then it would be beneficial to schedule those first so as to reduce mean cycle time.

However, if no one class is dominant, other factors need to be considered to attain the minimum average cycle time.

The second issue relates to the lot size of the parts flowing through the system. Each lot consists of a number of identical wafers. These wafers flow through the system as a single unit and are never split. The number of wafers in a lot is determined by the cassette size (in which the wafers are kept) and the number of wafers that can be processed at the same time on each machine. If the time taken by each lot to go from one machine to the other is negligible, it would be beneficial to have a lot size equal to the smallest number of lots a machine can take. If, however, the transit times are large, it would be advantageous to have larger lot sizes.

The third issue relates to the capacity of each machine. Some machines may be capable of processing only a single lot at a time. However, there are a few machines which are able to process multiple lots at a time. This depends on the type of lots and the type of operation that needs to be done. These machines have a significant impact on the cycle time of the products and, hence, this factor needs to be taken into account while considering the sequencing of the lots.

1.4.1 Motivation behind analysis

One scheduling policy is to minimize the average cycle time of all the products in the system. However, due to the reentrant flow characteristic, production planning and scheduling in the semiconductor industry becomes particularly difficult. This is because of the following factors:

1. Complexity of Product Flows: The number of process steps, for each product, is high and a number of these steps take place on the same production equipment. This increases the complexity of the problem and makes the search for a near optimal schedule difficult.
2. Diverse Equipment Characteristics: The characteristics of the equipment used in semiconductor manufacturing vary widely. Some machines have significant sequence dependent setup times, while others do not.
3. Equipment Downtime: The production equipment used in semiconductor manufacturing is technologically, extremely sophisticated. It requires extensive preventive maintenance and calibration and is still subject to unpredictable failures.

The main focus of manufacturing strategies in the semiconductor industry is on minimizing production costs and increasing productivity while improving both quality and delivery time performance. The reduction of the average cycle time is an appropriate criteria that applies to the realization of manufacturing efficiency of the semiconductor industry.

1.4.2 Difference of Semiconductor Manufacturing Plants from Traditional Manufacturing Plants

Manufacturing systems can be broadly divided into two categories – flow shops and job shops. Flow shops typically have an acyclic route through the plant, never visiting one station more that once. Flow shops are dedicated to the production of a single type of part in high volume.

Job shops typically consist of a variety of machine types. The routes of the jobs vary depending on the operations required for each part. The routes here may revisit a machine. This type of machine environment is used for medium volume products.

With the advent of semiconductor manufacturing, a new type of system has emerged which does not fit entirely to either of the classes mentioned above. Unlike flow shops, this type of flow is reentrant, but unlike jobs shops the flow is more structured and less random.

1.5 Research Objectives

Given the semiconductor manufacturing environment, the objective of the research is to model the system and formulate a scheduling algorithm which would result in the reduction of the average cycle time of all the products in the system. The algorithm needs to be applied in a dynamic environment in which the lots enter and exit the system periodically. Hence, the problem can be stated as follows:

- There are a variety of parts of different types, which enter and exit the system dynamically.
- There are different types of machines, which can be used only for specific operations.

- The lots follow a reentrant type of route through the system and hence a single lot may visit a machine many times during its manufacturing cycle.
- The machines are subject to breakdowns and calibrations.
- Some of the machines may be single-lot processing machines while the others may be multiple-lot processing machines.

Hence this system, though similar to a classical sequencing problem, has its own unique characteristics, which makes it different from the usual sequencing problems.

1.6 Proposed Methodologies

Two methodologies are proposed to reduce the cycle time of the wafer fabrication area. A brief description of each of these methodologies is as follows:

1. **Heuristic Algorithm based on a new Dispatching Rule**: The proposed approach, to reduce cycle time, is based on the principle of reduction of idle time at the bottleneck machine. The bottleneck machine controls the throughput of the system and any unnecessary idle time at the bottleneck leads to an increase in the average cycle time. This algorithm is applied to a real life semiconductor wafer fabrication facility. A simulation software AutoSched AP is used to model the functioning of MA/COM's facility in Roanoke, Virginia. The software enables the accurate modeling of the existing system using actual part routes, station definitions, operator definitions, shift calendars, input orders, machine breakdowns and processing and setup time distributions. The software enables the user to write custom scheduling rules using C++ and integrate it with the simulation model. The performance of the proposed procedure is compared with that of various other rules that the software provides.

2. **Mathematical Programming Based Approach**: Given the processing times of various operations and the routes that the products follow through the system, the entire problem can be formulated as an integer program. This integer program can be run on any linear and integer programming software to give optimal results. The integer program uses the start time of the jobs at various operations as variables. This can then be used to minimize any objective function such as mean cycle time or

maximum cycle time. This makes this approach flexible in terms of minimizing any objective function. Appropriate solution procedures are developed to solve this mathematical model.

3. **Conjunctive-Disjunctive Graph Based Approach**: Even though the wafer fabrication environment is dissimilar to a job shop or a flow shop environment, a few similarities do exist. As in the job shop, the products in a wafer fabrication environment follow different routes through the system. However in a job shop the routes are more random than in a wafer fabrication environment. Hence using this a conjunctive-disjunctive graph based approach is proposed which uses the routes that the products follow. Though initially developed for a job shop, with a few modifications it can be applied to a wafer fabrication environment. The objective of this approach is to minimize maximum cycle time by minimizing the maximum lateness on each of the machines. This approach has been only developed in theory and has not been applied or analyzed.

1.7 Description of the facility

M/A-COM is a world leading developer and manufacturer of radio frequency (RF) and microwave semiconductors, components and IP Networks to the wireless telecommunications and defense-related industries. M/A-COM's products include semiconductor devices, RF integrated circuits, passive and control devices, antennas, subsystems and systems. M/A-COM has nearly 3,000 employees, with several hundred degreed engineers on staff. With its corporate headquarters in Lowell, Massachussetts, the company maintains nearly 1.25 million square feet of operations space around the world. M/A-COM has more than 20 manufacturing locations and sales offices in the United States alone, with numerous locations spread throughout the European and Asia-Pacific markets – including sales offices in 33 countries.

At its facility at Roanoke, Virginia, M/A-COM manufactures gallium arsenide (GaAs) wafers. These wafers are 8” in diameter and flow through the system in the form of lots. The facility manufactures a variety of products, which can be split into essentially six different types, due to similarities among the products. The facility has a clean room

environment to prevent contamination of wafers and is presently operating at a throughput of 175 lots per week. The objective of the research is to reduce the time the lots spend in the system by increasing the throughput. The facility receives production orders, which serve as a start to the manufacturing of the lot. The lots then flow through the system according to their respective lot routes and typically each lot goes through 200-300 process steps.

1.7.1 Order fulfillment

The wafers at the M/A COM facility flow through the system in the form of lots. One lot of wafers consists of a cassette containing 8 wafers. This lot flows through the system as a single entity i.e. it is not broken into sublots.

When an order is placed for a certain number of wafers, the number of wafers required is divided by the yield ratio of the fabrication line. The number thus obtained is rounded to the nearest higher multiple of 8. Hence, the wafers can be divided into lots of 8 each. If the number obtained after using the yield ratio is very close to the nearest higher multiple of 8, the Production Scheduler usually monitors these lots so as to ensure that the required quantity of wafers is produced.

1.7.2 Product Flow

As mentioned earlier, the wafer fabrication area is divided into 5 different processing areas. All the lots flowing through the system are sequenced before they enter each processing area. Once sequenced, these lots flow within each processing area in the order in which they were originally sequenced. After they come out of each area, they are sequenced again for the next processing area. The rules by which they are sequenced are described in the subsequent section.

Within each processing area, the lot goes through a series of steps labeled a, b, c, … Lots at different stages of their manufacturing cycle may be present at each processing area. For example, in photolithography, lots undergoing operations 1010 and operation 3210 may be present, since both operations require photolithography. All the lots may not visit all the machine types within each processing area. Some lots may also skip some machine types. This depends on the type of product and its operation number. However, the flow within each process area will usually be unidirectional i.e. the lots will usually

flow in one direction and will not move back and forth within each process area. Also, all the lots need not start from the same machine within each process area. Hence, within each process area, it can be assumed that a flow shop type environment exists. However, lots of different types, visiting the same machine, may require different settings and the processing times for each also maybe different.

As mentioned earlier, the flow within the entire fabrication area does not follow a definite flow pattern.

1.7.3 Overview of Wafer Fabrication

This section will provide a basic overview of the wafer fabrication process. The basic operations performed on the wafer are explained.

The wafers used at M/A-COM are made of Gallium Arsenide and are 4" in diameter. This wafer undergoes a series of operations, at different layers, to acquire a certain chip design. The four basic wafer operations performed at the M/A-COM facility can be classified as:

1. Layering
2. Patterning
3. Doping
4. Heat Treatments

Each of these processes are briefly explained below:

1. Layering

Layering is an operation used to add thin layers to the wafer surface. These layers are either insulators, semiconductors or conductors. They are of different materials and are deposited by various techniques. The deposition techniques used at M/A-COM include chemical vapor deposition (CVD) and sputtering.

2. Patterning

Patterning is the series of steps that results in the removal of selected portions of the added surface layers. After removal, a pattern of the layer is left on the wafer surface. The patterning process is called photolithography. It is the patterning operation that creates the surface parts of the devices that make up a circuit. The goal of the operation is

to create in or on the wafer surface the parts of the device or circuit in exact dimensions required by the circuit design and to locate them in their proper location on the wafer surface.

3. Doping

Doping is the process that puts specific amounts of dopants in the wafer surface through openings in the surface layers. The purpose of the doping operation is to create either N-type or P-type pockets in the wafer surface. These pockets form the N-P junctions required for the operation of the transistors, diodes, capacitors and resistors in the circuit.

4. Heat Treatment

Heat treatments are the operations in which the wafer is simply heated or cooled to achieve specific results. In the heat treatment operations, no additional material is added or removed from the wafer.

1.7.4 Machine description

At a step, within each processing area, there maybe more than one machine of a given type. This is similar to a flexible flow shop type environment. The processing times at a given machine may depend on the number of pieces in a lot or maybe independent of the number of pieces in a lot. This is useful in deciding whether lot splitting will help in improving the performance of the system.

A setup time, at a machine, maybe required every time a new lot comes at the machine or sometimes a setup maybe avoided if the subsequent lot is of the same type as the lot being processed at the machine.

Some machines can process more than one lot at a time. There are a few machines, which can process different types of lots at the same time. This depends on the lot types.
A description of each of the stations is provided below:-

Inspection Microscopes

Microscopes are used throughout the clean room for visual inspection of wafers for defects and contamination. They are generally referred as “scopes”. The inspection is

done manually by the staff one wafer at a time. Some operations require inspection on all the wafers while others require only a defined quantity to satisfy the inspection requirements.

Computer Terminals

Computer terminals are used for manufacturing systems control. The Manufacturing Execution System is accessed from these terminals. Also, the necessary data and information about the company and manufacturing operations are reached from these terminals situated in the clean room. The in and out moves of the lots are done at these terminals. Operators use the terminals to enter necessary information about the lots before and after a certain operation.

Implant

There are two implant machines. Some operations need specifically one of the machines, while others do not need a specific machine. Some operations require more time on one machine than the others do. For some operations Implant 400 is faster, while for some operations Implant 200 is faster. An operator is required during setup but the not during processing phase.

Implant 400

Implant 400 is used in the implant operation. It occupies almost an entire room. Wafers are placed in a Implant 400's fixtures and put in a specific cassette. The entire lot is put together in the machine after the desired set up for the specific operation is done. The machine, then, processes each wafer one by one.

Implant 200

Implant 200 requires a lower setup time since it does not require any special fixture for wafers. It uses regular cassettes in its operations. The robotic arm picks up a wafer from the in-cassette, puts it in the process, brings the wafer out after operation, and places it in out-cassette. Then does the same for the rest of the wafers in the lot.

Stepper

Stepper I, Stepper II

There are two identical stepper machines used for photolithography operations. At the stepper step of the photolithography operation, a specific mask set is placed in the stepper. Steppers use the regular process cassettes. Wafers are picked up by the stepper one by one to perform the process and put back in the same cassette. An operator is required during setup but not during the processing phase.

Suss

Suss is used in the contact lithography. Wafers are in the same process cassette, which is placed in the in-slot. Suss picks up the wafers one by one, processes them and then places in the same cassette. Suss requires an operator to attend the machine at all times while it is running.

Fusion

Fusion I, Fusion II

Both Fusions use regular cassettes for their processing. The lot goes in the in-cassette slot and wafers are processed one by one. Fusion needs an operator only during its setup.

Mask Cleaner

Mask cleaner cleans the mask after the masks are used. The mask set is put in the machine in their carry-box and the mask cleaner cleans them all together.

Polyamide Coater

Polyamide coating is done on a Solitech machine. There is only one coating track on this machine. Wafers are put on one side of the machine where the in-cassette slot is. The wafers are then picked up one by one to pass from several consecutive steps to reach to end cassette. When a wafer is done with a step it moves to the next step and the one before it moves to its subsequent step, thereby implying that the wafers do not wait to proceed until one lot finishes the whole sequence of steps on the track. An operator is needed only when the setup is done but the operator also needs to monitor the machine while doing any other work.

Adhasive Coater

Same as the polyamide coater.

Developer

There are three developers and they operate similarly. They are also same as Polyamide Coater and Adhasive Coater.

Coat

There are three coaters and they operate similarly. They are also same as Polyamide Coater and Adhasive Coater.

Lift-off Evergreen

One of the lift-off machines is by made by Evergreen. Evergreen requires an in-cassette and an out-cassette. Cassettes used are the regular process cassettes. A robotic arm picks up the wafers one by one and places them on to suction chucks that hold down the wafers firmly. Since there are only four chucks, only four wafers can be processed at a time. This means that it takes two runs to finish an eight-wafer run. An operator is used during setups only.

Lift-off Semitool

The Lift-off machine, made by Semitool, can be used in interchangeably, for certain operations, with Evergreen's lift-off machine. Semitool has a laundry machine like door and the lot is placed in a regular process cassette. The lot is processed all at once. An operator spends time only during the setup of the machine.

Branson Barrel Etch

There are identical two Branson barrel etch machines. The lot is placed in a special quartz cassette. The transfer is always performed automatically with the wafer transfer system, which makes the transfer a part of the setup for Branson. All the wafers are processed at the same time and in the same quartz cassette.

Technics

Technics can hold up to four wafers at a time. An eight-wafer lot will run twice on this machine to process all the wafers. The wafers are placed in Technic's chamber four at a time. Each run requires a separate setup which includes the wafers being placed in the chamber.

Bonder

Bonder is a hairdryer like equipment that is used to heat the adhesive in between the sapphire and the front of the wafer. It is a manual procedure. Each wafer is processed by it self. An operator picks up a wafer that is already mounted to a sapphire.

Rinser & Dryer

Rinser & Dryers are used in several operations and are spread in different sections of the clean room. They accommodate one process cassette at a time. An entire lot of wafers can be rinsed & dried all together. The machine has front load laundry machine like door. The cassette is placed inside the machine from this door. Rinser and Dryer A is in the materials lab it is for 4"wafers. Two identical Rinser and Dryer B's are for 4" wafers and are located in process lab and photolithography lab, respectively. Both the Rinser and Dryer C's are located in process and photolithography, respectively. For all the operations that require these Rinser and Dyer's, the operator is only required during the setup of the machine.

Wetbench

Lots that are in process cassettes are subjected to a series of chemicals at specified times. These chemicals are placed in tanks that are in the wetbenches. Operators are used at all times during the wetbench steps. All wetbenches are used for different operations and they cannot replace each other.

Sputter

There are three sputters, namely, CVC and Perkin Elmer and XM8, and arelocated in the process II lab. CVC is preferred because of its easier usage and less run time. They

have very different setups making their setup times also different. Perkin Elmer tends to be used as an alternative to CVC. An operator spends time only during the setup. XM-8 is another sputter that is used for other operations and is located in the materials lab It is a completely different machine having large setup and operation times. Like the others, it can process an eight-wafer lot at the same time.

Plasma Etch PT-740

There are two plasma etch machines. Each has two different chambers with each chamber holding four wafers at a time. Each chamber is used for a different operation. So, for one operation, the machine has to run twice to finish an eight-wafer lot. An operator is needed only during setups. There are two setups since there are two runs.

Evaporator

There are two evaporators, which are the identical machines, but are used for different operations. They can hold up to twenty wafers at a time, so they can process two lots in the same operation . An operator is used during the setup and also monitors the run periodically.

Deposit PT 2411

PT 2411 is the machine used for deposit operations. This machine can accommodate all the wafers of an eight-wafer lot.. All the wafers are processed at the same time in the same chamber. An operator is needed during the setup and also monitors the machine frequently during its run.

Omega Ion Etch

There are two identical omega ion etch machines. The in-cassette is placed in the machine where each wafer is picked up by a robotic arm and processed one by one. Before each production run, a test run is performed on three wafers. The wafers are then tested and inspected. If they conform to the required specifications, the production lot is put in to process in the Omega Ion Etch. An operator sets the machine up and lets it run by itself.

Alpha Step

Alpha Step is a thickness measurement equipment. Operators use Alpha Step to measure thickness of the layers, one by one. Some operations require only a test wafer, which is processed with the lot, to be measured, while others require the entire lot to be measured. An operator manually feeds the wafers and performs the necessary measurements.

OSI

OSI is another inspection machine. An operator checks certain characteristics of the circuit design. Work instructions specify the quantity of wafers that should be inspected. Wafers are inspected one by one by the operator.

Thickness Elipsometer

There are two thickness measurement machines that are identical but perform different operations. Each requires wafers to be measured one by one. The work instruction for each operation specifies the quantity of wafers to be measured. An operator is required during the entire processing of the wafers.

Sheet Resistance 4-Point Probe

Usually, a test wafer, which is processed with the production lot, is used to get data from the 4-Point Probe. The machine processes a wafer at a time. An operator is used during the entire processing of the wafers.

Wafer Transfer MGI

Wafer transfer is used to transfer all the wafers from one cassette to another, automatically. This eliminates some breakage problems with cassette to cassette manual wafer transfer. According to the specifications of the transfer, wafers can be transferred one by one or all at the same time. An operator is used during the entire run.

M-Gage

M gage inspects the wafers one by one. Operators manually setup each wafer and let it run. All the wafers need not be inspected depending on the work instructions. An operator is used during the entire run.

Anneal

Annealing is done with a RTA machine. There is an in-cassette and an out-cassette slot. A robotic arm picks up the wafer and inserts it into the machine. The processed wafer is then placed in the out-cassette. Wafers are processed individually. An operator is used only during the setup time.

Alloy RTA

Alloy RTA operates exactly identical to the anneal machine, explained above.

Priming Oven

A Priming oven looks similar to a regular kitchen oven. It is located in the photolithography area. It can hold up to four lots at one time. There are two priming ovens. An operator is needed only during the setup time.

Polyamide Bake Oven

This Oven operates identical to the Priming oven

Adhesion Promoter Oven

Adhesion promoter operates identical to the Polyamide Bake Oven.

Laser Scribe Wafer Mark II

Wafers are laser scribed their unique mask set number. Wafers are put in the in cassette slot and they are laser scribed one by one and returned to the same cassette. An operator is needed only during the setup time.

Scribe & Break

The wafers are broken into individual “dies” with these machines. One wafer can be processed at a time. An operator is needed only during the setup time.

Mounting

This machine can hold up to three wafers at a time. There are three separate wafer size chambers that the wafers are placed in, with sapphire stuck to their front. It takes three runs to finish an eight wafer lot(1st run-three wafers, 2nd run- three wafers, 3rd run-two process wafers + one dummy wafer). An operator is needed during a setup before each run.

Grinder

Wafers that are bonded to sapphires on their front, are ground down to a specific thickness with this machine. It can hold up to four wafers at a time. Four wafers are laced on four suction chucks. An eight-wafer lot requires two runs. An operator is needed only during the setup time.

Demount S&K

Sapphires that are mounted to the front of the wafers are removed with S&K demount. All the wafers are placed on special fixture that holds the wafers in an angled position and then dipped into cyclohexane tank in the S&K demount station. All wafers are processed at the same time. An operator is needed during the setup time only.

1.7.5 Preliminaries

Before the lot selection rules at M/A-COM’s facility in Roanoke, Virginia, are described, it is necessary to understand the a few important details about the working of the facility.

The facility manufactures a variety of products. However, due to similarities in their processing, the products can be classified into six main types, namely, p1 to p6. Each

product follows a predetermined route, which consists of the various operations that it undergoes, along with the machine at which the operation is to be done.

Each operation is called a phase. Each phase is numbered by a four digit number (eg. 1000, 1010, 2400 etc.). Each phase consists of various sub-phases a,b.c,d…… The number of sub-phases within each phase depends on the phase number and the type of product.

The facility is divided into 5 different areas, namely, Photolithography, Process I, Process II, Materials and Implant. Typically each phase is done within one area and on the completion of the phase the lot moves to another area where it undergoes further processing. Each time a lot starts a particular phase, its entry time is logged in a computer as its in-time and each time a lot finishes a particular phase, its finish time is logged in a computer as its out-time. Thus the time required for each phase is recorded.

For example, for product 5, phase 1531 undergoes the following sub-phases in the Process I area:

a. Computer terminal in-time entry
b. Branson
c. Rinser and Dryer B
d. Inspection Microscopes
e. Computer terminal out-time entry

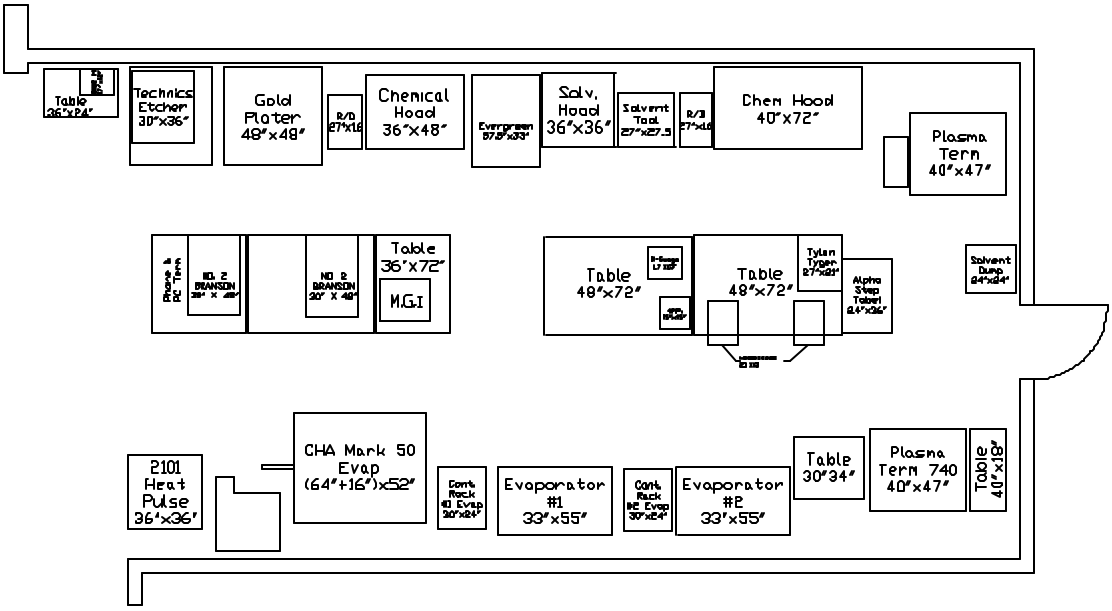


Figure 1.2: Process I area in M/A-COM

The diagram of the Process I area is shown in Figure 1.2. Though there are many different machines in the Process I area, phase 1531 uses only three of them. Other phases, which are required to be processed in the Process I area, may use different machines. Presently, the lots are only sequenced between phases and not at each sub-phase. At each sub-phase the lots flow in the order of First In, First Out (FIFO).

1.7.6 Description of current lot processing rules

Next, we describe the procedure by which the lots are processed in the fabrication area. The performance of this procedure will be compared with the proposed heuristic and other procedures from literature.

All the lots that need to be processed are entered in the startup queue. The jobs from the startup queue are subsequently entered into the fabrication line for processing. A lot is entered into the line as soon as one lot has completed its processing from the line. Hence, the number of lots at any time in the line is a constant, which is a predefined maximum. This number is decided using past data and experience.

1.7.7 Pull Report

The Pull Report gives the specification of the lots in queue at the Pull Area, at any given point in time. It does not give the specification of the job being processed. The Pull Report is divided into a number of columns, which are: -

Lot: This column specifies the number of the lot in the queue.

Use: This column has a * symbol in it if the particular lot in the queue is ready to be worked on. If the lot is not ready to be worked on, the column is left blank. A lot in a queue may not be ready to be worked on if the subsequent Pull Area, to which it is scheduled to go to, has reached its limit in terms of the number of lots that it can accommodate.

Phase: This column gives the operation number of the lot in queue. Different lots in the same queue may have different operation numbers due to the reentrant nature of the system.

Hold: A lot may be ready but could be on hold for different reasons like:-

Machine: There is a breakdown on the machine.

Engineer: The engineer has questions on the lot being processed.

Reticle: The lot needs a new mask or reticle.

Program Management: The lot may be on hold due to some executive decisions that need to be taken.

Process down: The process at which the lot is present, is down.

No technician: A technician is not present to process the lot.

Rework: The lot needs rework.

1.7.8 Sequencing Polices

The lots are grouped into three types, namely, Superhot, Hot and regular lots.

Superhot lots are the ones that have queue time = 0. As soon as an operator starts processing a Superhot lot, he notifies the subsequent area that a Superhot lot is to be started processing after the processing time at the present operation. This ensures that the equipment is ready for a Superhot lot at all times. Hence, the cycle time of a Superhot lot is equal to the sum of its processing times.

Hot lots are the ones that have priority over the other lots (except Superhot lots). However, unlike a Superhot lot, a Hot lot may have some queue time. This queue time would however be kept at a minimum.

If a lot is not a Superhot or a Hot lot, the following rules are used, in the given sequence, to determine which lot is to be processed next:

1. The expected completion time of the lot is calculated using the average cycle time and the average queue time from that process onwards. The due date of the job is known. The lot having maximum (Expected Completion Time – Due date) value is processed next.
2. If lots have the same (Expected Completion Time – Due date) value, the lot which has spent the longest time in the line is processed next.

If the time spent in the line is the same, the lots are sequenced according to the process numbers.

Chapter 2: Literature Review

2.1 Introduction

The interest in semiconductor wafer manufacturing planning and scheduling arose in the late 1980's. A semiconductor manufacturing facility differs from the common job – shop and flow shop type facility due to the presence of reentrant flows at different locations. In wafer manufacturing the number of different processors may be assumed to be deterministic or stochastic depending on the processor and the desired accuracy. The batch size and the machine utilization rate also affect the processing time. The set-up times of operations such as photolithography, test and burn-in, are commonly modeled as sequence dependent. Moreover, some successive operations must be processed within a set amount of time to ensure satisfactory yield. The yield is inversely proportional to the time spent by the lot in the system. The introduction of stochastic events such as machine breakdown and yield increase the accuracy of the model at the expense of its tractability.

All of these important features of a semiconductor facility have led to the development of numerous models and techniques for the solution of scheduling problems occurring in this environment. In 1992 and 1994, an extensive literature review was conducted by Uzsoy, Lee and Martin-Vegas [1]. They defined three types of planning and scheduling problems:

- Performance evaluation: Descriptive models employed to understand the system behavior.
- Production planning: Aggregate, long term production planning.
- Shop Floor Control: Short term control for the processing of orders.

This review was based on this classification of the paper related to the planning and scheduling issues in semiconductor manufacturing. The present review will focus on the type of approaches used for the planning and scheduling problems in semiconductor manufacturing. The approach is classified as follows:

- Dispatching Rules
- Linear and Non-Linear Programming
- Simulation Model

2.2 Dispatching Rules

The dispatching rules (DR) are the most common used tools to schedule the wafer manufacturing process. They are applied to select which job to process next on a particular machine. The use of the DR is often motivated by the fact that they are fast and simple to implement. Moreover, they are also easy to understand and so the managers and operators tend to like them.

The most common dispatching rules include first-in first-out (FIFO), shortest processing time (SPT), longest processing time (LPT) and due date based rules (earliest due date, least slack e.g.) among others.

Lu, Ramaswamy and Kumar [3] and [4] introduced deterministic rules based on the Least Slack (LS) policy, in order to smooth out the mean and variance of cycle times. The proposed rules are as follows.

LS policy : $\text{Slack}_k = (\text{Set Due Date})_k - \text{Estimated Remaining Processing Time})_k$

The highest priority is given to the lot k with the least slack.

(This rule attempts to make every lot equally early or equally late.)

FSVCT policy : $S_k = (\text{Release time of lot})_k - (\text{Estimated Remaining Processing Time})_k$

The highest priority is given to the lot k with the least value of S.

(This rule attempts to reduce the variance of cycle time.)

FSMCT policy : $S_k = (\text{Est. Remaining Proc Time})_k - (\text{Estimated time to Reach } k+1)_k$

The highest priority is given to the lot k with the least value of S.

(This rule attempts to reduce the variance of the interarrival time at each buffer.)

Using simulation the authors compared the results of the above rules to those obtained using numerous other rules (such as FIFO,SRPT, EDD, etc). It appears that the "fluctuation smoothing of variance of the cycle time (FSVCT)" policy efficiently reduces the variance of the cycle time. The fluctuation smoothing policy for mean cycle time (FSMCT) policy decreased the mean cycle time and tend to reduced its variance by avoiding bursting in the arrivals of buffers.

The authors claim to reduce the cycle time by more than 20% and its variance by more than 40% compared to FIFO. However, the effects of these rules on other performance measures such as WIP and machine utilization are not presented.

Hung and Yang [5] address the due date related criteria. They consider DRs like Expected Due Date (EDD), Shortest Remaining Processing Time (SRPT), Urgent Factor Index (UF), as well as other rules with the aim of balancing the line. These DRs are applied on a model including a few machines and assuming deterministic processing time. The efficiency of each DR is presented under different number of machines. As expected , the results vary, depending on the model.

Kayton, Teyner, Schwartz, and Uzsoy [6] compare many common DR in a complete simulation study. The interest of their work dwells on the inclusion of processor downtime in the model. This particularity enhances the result validity. The authors outline the significant effect of non-bottleneck downtimes on the mean cycle time.

The most complete study of the scheduling problem seems to be the work of Wein. He reviews 12 different DRs, simulates their applications on two existing fabs data under six common distributions. The extensive amount of result is clearly summarized to enable comparisons. It appears that no rules dominate in performance but the author observes that if the input distribution is know, some DR can be disregarded since they cannot perform better than others. The main interest of this paper lays in the great diversity of rule and distribution applied to realistic set of data.

Related to DR efficiency, the work of Kumar and Kumar [7] addresses the performance bounds of Markovian queuing network and DR. The DR's considered in the article are the First Buffer First Serve (FBFS), Last Buffer First Serve (LBFS). Those DRs are specific to the reentrant flow. The authors analyzed the behavior of these policies under traffic conditions varying from light to very heavy. They observed that the efficiency of the FBFS and LBFS policies could be improved by the addition of constraints on buffer capacity. The authors consider both open and closed queuing network and establish performance bounds for different simple models. The computation of bound for such rules provides important information about DR limitations and enables one to select easily the most efficient policy for a particular system under a pre-defined traffic condition.

2.3 Linear and Non-Linear Programming Models

eHung and Leachman [2] present a linear programming based planner and scheduler. The results given by the system are used as an input to a simulation model that validates the result. When the LP and simulation systems do not agree, the LP is reformulated until satisfactory agreement between the two models in obtained. The LP minimizes a weighted production cost and includes capacity, demand and processor availability. The use of two models ensures reliable results under varying conditions.

Glasssey, Shantikumar and Seshadri [8] address the job –release problem for a single product, high volume semiconductor fab. The objective is to minimize the cycle time. The linear control rules are based on intersecting hyperplanes and determine the time of release. These rules can be applied to the fab model developed by Wein. Lou and Kager [9] worked along the same direction. They minimize the WIP while maintaining the output. The validation of their model is achieved by comparing its results to two other simulation based schedulers.

Connors and Yao [10] paid attention to the total demand for a multi-product fab. The authors define numerous technological constraints and assume a random yield. The optimization of the production is achieved with the help of six linear programs. The main contribution of this work is the integration of constraints aiming at reaching the expected demand set. The problem to meet production targets in an IBM facility motivated the design of this system.

Bai, Srivastsan and Gershwin [11] decompose the scheduling problem into a hierarchical structure to allow for the integration of non-linear and linear programming. The program applies non-linear programming data to establish long run values such as the set-up rate. At a lower level, the system executes linear programming computations to find the production rate etc. The lower level is controlled by different DRs that depend on the results previously obtained.

Mehta and Uzsoy [12] deal with the scheduling of several incompatible product families. The minimization of total tardiness appears to be NP-hard. Dynamic Programming (DP) optimally solves small size systems but larger problems cannot be treated within reasonable amount of time. By applying a decomposition algorithm to the large size

problem, they divide it into smaller ones that can be solved using DP. The results of the heuristic are obtained quickly and are appear to be robust and near optimal.

2.4 Simulation Model

A simulation model can take into account numerous parameters like upstream and downstream state of each machine, identifying bottlenecks to find good schedules quickly. Thompson [13] describes the integration of simulation techniques in the semiconductor industry and presents the perspective given by the fast pace of development of the simulation program. The simulation tools presented in his paper aim at solving two of the main wafer fabrication problems: when to release lots into production and what each piece of equipment should work on next. The simulation system may be used in three different modes: offline policy development, predictive planning and reactive scheduling.

El Adl, Rodriguez and Tsakalis [14] focus their study on the hierarchical aggregation/disaggreation model of the facility. This decomposition yields smaller models that can be scheduled through discrete event simulation within a shorter amount of time. This strongly enhances the performance of the scheduler since excessive computation time is the main disadvantage of this type of system.

Pickett and Zuniga [15] present a simulation system implemented in a Japanese facility. The choice of discrete event simulation was motivated by the inability of analytic technique to integrate simultaneously divers parameters. The dispatching accuracy of the system is claimed to be 95%. It also reduces the involvement of the operator and can perform sensitivity analysis.

Chapter 3: Proposed Methodologies

3.1 Introduction

In this chapter three methodologies are presented to minimize the cycle time at M/A-COM's wafer fabrication environment. The chapter is divided into three sections:

In the first section the development of a heuristic algorithm based on a new dispatching rule is presented. This section details an the development of an algorithm, which is subsequently analyzed using a simulation software.

The second section details the development of a mathematical based program to model the wafer fabrication system. An integer programming model is developed and analyzed. Various modifications are suggested for improved performance.

The third and last section presents an approach to minimize the maximum cycle time using a conjunctive disjunctive graph based approach.

3.2 Development of a heuristic algorithm based on a new Dispatching Rule

3.2.1 Introduction

In this chapter, the formulation and analysis of the algorithm used to minimize average cycle time, are presented. The analysis of the algorithm is done using the AutoSched AP software by Auto-Simulations Inc. This software is a C++ object oriented based application. This allows the user to use various features within the simulation software to formulate new scheduling rules and test them against the rules already in the software. Using this feature of the software, a new scheduling rule is formulated and its performance is tested against various other rules.

The use of effective scheduling rules is of paramount importance in any manufacturing industry. Deciding which product to be processed next is a classical type of sequencing problem. Scheduling rules have widespread applications in various industries due to their ease of use. The type of scheduling rule to be used, however, varies according to the manufacturing environment and the objective that is to be optimized.

The concept of minimization of idle time at bottleneck was introduced by Sarin and Kalir [16]. They propose that if a schedule minimizes the idle time at the bottleneck, then that schedule must be a near optimal schedule. They devised an algorithm, for a flow

shop, on the basis of this concept and compared it to various other scheduling rules. The analysis showed that the algorithm performed better in terms of minimizing mean cycle time and standard deviation. Though the idea of minimization of bottleneck idleness was applied to a flow shop situation, it can be extended to take into account the re-entrant nature of the semiconductor wafer fabrication. We develop such an algorithm for the problem on hand.

3.2.2 Motivation for the New Dispatching Rule

Sarin and Kalir [16] have proposed the idea of minimization of idle time at the bottleneck machine. They state then "When sufficiently small lot sizes are utilized, minimizing the idle time on the bottleneck machine must necessarily result in a near optimal schedule."

Using this idea a Bottleneck Minimal Idleness (BMI) heuristic has been presented which endeavors to minimize the idle time at the bottleneck. This is done by sequencing the lots in decreasing order of closeness of their secondary bottleneck machine to the primary bottleneck machine. 'Secondary bottleneck machine' means the upstream machine with the next largest unit processing time after the bottleneck machine (which is referred to as the primary bottleneck machine). By utilizing this approach, some of the idle time that might have been created on the machines closer to the bottleneck machine is absorbed, because it overlaps with the processing of the previous lots.

All the analysis done for this heuristic is applied to the flow shop environment. For further details of the algorithm and results refer to the paper [16].

The idea of minimizing the idle time at the bottleneck machine can be extended to the reentrant nature of a wafer fabrication environment.

Consider an example of four lots, two each, of part type A and B. The start times for the two lots of part type A are {0, 6} and the start times for the two lots of part type B are {0,6}. The route and the processing times of part types A and B are given below:

Route for Part A	**Processing times for part A**	**Route for part B**	**Processing times for part B**
Machine 1	3	Machine 1	3
Machine 2	12	Machine 3	4
Machine 1	4		
Machine 3	3		

Table 3.1: Processing Time data for example 1.

For this example, it can be seen that machine 2 is the bottleneck machine. Hence, the proposed algorithm will try to minimize the idleness at machine 2. Hence, the schedule is as follows:

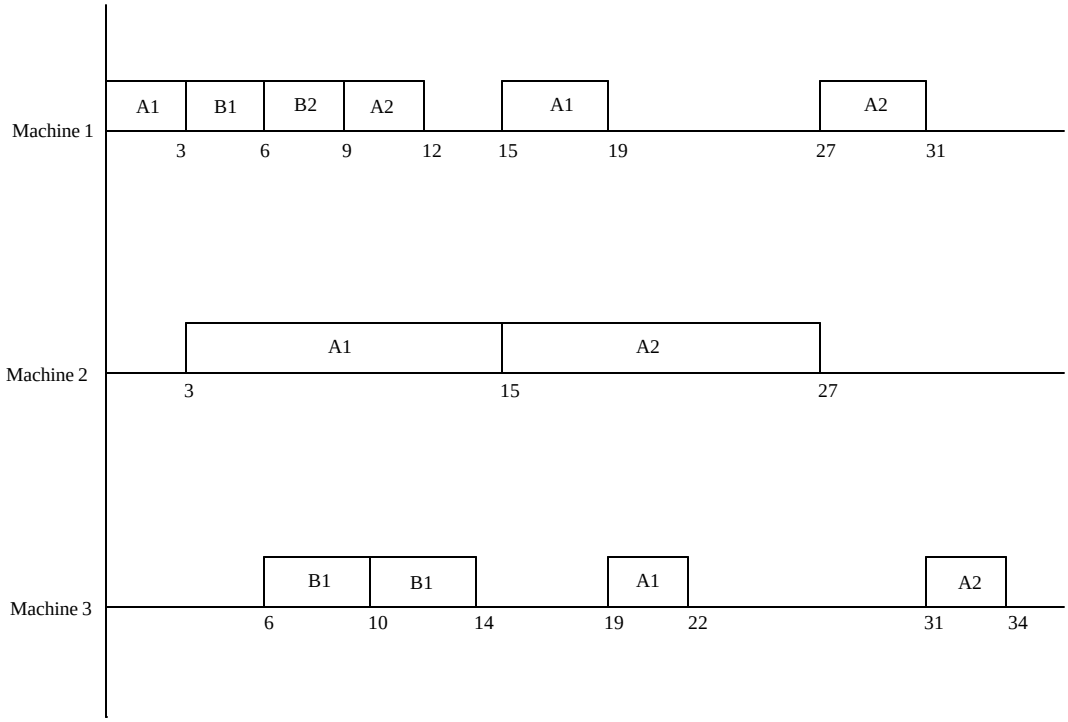


Figure 3.1: Gantt Chart for experiment 1.

The completion times of the lots are as follows:

Lot Type	Completion Time	Ready Time	Cycle Time
A1	22	0	22
B1	10	0	10
A2	34	6	28
B2	14	6	8

Table 3.2: Completion time table for experiment 1.

Mean Cycle Time = (22+ 10+ 28+ 8)/4 = 17 units

Hence, the mean cycle time for the lots is 17 units. Also, notice that at time 6, when lots A2 and B2 enter the system, lot B2 is chosen over lot A2 since it does not cause any idleness at the bottleneck machine.

If at time 0, at machine 1, lot B1 were to be chosen before lot A1, the schedule would have been :

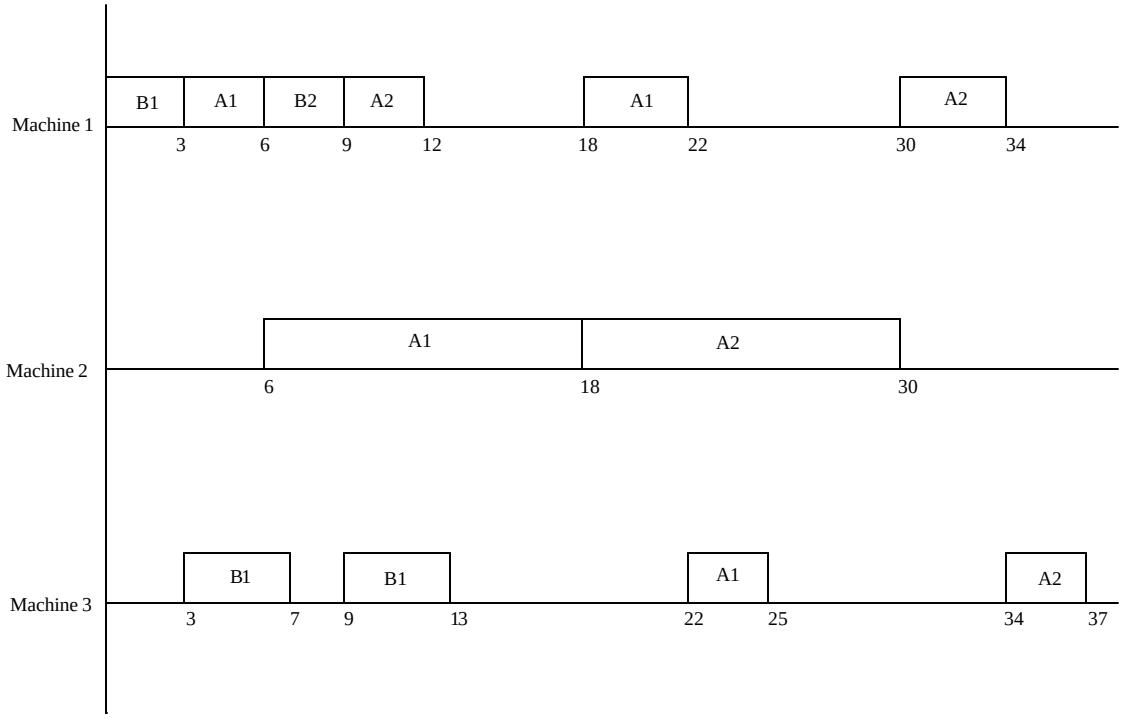


Figure 3.2: Gantt Chart for experiment 2.

with the completion times:

Lot Type	Completion Time	Ready Time	Cycle Time
A1	25	0	25
B1	7	0	7
A2	37	6	31
B2	13	6	7

Table 3.3 Completion time table for experiment 2

Mean Cycle Time = (25+7+31+7)/4 = 17.5 units

The mean cycle time for the lots is 17.5 units.

Hence a delay in the start of the bottleneck machine results in an increase in the average cycle time of the lots.

Also, in schedule 1, if at time 6, lot A2 were to be chosen over lot B2, the corresponding schedule would be:

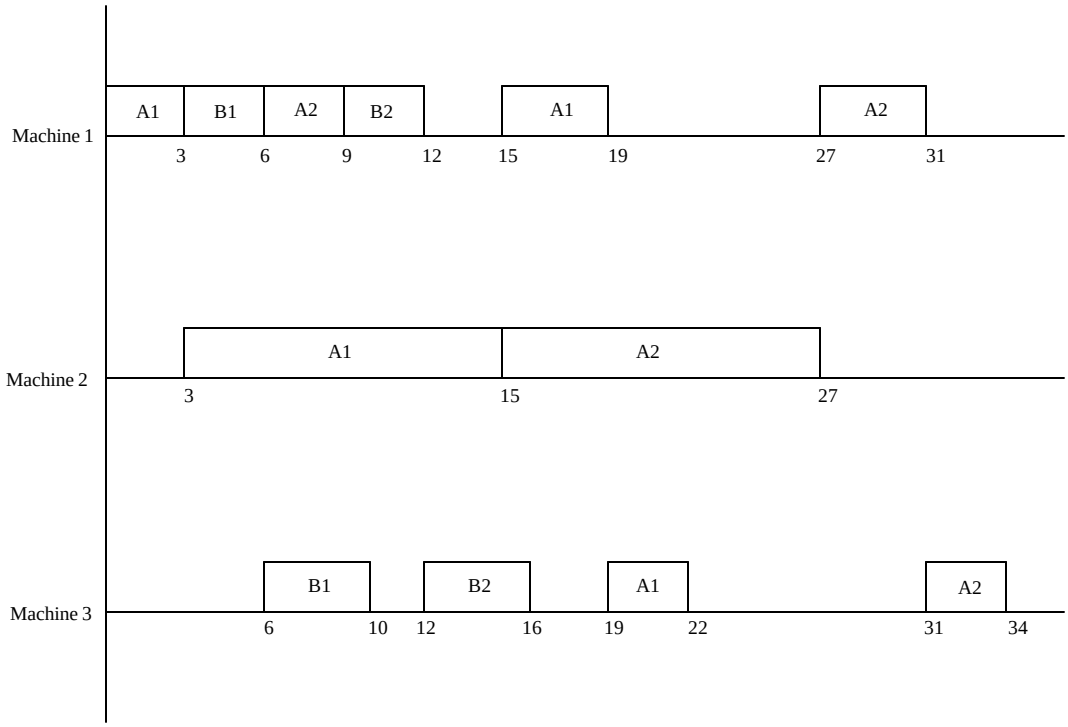


Figure 3.3: Gantt Chart for experiment 3.

with the completion times:

Lot Type	Completion Time	Ready Time	Cycle Time
A1	22	0	22
B1	10	0	10
A2	34	6	34
B2	16	6	10

Table 3.4: Completion time for experiment 3.

Mean cycle time = (22+10+34+10)/4 = 17.5 units.

Here lot B2 was delayed unnecessarily in favor of lot A2. Inspite of processing lot A2 first, there is no improvement in its cycle time since it is waiting for the bottleneck machine to complete its processing. Hence, when looking at the idleness of the bottleneck machine it is also necessary to check if any lots, which have lesser processing time remaining, can be processed without causing any delay in the system.

Therefore, while it is necessary not to cause idleness at the bottleneck machine, if the lot with lesser percent of processing time remaining, can be processed without causing any idleness on the bottleneck machine, it should be done first.

3.2.3 Development of Necessary Concepts

In this section details of the various concepts applied in the proposed algorithm are presented.

Determination of the Bottleneck

The bottleneck machine is defined as the machine at which has the cumulative time of all the lots in the system, at that instant is the largest. Let:

M - number of machines in the fabrication area m = 1,……,M

L_i - identification of lot i

Q - set of all the lots in the system Q = {L_1,L_2,…….L_N}

p_{im} - the processing time of lot i on machine m.

BN - The bottleneck machine; BN = $\arg \max_{1 \le m \le M} \{ \sum_{L_i \in Q} L_i . p_{im} \}$

Dynamic nature of the system

The system that is modeled is a dynamic system i.e. lots enter and leave the system at regular intervals of time. This could cause the bottleneck machine to change from one machine to the other, since the bottleneck machine is a function of the lots in the system at that instant. This necessitates calculating the bottleneck regularly. Hence to take into account the dynamic nature of the system, the bottleneck machine is calculated each time a decision needs to made at the machine. This takes into account the various changes that the system may have undergone in between two decisions at a particular machine.

Batching Machines

Some machines are capable of processing more than one lot at a time. These machines are referred to as batching machines. The batching machines have a specified minimum batch size, as well as, a specified maximum batch size, which they can process at one time. Each time a batching machine selects lots from its queue to be batched, it checks if the minimum batch size has been attained. If the minimum batch size has not been attained, then it will wait for more lots to enter its queue.

Machine Downtimes

The machines are subject to failures and calibrations. Both these events are defined as downtime for a machine.

Least Percent Processing Time Remaining

Each lot in the station's queue is initially sorted in the order of least percent processing time remaining first. This is calculated as follows:

$$\text{percent processing remaining} = \frac{\text{remaining processing time}}{\text{actual processing time} + \text{remaining processing time}}$$

The actual processing time is the processing time the lot has incurred up to the current step. As seen in the previous examples in this chapter, by unnecessarily delaying lots having least percent processing time remaining, the average cycle time of the system

increases. Hence this rule is used to initially to give preference to lots which can leave the system quickly.

3.2.4 Proposed Algorithm

On the basis of the minimization of idle time on the bottleneck and checking the least percent processing time remaining a new algorithm is proposed which endeavors to minimize the idle time at the bottleneck and hence reduce the average cycle time of the lots in the system. The algorithm is as follows:

Step 1. The lots in the present station's queue are sorted on the basis of least percent processing time remaining. If $L(\theta_1)$ is the percent processing time remaining of lot θ_1, then the lots are arranged in the order of $\theta_1, \theta_2, \theta_3, \theta_4, \ldots\ldots\ldots$ where $L(\theta_1) <= L(\theta_2) <= L(\theta_3) <= L(\theta_4) \ldots..$

Step 2. The bottleneck of the entire system is calculated based on the lots, which are in the system.

Step 3. After calculating the bottleneck of the system, the lots in the present station's queue are divided into three groups

Group 1: The lots visiting the bottleneck during the remainder of their route. While looking at the stations on the remainder of a lot's route, the status of the station is also determined. If the station is down or will be down when the lot reaches it, the lot is **not** placed in group 1. Also the lot with the least time to the bottleneck is identified.

If B is the set of lots visiting the bottleneck during the remainder of their route and the time to the bottleneck for lot θ_1 is $t_{\theta 1}$ then lot θ_B is the lot with the least time to the bottleneck.

$$\theta_B = \min_{q_i \in B} \{tq_i\}$$

Group 2: The lots not visiting the bottleneck during the remainder of their route and the stations along the remainder of the route are not down or will not be down when the lot reaches it.

Group 3: All the remaining lots.

Step 4. After dividing the lots into groups, the status of the bottleneck machine is determined. If the bottleneck machine is idle and if group 1 is not empty, then lot θ_B, is scheduled.

Step 5. If group 1 is not empty and if the bottleneck machine is not idle, the lots in group 2 are checked to see if by scheduling any one of them now will it subsequently cause any idleness on the bottleneck machine. This is done by adding the processing time of the lot on the present machine and the time taken by the lot θ_B. Ties are broken using the least percent processing time remaining.

Step 6. If no lots from group 2 are scheduled and group 1 is not empty, the lots of group 1 are checked to see if by scheduling any of them now, will there be any idleness caused on the bottleneck. If a lot does not cause any idleness on the bottleneck, it is scheduled next. Ties are broken using the least percent processing time remaining. If all lots cause idleness on the bottleneck, then lot θ_B is scheduled next.

Step 7. If group 1 is empty and group 2 is not empty, the lot with the least percent processing time remaining is scheduled among lots in group 2.

Step 8. If group 1 and group 2 are both empty, the lots in group 3 are now divided into two sub groups.

Sub Group a: The lots visiting the bottleneck along the remainder of their route.

Sub Group b: The lots not visiting the bottleneck along the remainder of their route.

The difference between these groups and groups 1 and 2 is that the status of the stations along their route are not considered.

Step 9. The same rules are applied as above by replacing **group 1** with **sub group a** and **group 2** with **sub group b**.

A multiple bottleneck version of the above algorithm has also been devised. It follows the same logic as above, except that it identifies two bottlenecks (primary and secondary) and tries to minimize the idleness on both the bottlenecks (giving preference to the primary

bottleneck first). This is useful when there is no single dominant bottleneck in the system, but two machines, which are the bottlenecks.

3.2.5 Description of the simulation model

The simulation model that is described next was developed using the AutoSched AP software by AutoSimulations inc. It is a C++ object oriented based application. The inputs are presented in the simulation model in a tabular format, using a Microsoft Excel interface. There are different worksheets, which are used to define the necessary manufacturing system components in the simulation model. These are:

- The Orders Worksheet
- The Stations Worksheet
- The Parts Worksheet
- The Routes Worksheet
- The Down Calendar Worksheet
- The Preventive Maintenance Calendar Worksheet

Additional optional worksheets are used to define non-standard scheduling rules, the format and the statistics collected and presented in the output reports and operator related input, such as operator certifications required to support certain machines, the periods of availability and unavailability of operators etc.

Next, the type of information specified in each of the input worksheets is described. Each of the worksheets can be found in the Appendix. For more detail refer to the AutoSched AP software user manuals.

The Orders Worksheet

This worksheet contains information on the orders, such as the product types listed on each of the orders; the starting date, or the date of arrival of the order, in the system; the due date of the order; the mean time between arrivals (as a constant or a distribution) of identical orders and the number of times these orders repeat throughout the simulation.

The Orders Worksheet is attached in Appendix A. The order name of the first order is Order 1, which contains a single lot L1. The product type is p5, the start date of the order is 3/20/2000, 8:00am and the due date is 3/31/2000, 12:00am.

The Stations Worksheet

This worksheet contains information on the workstations that perform the processing of the orders. It contains the names of the station families and the names of the stations, or machines, within each station family. A station family may consist of several machines of different capabilities. The quantity of machines of each type and the scheduling rules to be used are also specified in this worksheet.

If the station is a batching station, there are additional fields that need to be specified. The minimum batch size and the maximum batch size are specified along with the "wait no longer that time" field. The "wait no longer than time" field denotes the minimum time that a station would wait after a minimum batch size has been attained. Also, the criteria required for batching various lots can be specified. Hence, lots, which require the same setup, or lots containing the same part type etc., can be formed as a batch.

The stations worksheet is attached in Appendix B. Each station family is named and the stations within the station family are identified. In this case, station family tablehptest contains a single station, namely, tablehptest_1. The ranking function that this station uses is rank_LBA (least balance ahead) and the rule function used is rule_FIRST. The rank and the rule functions are used in conjunction with each other to determine which lot the station should process next. The working of rule and rank functions can be described as follows:

When a station becomes available for processing, all the lots in its queue are sorted according to the rank function. After this is done, the rule function is applied to determine which lot will be processed next.

Each of these rule and rank functions are C++ written functions, which the software calls everytime a decision is to be made at the station. The proposed algorithm is such a C++ function, namely, rule_bottleneck, which has been interfaced with the AutoSched AP software. For the multiple bottleneck case the function name is rule_multbottleneck.

For both these custom rules no ranking function is required as the ranking of lots is done within the rule itself.

The Parts Worksheet

This worksheet contains the names of the products, the route file in which the route of the product is stored and the route name. It serves as a link between product types and their respective route file(s).

The parts worksheet is attached in Appendix C. For the first part p1, the route file is specified to be route1.txt and the name of the route is route-p1. The standard lot size for the part is specified to be 8.

The Routes Worksheet

The routes worksheet includes information pertaining to the routings of the different product types. It lists the product names, the identifying numbers for the steps of each; the station family at which each step is performed; the processing times and their respective distributions; the entity for, which the processing time applies (eg. lot, piece) and the setup time required prior to the processing of each lot. Also, the operator required at a particular step and the operator certification required is also specified. Each step may have a particular yield ratio. This can be specified under the yield column in the routes worksheet.

The route worksheet for parts p1 to p6 are attached in Appendices D to I. For part p1 in Appendix D, the first phase in the route is r1000. The STNFAM field specifies the station family required and the processing time for the lot at the first step of phase 1000 is specified. The field STEP_ACTLIST specifies if the operators are needed only during setup, load and unload or during the entire processing time. The default is for the operator to spend the entire processing time at the station.

The Down Calendar Worksheet

The machines or stations are subjected to failure at various intervals of time. This can be simulated using the Down Calendar Worksheet. A down calendar can be attached to each station in which the mean time between failures (MTBF) and the mean time to repair (MTTR) times are specified.

The Down Calendar Worksheet is attached in Appendix J. For down calendar impfour_down the calendar type is mttf_by_cal, which denotes that the mean time to failure is to be calculated using calendar time.

The Preventive Maintenance Calendar Worksheet

The machines are periodically subject to preventive maintenance. This is simulated using the Preventive Maintenance Calendar Worksheet. A preventive maintenance calendar is attached to each station in which the mean time between preventive maintenance (MTBPM) and the mean time to repair (MTTR) times are specified.

The Preventive Maintenance Calendar Worksheet is attached in Appendix K. For preventive maintenance calendar impfour_30 the calendar type is mtbpm_by_proctime, which denotes that the mean time between preventive maintenance is to be calculated using the processing time of the station.

3.3 Mathematical Programming Based Approach

3.3.1 Introduction

Linear and integer programming based models are widely used to analyze manufacturing systems. The advantage of these mathematical models is that they can be changed easily to take into account any changes in the manufacturing system. Also, since they almost always provide optimal solutions, their use can considerably increase the efficiency of the system. To this end, an integer programming model of the

manufacturing system at M/A-COM is developed and a procedure for efficiently solving the model is presented.

The integer program has been formulated considering binary variables to be the starting time of the operation of a particular job on a particular machine. Using these binary variables the necessary constraints are formulated. The CPLEX software is used to solve this integer program. CPLEX is a non graphic based linear and integer programming software which can read text files containing constraints, objective function and variables and uses them as input for the integer program. Hence, a C++ program is written which can read the files containing route information, such as processing time, and machine information and convert them into the appropriate constraints. Since, integer programs require a lot of computer time to solve and also use up a lot of computer memory, various modifications are proposed, in the latter part of this chapter, which help in reducing the integer program into simpler programs. This would be useful in solving the problem in reasonable time.

3.3.2 Description of the model

As mentioned earlier, the model uses the start times of each operation of a job at a particular machine, as a binary variable. The time horizon T is assumed to be discrete and specified by the user. The variables are defined as follows:

X_{ijkt} = 1, if operation i of job j is processed starting at time t on machine k

= 0, otherwise.

N = Number of jobs.

M = Number of machines

L_j = Last operation of job j. (dummy operation)

p_{ijk} = processing time of operation i of job j on machine k, including setup time.

T = Upper bound on the time horizon.

$K(i,j)$ = set of alternate machines on which operation i of job j can be processed.

$I(k)$ = set of operations that can be worked on machine k.

Objective Function:

Minimize flow time over all the jobs.

$$\min \sum_{j} \sum_{t=1}^{T} \sum_{k \in K(i,j)} t.X_{Ljkt}$$

The constraints are as follows:

Constraints 1 : (Sequential undertaking of the jobs)

An operation i+1 of job j must start after the completion of operation i.

$$\sum_{k \in K(i,j)} \sum_{t=1}^{T} (t + p_{ijk}) X_{ijkt} \leq \sum_{k \in K(i+1,j)} \sum_{t=1}^{T} t.X_{i+1jkt} \qquad \text{for i} = 1,2,3,\ldots, L_j$$

$$\text{j} = 1,2,3,...,\text{N}$$

Constraints 2: (Unique processing of an operation of a job)

Each operation i of a job j must be processed only once.

$$\sum_{t=1}^{T} \sum_{k \in K(i,j)} X_{ijkt} = 1 \qquad \text{for i} = 1,2,3,\ldots, L_j$$

$$\text{j} = 1,2,3,\ldots,\text{N}$$

Constraints 3: (Each station has a capacity of 1)

At any given time, the number of lots at any machine must be less than or equal to 1.

$$\sum_{t=1}^{T} \sum_{I(k)} \mathbf{y_{ijkt}} . X_{ijkt} \leq \mathbf{1} \qquad \text{for k} = 1,\ldots,\text{M}$$

where $\mathbf{y_{ijkt}}$ = a column vector with T elements and consists of zeros except from the t^{th} element to the $(t + p_{ijk} - 1)$ element, where it consists of one's.

$\mathbf{1}$ = is a column vector with T elements consisting of one's.

Constraints 4: (Processing at Batching machines)

At batching machines, a number of jobs can be processed simultaneously. This requirement can be captured by a constraints identical to that in Constraint 3 above, as follows:

$$\sum_{t=1}^{T} \sum_{k \in B} \mathbf{y_{ijkt}} . X_{ijkt} \leq \mathbf{J_k}$$

where $\mathbf{J_k}$ = a column vector with T elements with each element equal to J_k, where J_k is the maximum batch size at the machine.

B = is the set of batching machines.

Constraints 5: (For batching machines, no partial overlap of jobs)

At a batching machine, the subsequent operation must precede the previous operation by atleast the processing time of the previous operation.

$$\sum_{t'=1}^{T} \sum_{t=1}^{T} (t' X_{i'j'k't'} - tX_{ijkt}) = b . p_{ijk} + a . p_{i'j'k'} + K_1 - K_2$$

$\forall k \in$ batching machines

$i, j, i', j' \in I(k)$

where a and b are binary integer variables

K_1 and K_2 are non-negative real variables.

$a + b \leq 1$

$K_1 \leq b . T$

$K_2 \leq a . T$

This completes the formulation.

3.3.3 Analysis

The mathematical formulation was initially run over smaller problems to test its functioning. The experiments are as follows:

Expt. 1:

Six lots of the same part type are in the system at time 0. The system consist of 3 machines. The route of the parts are

Step No.	Machine No.	Processing Time
1	1	2
2	2	2
3	1	1

Table 3.5: Route data for expt.1 of integer programming

The model is run over a time horizon of 25 units. The solution obtained is:

Product	Start time at Machine 1	Start Time at Machine 2	Start Time at Machine 1	Finish Time
1	8	10	12	13
2	2	4	6	7
3	10	12	14	15
4	0	2	4	5
5	6	8	10	11
6	2	6	8	9

Table 3.6: Optimal solution of expt.1 of integer programming

Integer optimal solution: **Objective function = 60.**

Solution time = 31.79 sec. Iterations = 57154 Nodes = 1480

Expt. 2

This time, machine 1 is designated to be a batching machine and can take 8 lots at a time. There are six identical products in the system and their routes are as follows:

Step No.	Machine No.	Processing Time
1	1	2
2	2	1
3	1	2
4	2	1

Table 3.7: Route data for expt.2 of integer programming

The model is run over a time horizon of 40 time units. The solution obtained is as follows:

Product	Start time at Machine 1	Start Time at Machine 2	Start Time at Machine 1	Start Time at Machine 2	Finish Time
1	0	4	5	10	11
2	0	3	5	7	8
3	0	8	9	11	12
4	0	6	7	9	10
5	0	2	3	5	6
6	0	13	14	16	17

Table 3.8: Optimal solution of expt.2 of integer programming

Optimal solution: Objective function : 64

Solution time = 1094.15 sec. Iterations = 745976 Nodes = 21260

In this experiment all the jobs are processed as a batch on machine 1 at time 0. This ensures that there is no idle time in the processing of the next operation at machine 2.

The proposed model captures all the important features of the problem on hand, and once solved optimally will provide the best solution for effective functioning of the fabrication area. However, it is also seen that the model is very cumbersome and takes very long to run even on very small problems. This is due to the integer nature of the problem which is time consuming. Hence, there is a need to improve the model by relaxing the integer constraints and formulating and equivalent linear or non-linear programming model. The dynamic arrivals of jobs needs to be taken into account in the

model. Also, the model needs to be modified to account for scheduled downtimes for the machines. All these modifications are discussed in the next section.

3.3.4 Modifications

Different Ready Times of the jobs: Given the start time of a particular job the first operation of the job cannot begin earlier than the start time. Hence an additional constraint can be added to the model such that if S_j is the start time of job j and X_{1jkt} is the start time of the first operation of job j then

$$Sj \leq \sum_{t=1}^{T} \sum_{k \in K(i,j)} X_{1jkt}$$

Scheduled Downtime of machines: Various machines need to undergo maintenance periodically. This can be accommodated in the model in Constraints 3. When a machine is down from a period t to (t + p), the corresponding 1's in the right hand side vector are changed to 0's. This denotes that no job can be processed at that machine during that time.

3.3.5 Reduction of the Proposed Integer Programming Model

At discussed in the previous chapter, the model is very time consuming even for very small problems. Hence, there is a need to reformulate the problem into an LP and then solve it, since linear programming problems require considerably less time to solve.
One way is to allow X_{ijkt} to take values anywhere between 0 and 1 and check to see the feasibility of the solution obtained. This was done for the following problem.
Six identical products flow through the system. This problem is similar to expt.1.

LP Solution

Product	Finish Time
1	0 (0.633) 30 (0.3667)
2	9
3	0 (0.766) 30 (0.233)
4	0 (0.416) 12 (0.583)
5	0 (0.3636) 11 (0.6363)
6	0 (0.5) 14 (0.5)

Table 3.9: Partial solution for LP relaxation

Primal - Optimal: Objective = 48.

Solution time = 6.708 sec

Hence, even though the solution time was reduced drastically, the solution obtained is not feasible since a part of the product is finished at a particular time and the rest of it is finished later (for product 1, 63% is finished at time 0 and 37% is finished at time 30). Hence, by only the integer variable constraints on X_{ijkt} the linear program cannot be solved. Hence some additional constraints need to be added.

One set of constraints that could be added is the completion of a particular operation of a job completely before the next operation can begin. This can be done as follows:

$$\sum_{t=1}^{T} \sum_{k \in K(i,j)} X_{ijkt} \geq d \quad \text{and}$$

$$X_{i+1jkt} \leq d$$

Even though the variable δ used here is an integer variable, the number of integer variables is much less compared to the initial formulation. The problem is run with these initial constraints. The problem is as follows:

Six identical products, following the same route. Machine 1 is a batching station and can process 8 lots at a time. The route is

Step No.	Machine No.	Processing Time
1	1	2
2	2	1
3	1	2
4	2	1

This problem is similar to expt. 2 in the earlier chapter.

The solution obtained was as follows:

Product	Start time at machine 1	Start time at machine 2	Start time at machine 1	Start time at machine 2	Finish Time
1	0	0 (0.6) 5 (0.4)	6	6 (0.6) 11(0.4)	6 (0.6) 11 (0.4)
2	0	0	6	10	11
3	0	0 (0.4) 1(0.25) 5(0.35)	6	7 (0.75) 11 (0.25)	11
4	0	3	6	9	10
5	0	1 (0.75) 5(0.25)	6	8	11
6	0	4	6	6 (0.4) 7 (0.25) 11 (0.35)	11

Table 3.10: Partial Solution for partial LP program

Hence the number of partial solutions obtained has decreased. However the solution is still not feasible due to the presence of partial processing of jobs at various stations.

3.3.6 Alternate Formulations

To overcome the formation of partial solutions two alternate formulation is proposed which will consist of all continuous variables.

Alternate Formulation 1: (Non-linear constraints)

Non-linear constraints can be added to the model to overcome the infeasibility of partial solutions. This can be done by adding the following constraints,

Constraints 2a:

The product of Xijkt and Xijk't' where $t \neq t'$ must be zero.

Constraints 5a:

The product of a and b must be zero.

The formulation remains the same, as discussed in the first part of the chapter, with one change. The variables Xijkt and a and b are now continuous within the range 0 and 1. This ensures that all the variables do not necessarily have to be integers.

Alternate Formulation 2: (Quadratic Objective Function)

In this formulation again, all the integer variables are relaxed to be continuous between zero and one. However additional terms are added in the objective function to ensure that no partial solutions are obtained.

New Objective Function:

$$\min \sum_{j}\sum_{t=1}^{T} \sum_{k \in K(i,j)} t.X_{Ljjkt} + \sum_{j=1}^{N}\sum_{i=1}^{Lj}\sum_{t=1}^{T} R.X_{ijkt}(1 - X_{ijkt}) + \sum_{a} R.a\,(1-a) + \sum_{b} R.b\,(1-b)$$

Where R is a very large number.

The objective will then force all the Xijkt's and a's and b's to be either 0 or 1 due to the high penalty R associated with it being any other value. CPLEX does solve problems with quadratic objective functions. However the condition is that for a concave objective function the coefficient's matrix should be negative semi-definite. Since this does not hold true for the above objective function, the problem cannot be solved using CPLEX.

Both the abovementioned alternate formulations use non-linearity in either the constraints or the objective function. To solve these problems a non-linear programming software is required.

3.3.7 Solution Procedures

3.3.7.1 Linear Programming based Procedure

In section 3.3.5 the solution obtained, by relaxing the integer constraints on the problem, was discussed. It was seen that the solution was highly infeasible due to the integer variables having values between 0 and 1. The motivation behind the analysis was to utilize the advantage of a linear program to solve an integer program. The procedure discussed subsequently is based on this idea.

Linear programs are preferred over integer programs due to the speed of computation. To use a linear program to obtain a feasible solution to an integer program it would be necessary to solve the linear program successively, with modifications every iteration, for many iterations. To do this the linear programming software CPLEX is used. CPLEX is mainly a UNIX based linear programming software. CPLEX has various pre-defined subroutines (called the Callable Library) which, helps the user to write programs to modify the working of CPLEX. In this case a C program is written (Appendix N) to obtain a feasible integer solution to a linear program.

The procedure used to obtain integer feasible solutions is as follows. Initially all variables binary variables are relaxed to be continuous between 0 and 1. This transforms the integer program into a linear program. The linear program is then solved and the solution is obtained. This solution will, most likely, have most of the desired binary variables with fractional values. The variable with the maximum fractional value (non-zero and not equal to one) is selected and is then constrained to have a value of 1. The problem is then resolved with this modification. If the problem is infeasible then the previous modification is reversed and the variable is set to 0. This essentially follows a branch and bound pattern, in which each variable is successively set to either zero or one. Since this is a minimization problem, any feasible solution obtained will serve as an upper bound for the remaining solutions. Hence if at any branch the calculated value of the objective function is greater than the upper bound, the branch is fathomed and no further iterations are carried along that branch. In this way all feasible solutions are obtained and the best solution is the optimal solution. The algorithm for using the branch and bound procedure is as follows:

Algorithm

Step 1. Relax the integer program into a linear program by allowing the binary variables to lie between zero and one. Let X be the set of binary variables. X = {x1,x2,….}
$0 \leq x \leq 1, \forall x \in X$

Step 2. Solve the linear program and obtain a solution.

Step 3. Check the values of all binary variables.
if x = 0 or x = 1, $\forall x \in X$…….. (1)
then STOP, the optimal solution has been obtained.

Step 4. If (1) is not true then identify the variable with the largest non integer variable.
$x_m = \max \{ x / x \neq 0, x \neq 1, x \in X\}$

Step 5. Set the value of x_m to 1. Resolve the problem.

Step 6. If the problem is infeasible, backtrack, set the value of xm to 0 and solve the problem, else obtain the values of all the variables in the optimal solution. if x = 0 or x = 1, $\forall x \in X$, then a feasible solution is obtained. If not go to step 4.

Step 7. If an upper bound is present and the present objective value is less than the upper bound then set the present objective value to be the new upper bound. If present objective value is larger than the upper bound then fathom the branch and backtrack. If no upper bound is present set the present objective value to be the upper bound and backtrack.

3.4 Conjunctive- Disjunctive Graph Based Approach

3.4.1 Introduction

As mentioned in earlier chapters, the wafer fabrication environment differs from the traditional flow shop and job shop environments. In a flow shop, jobs have an acyclic route through the system, never visiting a machine more than once. In a job shop, the jobs have different routes and may revisit the machines on which it had been processed earlier. In a wafer fabrication environment the flow is typically reentrant, with lots visiting the machines several times during their manufacturing cycle. However unlike a job shop, the routes are much less random and are more structured.

A methodology is proposed to reduce the maximum cycle time of a wafer fabrication facility using a conjunctive-disjunctive graph based approach. This approach is usually applied to job shops to reduce the maximum cycle time. However the more structured environment of wafer fabrication can be used to modify the approach. The application of the conjunctive disjunctive graph approach to a wafer fabrication environment is discussed. This approach has been developed only in theory and has not been applied or analyzed.

3.4.2 Approach

The problem can be represented by a conjunctive-disjunctive graph. This is shown in Figure 3.4. Each time the job recurs it is treated as a new job. Hence, if the total number of jobs is two and each job recurs once, the total number of jobs in the graph will be four. Each node in the graph is labeled i,j , where i is the job number and j denotes the processing stage. Note that jobs 1 and 2 are designated as jobs 3 and 4, respectively, during their reentrant phase.

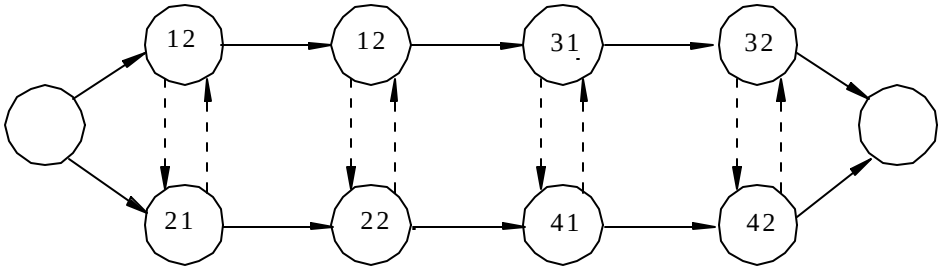


Figure 3.4: Conjunctive-Disjunctive Graph

The solid lines in the graph represent the precedence constraint among the jobs. The dashed lines represent the sequencing problem that needs to be solved to obtain the relationships between the jobs at each stage. The jobs need to be sequenced at each stage such that the times at the subsequent stages are affected the least. The ready times at the subsequent stages depend on the completion times of the jobs at the previous stage. Hence, this problem becomes a dynamic sequencing problem. In this case, at each stage, there are more than one machine in parallel on which the jobs have to be sequenced. Hence, this problem reduces to a problem of finding an effective algorithm to solve the $m/n/r_j$, L_{max} problem, which would then reduce the cycle time of the overall operation.

Here, m stands for the number of machines, n for the number of jobs, r_j for the ready time of job j and L_{max} represents maximum lateness. This would give us the entire schedule of the jobs that are in the system.

In case the machine, at which the L_{max} problem is to be applied, is a batching machine the problem becomes m, batching/n/r_j, L_{max}.

3.4.3 Application to a dynamic environment

Since this approach is mainly used for a static environment a few modifications need to be made when applying it to a dynamic environment, where lots enter and leave at regular intervals of time.

When considering machine downtimes or preventive maintenance, the heuristic would have to take it into account while solving the Lmax problem. If at that instant the machine is down, the heuristic will project when the machine is likely to be up again and find Lmax.

When new lots enter the system, the heuristic will have to be applied again since only the schedule for the lots in the system has been obtained.

Chapter 4: Results

4.1 Introduction

The fabrication area at M/A-COM was simulated using the AutoSched AP software. The fabrication area consists of 96 stations, which are classified into station families. There are six different product types, which flow through the system in the form of lots consisting of 8 wafers each. As indicated earlier these lots are not broken into sublots and flow through the system as a single unit. The number of steps in the route of each lot depends on the part types. Typically the number of steps could vary for 200 to 500 steps.

Also, the mathematical program, formulated in chapter 3, is run for very small models, under different scenarios, to gain an insight into the characteristics of the optimal solution. This analysis is used to justify the procedure used in developing the heuristic algorithm.

4.2 Results of the Proposed Heuristic Algorithm

The algorithm used is as described in the chapter 3. This algorithm is compared to various others already provided for in the software. The objective considered is the mean and variance of the cycle time of the lots flowing through the system. Initially deterministic processing times are assumed for the various steps. The various dispatching rules used are explained as follows:

Bottleneck: The proposed algorithm described in chapter 3.

Multbottleneck: The multiple bottleneck version of the proposed algorithm.

LPR: Least Percent Processing Time Remaining. This is calculated as follows,

$$\text{percent processing remaining} = \frac{\text{remaining processing time}}{\text{actual processing time} + \text{remaining processing time}}$$

The actual processing time is the processing time the lot has incurred up to the current step. The lot with the least percent processing time remaining, is ranked first.

LTR: Least Time Remaining. The lots are ranked in order of their remaining processing time. The lot with the least remaining processing time is ranked first. Remaining processing time is the sum of processing times at all remaining steps.

SPT: Shortest Processing Time. The lots are ranked by their processing time at the current step, shortest time first.

LLA: Least Lots Ahead. The lot that has the fewest lots (of any part type) at its next step is ranked highest.

LBA: Least Balance Ahead. The lot that has the fewest lots of the same part type at its next step is ranked highest.

Existing System: The scheduling policies used currently by M/A-COM at its facility. A description of these rules is given in chapter 1.

FIFO: First In, First Out. This rule selects the lot that arrived first on the worklist.

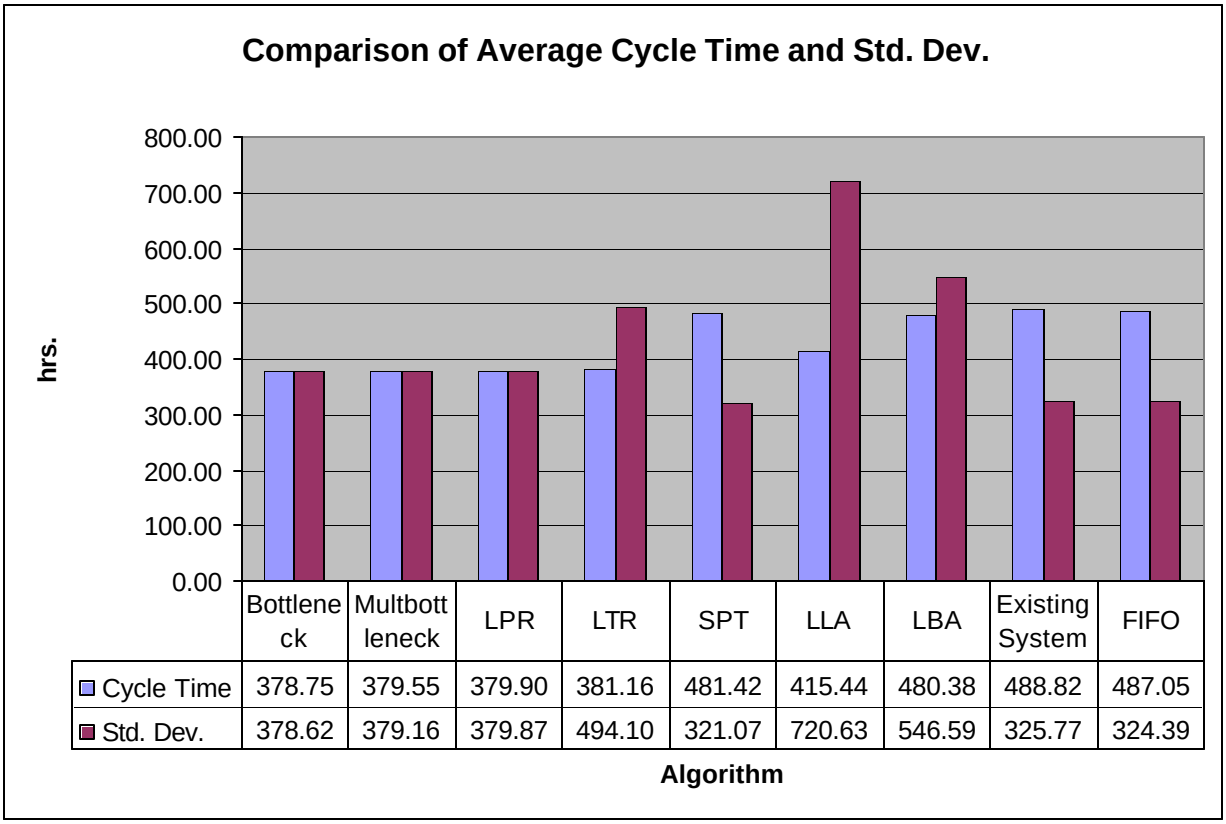
Experiments are carried out under various scenarios of order entries. The data used is deterministic data, which has been collected using previous time studies at M/A-COM. Currently data is being collected for the system and further studies would include stochastic times. The data used for the simulation runs can be found in Appendices A to K.

Two different sets of data are used to analyze the problem. Data set I consists of processing times that are entered using only one observation of the process. Data set II consists of processing times, which for some operations, is averaged over a set of three observations. For each data set different order files (i.e. arrival of products in the system) are used and the observations are depicted in a chart.

Section 4.2.1

Expt. 1

This experiment was conducted using the times given in the route files (ref. Appendices D to I) and no stations were batching stations. The order file used in this experiment is the original order file used at M/A-COM (ref. Appendix A).

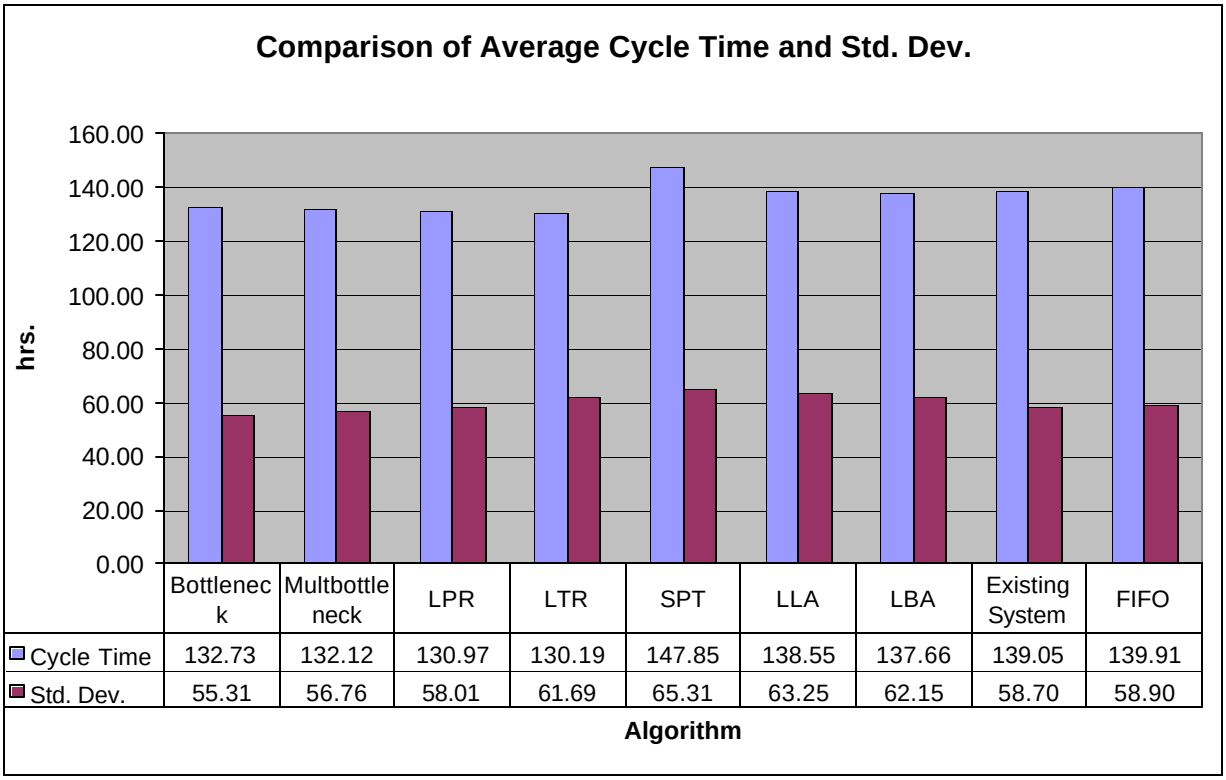


	Bottleneck	Multbottleneck	LPR	LTR	SPT	LLA	LBA	Existing System	FIFO
Cycle Time	378.75	379.55	379.90	381.16	481.42	415.44	480.38	488.82	487.05
Std. Dev.	378.62	379.16	379.87	494.10	321.07	720.63	546.59	325.77	324.39

Figure 4.1: Comparison of Average Cycle Time and Standard Deviation for expt.1 with Data Set I.

Expt. 2.

This experiment was conducted using the same data as in experiment 1, but this time the station polybake was assumed to be a batching station, with a minimum batch size of 1 and a maximum batch size of 8.

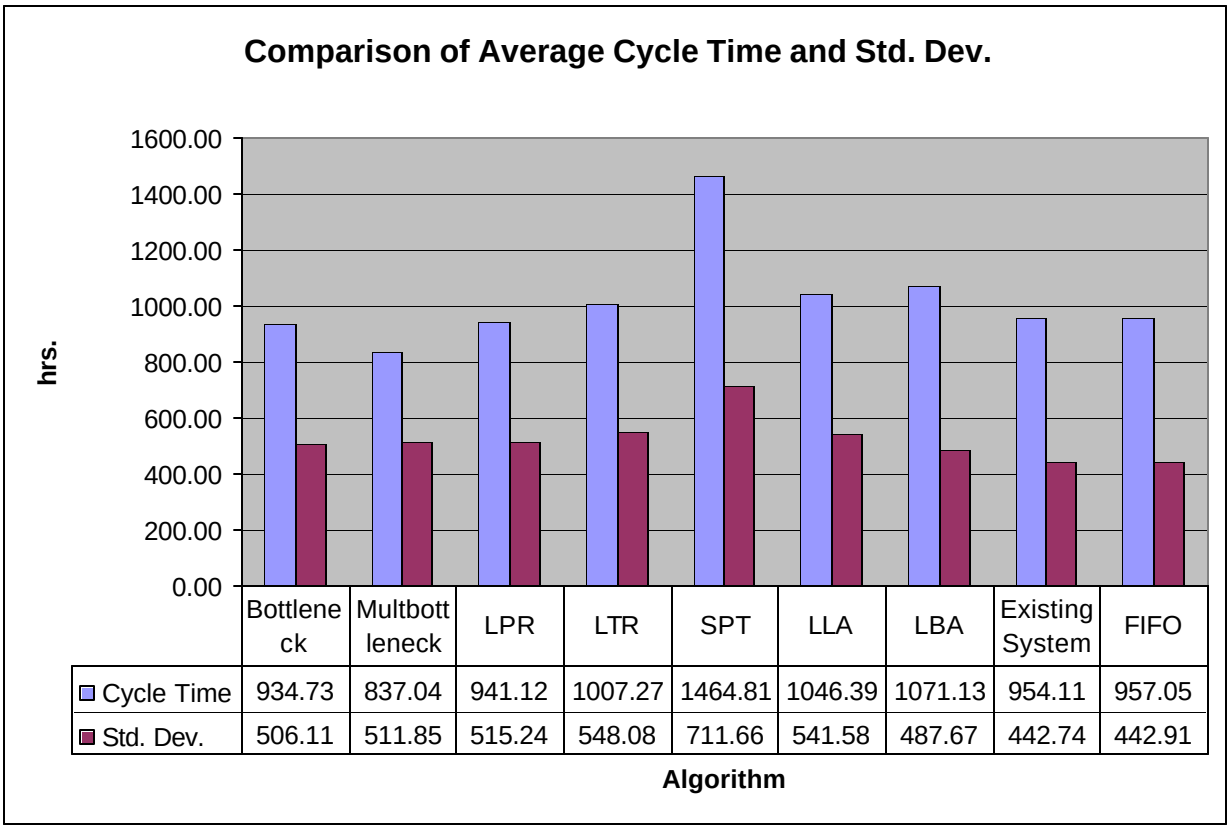


	Bottleneck	Multbottleneck	LPR	LTR	SPT	LLA	LBA	Existing System	FIFO
Cycle Time	132.73	132.12	130.97	130.19	147.85	138.55	137.66	139.05	139.91
Std. Dev.	55.31	56.76	58.01	61.69	65.31	63.25	62.15	58.70	58.90

Figure 4.2: Comparison of Average Cycle Time and Standard Deviation for expt.2 with Data Set I.

Expt 3.

This experiment was conducted using the same data as in experiment 2 but this time the order file was changed. The order file (ref. Appendix L.) causes orders to come in at a slightly higher rate than expt.'s 1 and 2.

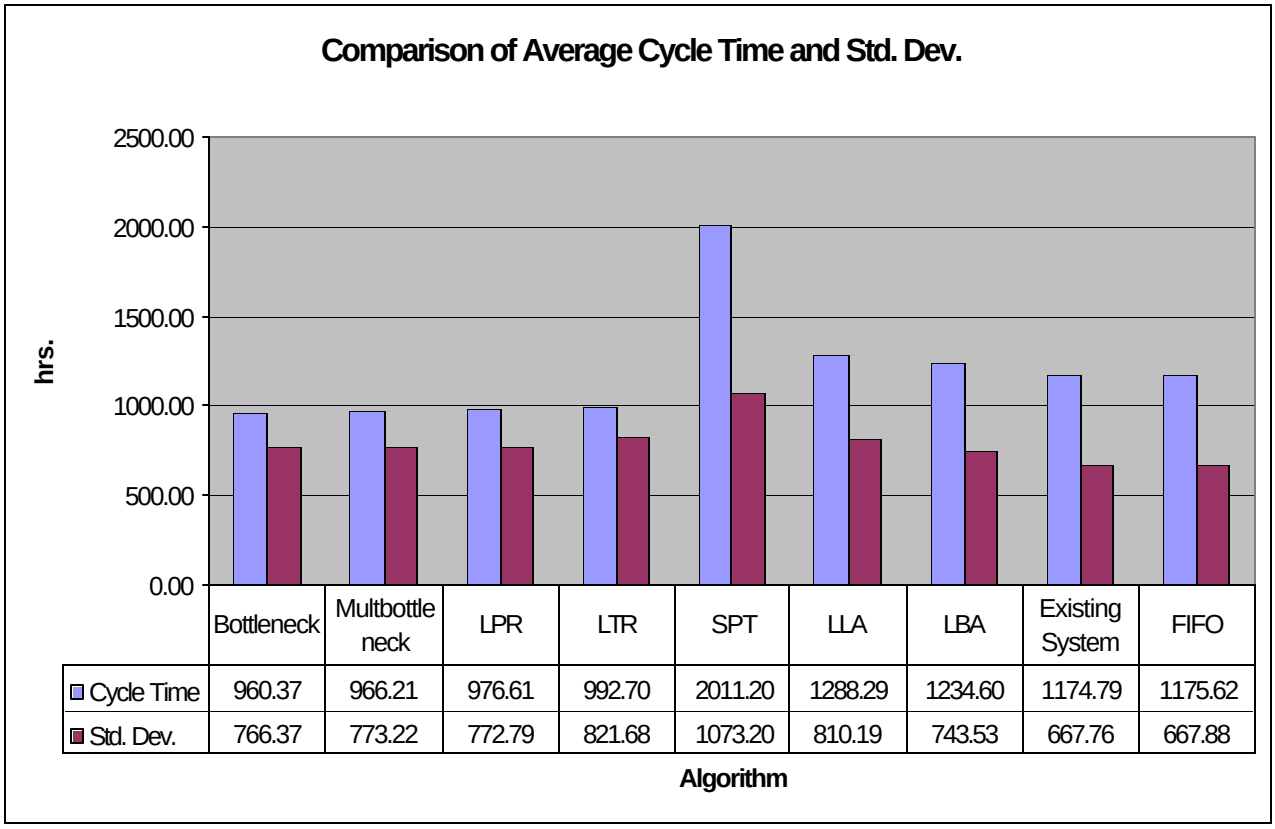


	Bottleneck	Multbottleneck	LPR	LTR	SPT	LLA	LBA	Existing System	FIFO
Cycle Time	934.73	837.04	941.12	1007.27	1464.81	1046.39	1071.13	954.11	957.05
Std. Dev.	506.11	511.85	515.24	548.08	711.66	541.58	487.67	442.74	442.91

Figure 4.3: Comparison of Average Cycle Time and Standard Deviation for expt.3 with Data Set I.

Expt 4.

This experiment was conducted using the same data as in experiment 3 with a changed order file. The changed order file is attached in Appendix L. In this case orders come in at a higher rate than expt. 3.



	Bottleneck	Multbottleneck	LPR	LTR	SPT	LLA	LBA	Existing System	FIFO
Cycle Time	960.37	966.21	976.61	992.70	2011.20	1288.29	1234.60	1174.79	1175.62
Std. Dev.	766.37	773.22	772.79	821.68	1073.20	810.19	743.53	667.76	667.88

Figure 4.4: Comparison of Average Cycle Time and Standard Deviation for expt.4 with Data Set I.

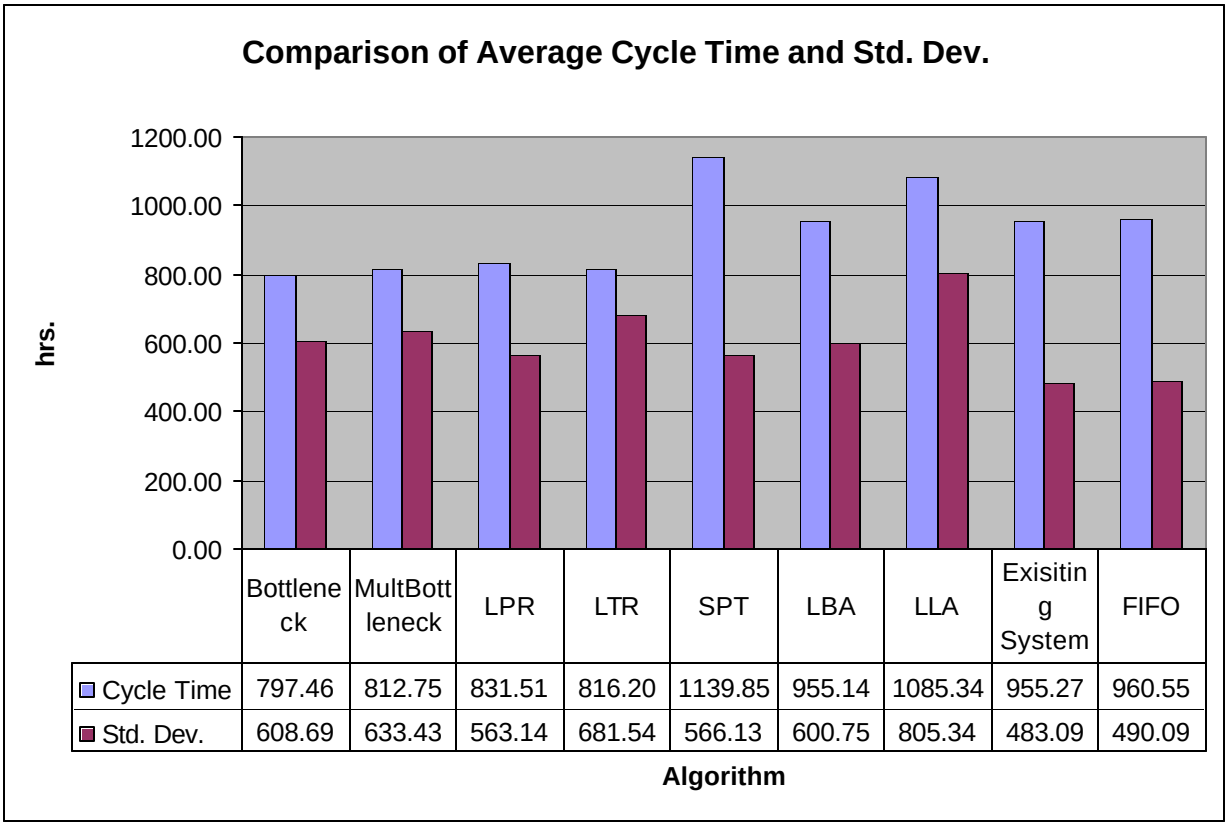
From the preceding graphs it is seen that the proposed algorithms perform better is almost all the cases in terms of mean cycle time. In experiment 1 it performs better that all the others in terms of average cycle time and is among the lower values in terms of standard deviations. In experiment 2, the proposed algorithms are the lowest in terms of standard deviation and among the lowest four in terms of average cycle time. In experiments 3 and 4 the proposed algorithms outperform all the other rules in terms of average cycle time.

Section 4.2.2

For all the experiments in this section the 'polybake' machine is a batching machine, with a maximum batch size of 8.

Expt. 1.

This experiment was conducted using the times given in the route files (ref. Appendices D to I). The order file used is given in Appendix L.

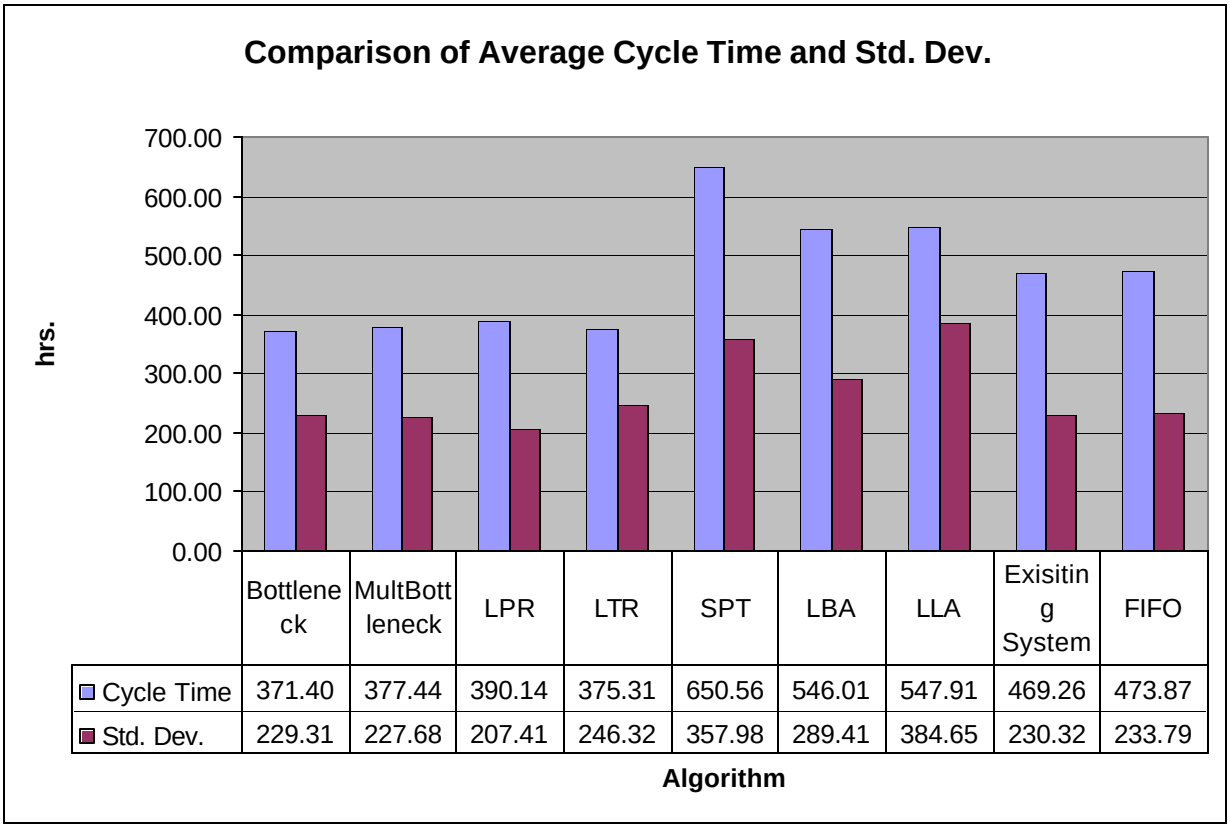


	Bottleneck	MultBottleneck	LPR	LTR	SPT	LBA	LLA	Exisiting System	FIFO
Cycle Time	797.46	812.75	831.51	816.20	1139.85	955.14	1085.34	955.27	960.55
Std. Dev.	608.69	633.43	563.14	681.54	566.13	600.75	805.34	483.09	490.09

Figure 4.5: Comparison of Average Cycle Time and Standard Deviation for expt. 1 with Data Set II.

Expt. 2.

This experiment was conducted using the times given in the route files (ref. Appendices D to I). This experiment is similar to expt. 1, however the order file was changed in this experiment (Appendix L). In this order file the products arrive more frequently and hence this causes an increase in the average cycle time of the system.

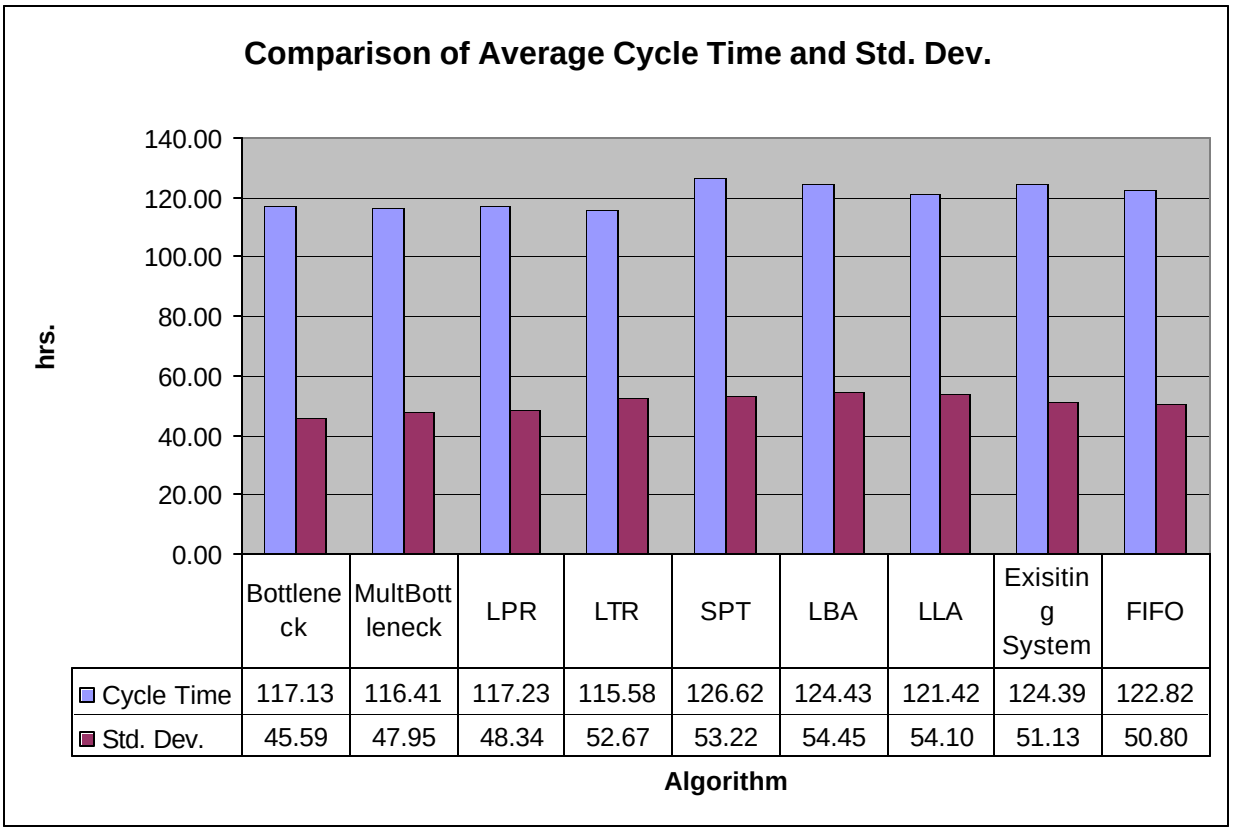


	Bottleneck	MultBottleneck	LPR	LTR	SPT	LBA	LLA	Exisiting System	FIFO
Cycle Time	371.40	377.44	390.14	375.31	650.56	546.01	547.91	469.26	473.87
Std. Dev.	229.31	227.68	207.41	246.32	357.98	289.41	384.65	230.32	233.79

Figure 4.6: Comparison of Average Cycle Time and Standard Deviation for expt. 2 with Data Set II.

Expt. 3.

This experiment was conducted using the times given in the route files (ref. Appendices D to I). The order file used in this experiment is the original order used at M/A-COM (ref. Appendix A).



	Bottleneck	MultBottleneck	LPR	LTR	SPT	LBA	LLA	Exisiting System	FIFO
Cycle Time	117.13	116.41	117.23	115.58	126.62	124.43	121.42	124.39	122.82
Std. Dev.	45.59	47.95	48.34	52.67	53.22	54.45	54.10	51.13	50.80

Figure 4.7: Comparison of Average Cycle Time and Standard Deviation for expt. 3 using Data Set II.

For the experiments in this section the proposed algorithms outperform all the others in terms of average cycle time, except in experiment 3 where Least Processing Time remaining performs better. However, even in experiment 3 the standard deviation of the proposed procedures outperform the where Least Processing Time Remaining heuristic.

Section 4.2.3

In addition to the above experiments, additional simulation runs were done to compare the performance of the proposed algorithm to the performance of the existing system. The experiments are divided into three categories:

1. Normal loading
2. Overloading (using 10% more lots/hr than normal loading)
3. Underloading (using 30% less lots/hr than normal loading)

Additionally, the runs were done at constant and variable input rates. The results are tabulated below:

	Normal loading		Overloading		Underloading	
	Mult Bottleneck	Existing System	Mult Bottleneck	Existing System	Mult Bottleneck	Existing System
Avg. Cycle time (days)	26.86	33.8	33.2	44.5	24.82	26.3
Std. Dev. (days)	3.86	4.99	5.2	7.3	4.02	3.63
Throughput (Lots/day)	617	567	600	609	437	424
Avg. WIP (lots)	100	133	131	200	66	73
WIP Std. Dev. (lots)	10	21	12	42	10	14

Table 4.1: Comparison between proposed algorithm and existing system at constant input

	Normal loading		Overloading		Underloading	
	Mult Bottleneck	Existing System	Mult Bottleneck	Existing System	Mult Bottleneck	Existing System
Avg. Cycle time (days)	25.37	28.48	30.78	35.17	24.10	24.14
Std. Dev. (days)	4.95	7.04	4.64	6.78	8.24	7.14
Throughput (Lots/day)	655	665	662	682	476	476
Avg. WIP (lots)	91	115	126	166	59	71
WIP Std. Dev. (lots)	18.5	34	16	33.02	22	26

Table 4.2: Comparison between proposed algorithm and existing system at variable input

From the above case studies, it is seen that for different order files the proposed algorithm(s) have a lesser average cycle time than all other heuristics, including the existing system. The order files are chosen in such a way that the system is loaded at low, medium and high levels.

4.3 Results of the Mathematical Programming Based Procedure

The integer program, formulated in chapter 3, is run for very small models and its results are analyzed. This serves as an insight into the characteristics of the optimal solution under varying conditions. The conditions that are considered are a flowshop environment, a reentrant environment, a machine with batching capabilities and dynamic arrival of jobs.

When a flowshop environment is considered, six products are considered to be in the system. However when a reentrant environment is analyzed, only three products are considered in the system. Since in a reentrant system the jobs visit the machines again, both systems can then be considered equivalent. The processing times are kept the same

for all the scenarios considered to ensure validity of analysis of the various experiments. The experiments are described below.

4.3.1 Experiments to analyze the significance of reentrant flow and batching stations

4.3.1.1 Expt. 1. : Flowshop environment with no batching stations.

In this experiment, six products are considered to be in the system and are all available at time zero. Since it is a flowshop all the products follow the same route and also have the same times at each machine. This is given in the following table:

Step No.	Machine No.	Processing Time
1	1	4
2	2	1

Table 4.3: Processing Time data for Expt. 1.

This problem is solved using the integer program and the following optimal solution is obtained.

Integer Optimal Solution

Product	Start time at Machine 1	Start Time at Machine 2	Finish Time
1	20	24	25
2	12	16	17
3	8	12	13
4	16	20	21
5	4	8	9
6	0	4	5

Table 4.4: Optimal Solution for Expt. 1.

Integer optimal solution: **Objective = 90**

4.3.1.2 Expt. 2. : Reentrant system with no batching stations

In this system, three products follow the same route. The route is such that after visiting the first two machines, as in the flowshop case, the products revisit the machines in the

same order. The route and time followed by the three products are identical and are as follows:

Step No.	Machine No.	Processing Time
1	1	4
2	2	1
3	1	4
4	2	1

Table 4.5: Processing Time Data for Expt. 2.

The problem is solved using the integer program and the optimal solution obtained is as follows:

Integer Optimal Solution

Product	Start time at Machine 1	Start Time at Machine 2	Start time at Machine 1	Start Time at Machine 2	Finish Time
1	0	4	8	12	13
2	4	10	12	16	17
3	16	20	21	25	26

Table 4.6: Optimal Solution for Expt. 2.

Integer optimal solution: **Objective = 56**

It is seen in this case that product 1 is started first on machine 1. Then product 2 is selected to be processed on machine 1. However when the processing of machine 1,for product 2, has finished, there are two products in its queue, namely product 1 and product 3. Product 1 has already completed half of its manufacturing cycle while product 3 is yet to begin its manufacturing cycle. In this case, the optimal solution lies in choosing product 1. A similar case is obtained at time 12, when product 2 and product 3 are competing for processing at machine 1. Here again, product 3 is yet to start while product 2 is halfway through. Again, the optimal decision lies in choosing product 2. This shows the benefit of picking the lot with the least percent processing time remaining.

4.3.1.3 Expt. 3: Dynamic Arrivals, with reentrant flow and no batching stations

This case is similar to experiment 2, however in this case, all products are not available at time zero. The ready times of the products are given as follows:

Product	Arrival Time
1	2
2	1
3	0

Table 4.7: Arrival Time Data for Expt. 3.

The problem is solved using the integer program and the optimal solution obtained is as follows:

Integer Optimal Solution

Product	Start time at Machine 1	Start Time at Machine 2	Start time at Machine 1	Start Time at Machine 2	Finish Time
1	16	20	21	25	26
2	4	8	12	16	17
3	0	5	8	12	13

Table 4.8: Optimal Solution for Expt. 3.

Integer optimal solution: Objective = 56

4.3.1.4 Expt. 4: Reentrant flow with a batching station

This experiment is similar to expt. 2, however now station 1 is the batching station with a maximum batch capacity of 2.

The problem is solved using the integer program and the optimal solution obtained is as follows:

Integer Optimal Solution

Product	Start time at Machine 1	Start Time at Machine 2	Start Time at Machine 1	Start Time at Machine 2	Finish Time
1	5	11	13	17	18
2	0	5	9	13	14
3	0	4	5	9	10

Table 4.9: Optimal Solution for Expt. 4.

Integer optimal solution: Objective = 42

Here at time zero, on machine 1, products 2 and 3 form a batch. This reduces the cycle time of each product. Also it is interesting to note that product 1 waits for 1 time unit, from 4 to 5, to allow product 3 to arrive at machine 1. This experiment shows the advantage of a batching station.

The above explained experiments provide an insight into the characteristic of reentrant flow and batching stations. Such insights are valuable for any research done in a system which involves these characteristics.

Chapter 5: Summary and Conclusions

This research work has been motivated by the need to develop an effective solution for the minimization of cycle time at M/A-COM's facility at Roanoke, VA, as a result of a large amount of Work-In-Process. Three methodologies have been proposed with the objective of minimizing cycle time. The first two methodologies have been developed and explained in chapter 3. The third methodology has been explained only in theory.

The heuristic developed in section 3.2 explains the bottleneck concept and it's incorporation in a dispatching rule. The bottleneck concept involves the scheduling of lots with the objective of minimizing the idleness at the bottleneck machine. An algorithm is developed based on this idea and implemented using a simulation software. The simulation software used is AutoSimulation's AutoSched AP. Using this simulation software, the effect of the heuristic is examined on the existing system. The existing system was modeled incorporating the rules and procedures used at M/A-COM's Roanoke facility. The results of the analysis are presented in Chapter 4. The proposed heuristic was found to outperform the existing procedure followed. The average cycle time is reduced by 11% and the average number of lots in WIP is reduced by 20%.

The second methodology discussed in Chapter 3 is related to mathematical programming. The working of the wafer fabrication system is modeled as an integer program. The starting times of the products at various operations were considered to be binary variables. Using this, the various constraints were setup and the problem was solved using CPLEX, which is a linear and integer programming software. The results of its implementation on problems of small sizes are discussed in the same chapter. Since the computation times of integer programs typically grow rapidly with increase in problem size, it is desired to build a more efficient solution procedure. With this in mind, a branch and bound algorithm using linear programming relaxations is discussed in Chapter 3. This method proposes to relax the problem and solve the resulting linear programs iteratively with additional constraints.

Bibliography

[1] “Production Scheduling Algorithms For A Semiconductor Test Facility”
R. Uzsoy, L.A. Martin-Vega, C.-Y. Lee and P.A. Leonard.
IEEE Transactions on Semiconductor Manufacturing, Vol. 4, No. 4, p.270-280, November 1991.

[2] “A Production Planning Methodology for Semiconductor Manufacturing Based on Iterative Simulation and Linear Programming Calculations”.
Y.-F. Hung and R.C. Leachman.
IEEE Transactions on Semiconductor Manufacturing, Vol. 9, No. 2, p.257-269, May 1996.

[3] “Scheduling Semiconductor Manufacturing Plants to Reduce Mean and Variance of Cycle Time”.
S.C.H. Lu, D. Ramaswamy and P.R. Kumar.
1993 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, p.83-85.

[4] “Scheduling Semiconductor Manufacturing Plants”.
P.R. Kumar.
IEEE Control Systems, December 19994, p.33-40.

[5] “Scheduling Job with Machine Dependant Processing Times”.
Y.-F. Hung and J.-L. Yang.
1996 IEEE/CMPT International Electronics Manufacturing Technology Symposium, p.42-48.

[6] “Effects of Dispatching and Down Time on the Performance of Wafer Fab Operating Under Theory of Constraints”.
D. Kayton, T. Teyner, C. Schwartz and Reha Uzsoy.
1996 IEEE/CMPT International Electronics Manufacturing Technology Symposium, p.49-56.

[7] “Performance Bounds for Queuing Networks and Scheduling Policies”.
S. Kumar and P.R. Kumar.
IEEE Transactions on Automatic Control, Vol 39, No. 8, August 1994.

[8] “Linear Control Rules for Production Control of Semiconductor Fabs”.

C.R. Glassey, J.G. Shantikumar and S. Seshadri.

IEEE Transactions on Semiconductor Manufacturing, Vol. 1, No. 1, pg36-46, February 1988.

[9] "A Robust Production Control Policy for VLSI Wafer Fabrication".

S.X.C. Lou and P.W. Kager.

IEEE Transactions on Semiconductor Manufacturing, Vol. 2, No. 4, p.159-164, November 1989.

[10] "Methods for Job Configuration in Semiconductor Manufacturing".

D.P. Connors and D.D. Yao.

IEEE Transactions on Semiconductor Manufacturing, Vol. 9, No. 3, p401-411, August 1996.

[11] "Hierarchical Real-Time Scheduling of a Semiconductor Fabrication Facility".

X. Bai, No. Srivatsan and S.B. Gershwin.

1990 IEEE/CMPT International Electronics Manufacturing Technology Symposium, p.312-317.

[12] "Minimizing Total Tardiness on a Batch Processing Machine with Incompatible Job Families".

S.V. Mehta and R. Uzsoy.

IIE Transactions (1998)30, p.165-178.

[13] "Simulation-Based Scheduling: Meeting the Semiconductor Wafer Fabrication Challenge".

M. Thompson.

IIE Solutions, p.30-34, May 1993.

[14] "Hierarchical Modeling and Control of Reentrant Semiconductor Manfuacturing Facilities".

M.K. El Adl, A.A. Rodriguez and K.S. Tsakalis.

IEEE Proceedings of the 35th Conference on Decision and Control, Kobe, Japan, December 1996, p.1736-1742.

[15] "Modeling, Scheduling and Dispatching in the Dynamic Environment of Semiconductor Manufacturing at FASL, Japan".

B. Pickett, M. Zuniga.

1997 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, p.448-450.

[16] “A near-optimal heuristic for the lot streaming sequencing problem in multiple-batch flow-shops”.
Adar A. Kalir, Subhash C. Sarin.

Appendices

Appendix A: The Orders Worksheet

ORDER	LOT	PART	START	DUE
Order1	L1	p5	3/20/00 8:00	3/31/00 0:00
Order2	L2	p5	3/20/00 8:00	3/31/00 0:00
Order3	L3	p1	3/20/00 8:00	4/7/00 0:00
Order4	L4	p5	3/20/00 8:00	4/7/00 0:00
Order5	L5	p5	3/22/00 8:00	4/7/00 0:00
Order6	L6	p1	3/20/00 8:00	4/14/00 0:00
Order7	L7	p1	3/20/00 8:00	4/14/00 0:00
Order8	L8	p5	3/20/00 8:00	4/28/00 0:00
Order9	L9	p2	3/24/00 8:00	4/7/00 0:00
Order10	L10	p6	3/24/00 8:00	4/8/00 0:00
Order11	L11	p6	3/24/00 8:00	4/8/00 0:00
Order12	L12	p5	3/24/00 8:00	4/18/00 0:00
Order13	L13	p5	3/24/00 8:00	4/18/00 0:00
Order14	L14	p1	3/24/00 8:00	4/21/00 0:00
Order15	L15	p1	3/24/00 8:00	4/21/00 0:00
Order16	L16	p1	3/24/00 8:00	4/21/00 0:00
Order17	L17	p1	3/24/00 8:00	4/21/00 0:00
Order18	L18	p1	3/24/00 8:00	4/21/00 0:00
Order19	L19	p1	3/24/00 8:00	4/21/00 0:00
Order20	L20	p1	3/24/00 8:00	4/21/00 0:00
Order21	L21	p5	3/24/00 8:00	4/21/00 0:00
Order22	L22	p2	3/24/00 8:00	4/15/00 0:00
Order23	L23	p2	3/24/00 8:00	4/15/00 0:00
Order24	L24	p2	3/24/00 8:00	4/15/00 0:00
Order25	L25	p2	3/24/00 8:00	4/15/00 0:00
Order26	L26	p2	3/24/00 8:00	4/15/00 0:00
Order27	L27	p4	3/24/00 8:00	5/2/00 0:00
Order28	L28	p4	3/24/00 8:00	5/2/00 0:00
Order29	L29	p4	3/24/00 8:00	5/2/00 0:00
Order30	L30	p4	3/24/00 8:00	5/2/00 0:00
Order31	L31	p3	3/30/00 8:00	3/20/00 0:00
Order32	L32	p6	3/27/00 8:00	4/16/00 0:00
Order33	L33	p6	3/27/00 8:00	4/16/00 0:00
Order34	L34	p6	3/27/00 8:00	4/16/00 0:00
Order35	L35	p3	3/27/00 8:00	4/18/00 0:00
Order36	L36	p3	3/27/00 8:00	4/18/00 0:00
Order37	L37	p3	3/27/00 8:00	4/18/00 0:00
Order38	L38	p2	3/28/00 8:00	5/5/00 0:00
Order39	L39	p5	3/30/00 8:00	4/22/00 0:00
Order40	L40	p6	3/30/00 8:00	4/13/00 0:00
Order41	L41	p5	3/30/00 8:00	4/17/00 0:00
Order42	L42	p2	3/30/00 8:00	4/18/00 0:00
Order43	L43	p2	3/30/00 8:00	4/18/00 0:00
Order44	L44	p2	3/30/00 8:00	4/18/00 0:00
Order45	L45	p3	3/30/00 8:00	4/20/00 0:00

ORDER	LOT	PART	START	DUE
Order48	L48	p3	3/30/00 8:00	4/20/00 0:00
Order49	L49	p3	3/30/00 8:00	4/20/00 0:00
Order50	L50	p2	3/31/00 8:00	4/21/00 0:00
Order51	L51	p2	3/31/00 8:00	4/21/00 0:00
Order52	L52	p2	3/31/00 8:00	4/21/00 0:00
Order53	L53	p2	3/31/00 8:00	4/21/00 0:00
Order54	L54	p5	3/30/00 8:00	4/10/00 0:00
Order55	L55	p5	3/30/00 8:00	4/20/00 0:00
Order56	L56	p5	4/6/00 8:00	3/31/00 0:00
Order57	L57	p5	4/6/00 8:00	4/21/00 0:00
Order58	L58	p3	4/6/00 8:00	4/21/00 0:00
Order59	L59	p1	4/7/00 8:00	4/20/00 0:00
Order60	L60	p5	4/7/00 8:00	4/28/00 0:00
Order61	L61	p5	4/7/00 8:00	4/28/00 0:00
Order62	L62	p2	4/7/00 8:00	4/22/00 0:00
Order63	L63	p5	4/13/00 8:00	4/21/00 0:00
Order64	L64	p3	4/11/00 8:00	4/20/00 0:00
Order65	L65	p2	4/11/00 8:00	4/21/00 0:00
Order66	L66	p2	4/11/00 8:00	4/21/00 0:00
Order67	L67	p2	4/13/00 8:00	4/21/00 0:00
Order68	L68	p2	4/13/00 8:00	4/21/00 0:00
Order69	L69	p6	4/13/00 8:00	4/24/00 0:00
Order70	L70	p6	4/13/00 8:00	4/24/00 0:00
Order71	L71	p3	4/13/00 8:00	4/26/00 0:00
Order72	L72	p3	4/13/00 8:00	4/26/00 0:00
Order73	L73	p3	4/13/00 8:00	5/1/00 0:00
Order74	L74	p2	4/13/00 8:00	5/3/00 0:00
Order75	L75	p2	4/13/00 8:00	5/5/00 0:00
Order76	L76	p2	4/14/00 8:00	5/5/00 0:00
Order77	L77	p2	4/14/00 8:00	5/5/00 0:00
Order78	L78	p2	4/14/00 8:00	5/5/00 0:00
Order79	L79	p2	4/14/00 8:00	5/5/00 0:00
Order80	L80	p2	4/14/00 8:00	5/5/00 0:00
Order81	L81	p2	4/14/00 8:00	5/5/00 0:00
Order82	L82	p2	4/14/00 8:00	5/5/00 0:00
Order83	L83	p2	4/14/00 8:00	5/5/00 0:00
Order84	L84	p2	4/14/00 8:00	5/5/00 0:00
Order85	L85	p2	4/14/00 8:00	5/5/00 0:00
Order86	L86	p2	4/14/00 8:00	5/5/00 0:00
Order87	L87	p3	4/14/00 8:00	5/8/00 0:00
Order88	L88	p3	4/14/00 8:00	5/9/00 0:00
Order89	L89	p3	4/14/00 8:00	5/9/00 0:00
Order90	L90	p6	4/14/00 8:00	5/12/00 0:00
Order91	L91	p2	4/14/00 8:00	5/12/00 0:00
Order92	L92	p2	4/14/00 8:00	5/12/00 0:00

ORDER	LOT	PART	START	DUE
Order97	L97	p5	4/19/00 8:00	5/16/00 0:00
Order98	L98	p5	4/20/00 8:00	5/16/00 0:00
Order99	L99	p5	4/20/00 8:00	5/16/00 0:00
Order100	L100	p1	4/19/00 8:00	5/13/00 0:00
Order101	L101	p3	4/19/00 8:00	5/12/00 0:00
Order102	L102	p3	4/19/00 8:00	5/12/00 0:00
Order103	L103	p2	4/19/00 8:00	5/12/00 0:00
Order104	L104	p2	4/19/00 8:00	5/12/00 0:00
Order105	L105	p4	4/20/00 8:00	5/12/00 0:00
Order106	L106	p2	4/19/00 8:00	5/12/00 0:00
Order107	L107	p2	4/19/00 8:00	5/12/00 0:00
Order108	L108	p4	4/20/00 8:00	5/12/00 0:00
Order109	L109	p4	4/20/00 8:00	5/12/00 0:00
Order110	L110	p4	4/20/00 8:00	5/12/00 0:00
Order111	L111	p3	4/20/00 8:00	5/18/00 0:00
Order112	L112	p3	4/20/00 8:00	5/18/00 0:00
Order113	L113	p3	4/20/00 8:00	5/18/00 0:00
Order114	L114	p3	4/20/00 8:00	5/18/00 0:00
Order115	L115	p3	4/20/00 8:00	5/18/00 0:00
Order116	L116	p3	4/20/00 8:00	5/18/00 0:00
Order117	L117	p2	4/20/00 8:00	5/19/00 0:00
Order118	L118	p2	4/20/00 8:00	5/19/00 0:00
Order119	L119	p2	4/20/00 8:00	5/19/00 0:00
Order120	L120	p2	4/20/00 8:00	5/19/00 0:00
Order121	L121	p2	4/20/00 8:00	5/19/00 0:00
Order122	L122	p2	4/20/00 8:00	5/19/00 0:00
Order123	L123	p2	4/20/00 8:00	5/19/00 0:00
Order124	L124	p2	4/20/00 8:00	5/19/00 0:00
Order125	L125	p2	4/20/00 8:00	5/19/00 0:00
Order126	L126	p2	4/20/00 8:00	5/19/00 0:00
Order127	L127	p5	4/28/00 8:00	5/19/00 0:00
Order128	L128	p5	4/28/00 8:00	5/19/00 0:00
Order129	L129	p4	4/28/00 8:00	5/12/00 0:00
Order130	L130	p4	4/28/00 8:00	5/12/00 0:00
Order131	L131	p4	4/28/00 8:00	5/12/00 0:00
Order132	L132	p2	4/28/00 8:00	5/12/00 0:00
Order133	L133	p2	4/28/00 8:00	5/12/00 0:00
Order134	L134	p2	4/28/00 8:00	5/12/00 0:00
Order135	L135	p3	4/28/00 8:00	5/18/00 0:00
Order136	L136	p3	4/28/00 8:00	5/18/00 0:00
Order137	L137	p3	4/28/00 8:00	5/18/00 0:00
Order138	L138	p3	4/28/00 8:00	5/18/00 0:00
Order139	L139	p3	4/28/00 8:00	5/18/00 0:00
Order140	L140	p3	4/28/00 8:00	5/18/00 0:00
Order141	L141	p6	4/28/00 8:00	5/24/00 0:00

ORDER	LOT	PART	START	DUE
Order146	L146	p5	5/4/00 8:00	5/26/00 0:00
Order147	L147	p4	5/4/00 0:00	5/16/00 0:00
Order148	L148	p3	5/4/00 8:00	5/20/00 0:00
Order149	L149	p3	5/4/00 8:00	5/20/00 0:00
Order150	L150	p3	5/4/00 8:00	5/20/00 0:00
Order151	L151	p3	5/4/00 8:00	5/19/00 0:00
Order152	L152	p6	5/4/00 8:00	5/20/00 0:00
Order153	L153	p6	5/4/00 8:00	5/20/00 0:00
Order154	L154	p6	5/4/00 8:00	5/20/00 0:00
Order155	L155	p5	5/5/00 8:00	5/27/00 0:00
Order156	L156	p6	5/4/00 8:00	5/24/00 0:00
Order157	L157	p6	5/4/00 8:00	5/24/00 0:00
Order158	L158	p5	5/5/00 8:00	5/31/00 0:00
Order159	L159	p5	5/5/00 8:00	5/31/00 0:00
Order160	L160	p3	5/5/00 8:00	5/24/00 0:00
Order161	L161	p3	5/5/00 8:00	5/24/00 0:00
Order162	L162	p3	5/5/00 8:00	5/26/00 0:00
Order163	L163	p6	5/4/00 8:00	5/26/00 0:00
Order164	L164	p6	5/4/00 8:00	5/27/00 0:00
Order165	L165	p6	5/4/00 8:00	5/28/00 0:00
Order166	L166	p2	5/5/00 8:00	5/26/00 0:00
Order167	L167	p2	5/5/00 8:00	5/26/00 0:00
Order168	L168	p2	5/5/00 8:00	5/26/00 0:00
Order169	L169	p2	5/5/00 8:00	5/26/00 0:00
Order170	L170	p2	5/5/00 8:00	5/26/00 0:00
Order171	L171	p2	5/5/00 8:00	5/26/00 0:00
Order172	L172	p3	5/5/00 8:00	5/26/00 0:00
Order173	L173	p4	5/5/00 8:00	5/26/00 0:00
Order174	L174	p4	5/5/00 8:00	5/26/00 0:00
Order175	L175	p4	5/5/00 8:00	5/26/00 0:00
Order176	L176	p4	5/5/00 8:00	5/26/00 0:00
Order177	L177	p5	5/5/00 8:00	6/7/00 0:00
Order178	L178	p5	5/5/00 8:00	6/7/00 0:00
Order179	L179	p3	5/5/00 8:00	6/1/00 0:00
Order180	L180	p2	5/5/00 8:00	6/1/00 0:00
Order181	L181	p2	5/5/00 8:00	6/1/00 0:00
Order182	L182	p2	5/5/00 8:00	6/1/00 0:00
Order183	L183	p2	5/5/00 8:00	6/1/00 0:00
Order184	L184	p1	1/25/00 8:00	2/8/00 0:00
Order185	L185	p5	1/25/00 8:00	2/11/00 0:00
Order186	L186	p3	1/26/00 8:00	2/11/00 0:00
Order187	L187	p3	1/26/00 8:00	2/11/00 0:00
Order188	L188	p3	1/26/00 8:00	2/11/00 0:00
Order189	L189	p3	1/26/00 8:00	2/11/00 0:00
Order190	L190	p3	1/26/00 8:00	2/11/00 0:00

ORDER	LOT	PART	START	DUE
Order195	L195	p6	1/27/00 8:00	2/23/00 0:00
Order196	L196	p6	1/27/00 8:00	2/24/00 0:00
Order197	L197	p6	1/27/00 8:00	2/24/00 0:00
Order198	L198	p3	1/27/00 8:00	2/24/00 0:00
Order199	L199	p3	1/27/00 8:00	2/25/00 0:00
Order200	L200	p3	1/27/00 8:00	2/26/00 0:00
Order201	L201	p3	1/27/00 8:00	2/27/00 0:00
Order202	L202	p1	1/27/00 8:00	2/28/00 0:00
Order203	L203	p1	1/27/00 8:00	2/28/00 0:00
Order204	L204	p3	1/27/00 8:00	2/28/00 0:00
Order205	L205	p5	1/28/00 8:00	2/24/00 0:00
Order206	L206	p3	2/1/00 8:00	2/21/00 0:00
Order207	L207	p3	2/4/00 8:00	2/23/00 0:00
Order208	L208	p6	2/4/00 8:00	2/29/00 0:00
Order209	L209	p6	2/4/00 8:00	2/29/00 0:00
Order210	L210	p3	2/4/00 8:00	2/29/00 0:00
Order211	L211	p6	2/4/00 8:00	3/1/00 0:00
Order212	L212	p5	2/7/00 8:00	2/25/00 0:00
Order213	L213	p3	2/7/00 8:00	2/25/00 0:00
Order214	L214	p6	2/7/00 8:00	3/2/00 0:00
Order215	L215	p6	2/7/00 8:00	3/3/00 0:00
Order216	L216	p6	2/7/00 8:00	3/3/00 0:00
Order217	L217	p6	2/7/00 8:00	3/3/00 0:00
Order218	L218	p5	2/7/00 8:00	3/3/00 0:00
Order219	L219	p5	2/8/00 8:00	3/1/00 0:00
Order220	L220	p5	2/8/00 8:00	3/1/00 0:00
Order221	L221	p3	2/10/00 8:00	3/1/00 0:00
Order222	L222	p2	2/10/00 8:00	3/3/00 0:00
Order223	L223	p3	2/10/00 8:00	3/3/00 0:00
Order224	L224	p3	2/10/00 8:00	3/3/00 0:00
Order225	L225	p6	2/10/00 8:00	3/4/00 0:00
Order226	L226	p6	2/14/00 8:00	3/4/00 0:00
Order227	L227	p6	2/14/00 8:00	3/5/00 0:00
Order228	L228	p6	2/14/00 8:00	3/9/00 0:00
Order229	L229	p3	2/14/00 8:00	3/10/00 0:00
Order230	L230	p5	2/14/00 8:00	3/13/00 0:00
Order231	L231	p5	2/15/00 8:00	3/10/00 0:00
Order232	L232	p5	2/15/00 8:00	3/10/00 0:00
Order233	L233	p5	2/15/00 8:00	3/10/00 0:00
Order234	L234	p3	2/15/00 8:00	3/10/00 0:00
Order235	L235	p5	2/15/00 8:00	3/13/00 0:00
Order236	L236	p3	2/15/00 8:00	3/23/00 0:00
Order237	L237	p3	2/15/00 8:00	3/23/00 0:00
Order238	L238	p3	2/15/00 8:00	3/23/00 0:00
Order239	L239	p3	2/15/00 8:00	3/23/00 0:00

ORDER	LOT	PART	START	DUE
Order244	L244	p3	2/15/00 8:00	3/27/00 0:00
Order245	L245	p2	2/15/00 8:00	3/27/00 0:00
Order246	L246	p2	2/15/00 8:00	3/27/00 0:00
Order247	L247	p6	2/15/00 8:00	3/27/00 0:00
Order248	L248	p6	2/15/00 8:00	3/27/00 0:00
Order249	L249	p4	2/15/00 8:00	3/28/00 0:00
Order250	L250	p4	2/15/00 8:00	3/28/00 0:00
Order251	L251	p3	2/15/00 8:00	3/30/00 0:00
Order252	L252	p3	2/15/00 8:00	3/30/00 0:00
Order253	L253	p3	2/15/00 8:00	3/30/00 0:00
Order254	L254	p3	2/15/00 8:00	3/30/00 0:00
Order255	L255	p3	2/15/00 8:00	3/30/00 0:00
Order256	L256	p3	2/15/00 8:00	3/30/00 0:00
Order257	L257	p3	2/15/00 8:00	3/30/00 0:00
Order258	L258	p3	2/15/00 8:00	3/30/00 0:00
Order259	L259	p6	2/15/00 8:00	4/3/00 0:00
Order260	L260	p6	2/15/00 8:00	4/3/00 0:00
Order261	L261	p6	2/15/00 8:00	4/3/00 0:00
Order262	L262	p6	2/15/00 8:00	4/3/00 0:00
Order263	L263	p3	2/15/00 8:00	4/5/00 0:00
Order264	L264	p3	2/15/00 8:00	4/7/00 0:00
Order265	L265	p3	2/15/00 8:00	4/7/00 0:00
Order266	L266	p3	2/15/00 8:00	4/7/00 0:00
Order267	L267	p3	2/15/00 8:00	4/7/00 0:00
Order268	L268	p3	2/15/00 8:00	4/7/00 0:00
Order269	L269	p3	2/15/00 8:00	4/7/00 0:00
Order270	L270	p3	2/17/00 8:00	3/8/00 0:00
Order271	L271	p3	2/17/00 8:00	3/8/00 0:00
Order272	L272	p3	2/17/00 8:00	3/8/00 0:00
Order273	L273	p3	2/17/00 8:00	3/9/00 0:00
Order274	L274	p1	2/23/00 8:00	3/22/00 0:00
Order275	L275	p2	2/24/00 8:00	4/4/00 0:00
Order276	L276	p6	2/25/00 8:00	3/17/00 0:00
Order277	L277	p3	3/1/00 8:00	3/30/00 0:00
Order278	L278	p3	3/1/00 8:00	3/30/00 0:00
Order279	L279	p3	3/1/00 8:00	3/30/00 0:00
Order280	L280	p5	3/3/00 8:00	3/23/00 0:00
Order281	L281	p1	3/3/00 8:00	4/7/00 0:00
Order282	L282	p2	3/7/00 8:00	4/4/00 0:00
Order283	L283	p4	3/7/00 8:00	4/4/00 0:00
Order284	L284	p4	3/7/00 8:00	4/4/00 0:00
Order285	L285	p5	3/8/00 8:00	3/30/00 0:00
Order286	L286	p5	3/10/00 8:00	3/23/00 0:00
Order287	L287	p3	3/11/00 8:00	3/29/00 0:00
Order288	L288	p3	3/11/00 8:00	4/14/00 0:00

ORDER	LOT	PART	START	DUE
Order293	L293	p3	3/11/00 8:00	4/21/00 0:00
Order294	L294	p3	3/11/00 8:00	4/21/00 0:00
Order295	L295	p3	3/11/00 8:00	4/21/00 0:00
Order296	L296	p3	3/11/00 8:00	4/21/00 0:00
Order297	L297	p3	3/11/00 8:00	4/21/00 0:00
Order298	L298	p3	3/11/00 8:00	4/21/00 0:00
Order299	L299	p5	3/13/00 8:00	3/30/00 0:00
Order300	L300	p3	3/13/00 8:00	4/21/00 0:00
Order301	L301	p3	3/13/00 8:00	4/21/00 0:00
Order302	L302	p3	3/13/00 8:00	4/21/00 0:00
Order303	L303	p3	3/13/00 8:00	4/21/00 0:00
Order304	L304	p3	3/13/00 8:00	4/21/00 0:00
Order305	L305	p3	3/13/00 8:00	4/21/00 0:00
Order306	L306	p3	3/13/00 8:00	4/21/00 0:00
Order307	L307	p3	3/13/00 8:00	4/22/00 0:00
Order308	L308	p3	3/13/00 8:00	4/23/00 0:00
Order309	L309	p3	3/13/00 8:00	4/24/00 0:00
Order310	L310	p3	3/13/00 8:00	4/28/00 0:00
Order311	L311	p3	3/13/00 8:00	4/28/00 0:00
Order312	L312	p3	3/13/00 8:00	4/28/00 0:00
Order313	L313	p3	3/13/00 8:00	4/28/00 0:00
Order314	L314	p3	3/13/00 8:00	4/28/00 0:00
Order315	L315	p3	3/13/00 8:00	4/28/00 0:00
Order316	L316	p2	3/16/00 8:00	3/31/00 0:00
Order317	L317	p2	3/17/00 8:00	4/4/00 0:00
Order318	L318	p6	3/17/00 8:00	4/7/00 0:00
Order319	L319	p6	3/17/00 8:00	4/14/00 0:00
Order320	L320	p6	3/17/00 8:00	4/14/00 0:00
Order321	L321	p6	3/17/00 8:00	4/14/00 0:00
Order322	L322	p6	3/17/00 8:00	4/14/00 0:00
Order323	L323	p6	3/17/00 8:00	4/14/00 0:00
Order324	L324	p6	3/17/00 8:00	4/14/00 0:00
Order325	L325	p6	3/17/00 8:00	4/16/00 0:00
Order326	L326	p6	3/17/00 8:00	4/16/00 0:00
Order327	L327	p6	3/17/00 8:00	4/17/00 0:00
Order328	L328	p6	3/17/00 8:00	4/17/00 0:00
Order329	L329	p6	3/17/00 8:00	4/17/00 0:00
Order330	L330	p6	3/17/00 8:00	4/17/00 0:00
Order331	L331	p6	3/17/00 8:00	4/17/00 0:00
Order332	L332	p6	3/17/00 8:00	4/20/00 0:00
Order333	L333	p6	3/17/00 8:00	4/20/00 0:00
Order334	L334	p6	3/17/00 8:00	4/20/00 0:00
Order335	L335	p6	3/17/00 8:00	4/20/00 0:00
Order336	L336	p6	3/17/00 8:00	4/20/00 0:00
Order337	L337	p3	3/16/00 8:00	4/28/00 0:00

ORDER	LOT	PART	START	DUE
Order342	L342	p3	3/16/00 8:00	5/5/00 0:00
Order343	L343	p5	1/3/00 8:00	1/12/00 0:00
Order344	L344	p3	1/4/00 8:00	1/31/00 0:00
Order345	L345	p3	1/4/00 8:00	1/1/00 0:00
Order346	L346	p3	1/4/00 8:00	2/1/00 0:00
Order347	L347	p3	1/4/00 8:00	2/1/00 0:00
Order348	L348	p3	1/4/00 8:00	2/2/00 0:00
Order349	L349	p3	1/4/00 8:00	2/2/00 0:00
Order350	L350	p3	1/4/00 8:00	2/3/00 0:00
Order351	L351	p3	1/4/00 8:00	2/3/00 0:00
Order352	L352	p3	1/4/00 8:00	2/4/00 0:00
Order353	L353	p3	1/4/00 8:00	2/5/00 0:00
Order354	L354	p3	1/4/00 8:00	2/5/00 0:00
Order355	L355	p6	1/4/00 8:00	1/28/00 0:00
Order356	L356	p6	1/4/00 8:00	1/29/00 0:00
Order357	L357	p6	1/4/00 8:00	1/30/00 0:00
Order358	L358	p2	1/4/00 8:00	1/21/00 0:00
Order359	L359	p2	1/4/00 7:00	1/21/00 0:00
Order360	L360	p2	1/4/00 8:00	1/21/00 0:00
Order361	L361	p6	1/4/00 8:00	1/29/00 0:00
Order362	L362	p6	1/4/00 8:00	1/30/00 0:00
Order363	L363	p6	1/4/00 8:00	1/31/00 0:00
Order364	L364	p6	1/5/00 8:00	1/30/00 0:00
Order365	L365	p6	1/5/00 8:00	1/30/00 0:00
Order366	L366	p6	1/5/00 8:00	1/30/00 0:00
Order367	L367	p6	1/5/00 8:00	1/30/00 0:00
Order368	L368	p6	1/5/00 8:00	1/31/00 0:00
Order369	L369	p6	1/5/00 8:00	1/31/00 0:00
Order370	L370	p6	1/5/00 8:00	1/31/00 0:00
Order371	L371	p5	1/5/00 8:00	1/28/00 0:00
Order372	L372	p5	1/5/00 8:00	1/28/00 0:00
Order373	L373	p5	1/6/00 8:00	2/1/99 0:00
Order374	L374	p3	1/11/00 8:00	5/2/00 0:00
Order375	L375	p3	1/11/00 8:00	2/23/00 0:00
Order376	L376	p3	1/11/00 8:00	4/24/00 0:00
Order377	L377	p3	1/11/00 8:00	4/24/00 0:00
Order378	L378	p3	1/11/00 8:00	2/14/00 0:00
Order379	L379	p3	1/11/00 8:00	2/15/00 0:00
Order380	L380	p3	1/11/00 8:00	2/15/00 0:00
Order381	L381	p3	1/11/00 8:00	2/16/00 0:00
Order382	L382	p3	1/11/00 8:00	2/16/00 0:00
Order383	L383	p3	1/11/00 8:00	2/17/00 0:00
Order384	L384	p3	1/11/00 8:00	2/17/00 0:00
Order385	L385	p3	1/11/00 8:00	2/18/00 0:00
Order386	L386	p3	1/11/00 8:00	2/18/00 0:00

ORDER	LOT	PART	START	DUE
Order391	L391	p3	1/11/00 8:00	2/11/00 0:00
Order392	L392	p2	1/11/00 8:00	2/11/00 0:00
Order393	L393	p2	1/11/00 8:00	2/11/00 0:00
Order394	L394	p2	1/11/00 8:00	2/18/00 0:00
Order395	L395	p2	1/11/00 8:00	2/18/00 0:00
Order396	L396	p2	1/11/00 8:00	2/18/00 0:00
Order397	L397	p2	1/11/00 8:00	2/18/00 0:00
Order398	L398	p4	1/11/00 8:00	2/12/00 0:00
Order399	L399	p4	1/11/00 8:00	2/12/00 0:00
Order400	L400	p2	1/11/00 8:00	2/12/00 0:00
Order401	L401	p5	1/11/00 8:00	1/23/00 0:00
Order402	L402	p5	1/12/00 8:00	2/14/00 0:00
Order403	L403	p5	1/12/00 8:00	2/7/00 0:00
Order404	L404	p5	1/12/00 8:00	1/25/00 0:00
Order405	L405	p5	1/12/00 8:00	1/28/00 0:00
Order406	L406	p5	1/14/00 8:00	2/16/00 0:00
Order407	L407	p5	1/14/00 8:00	2/15/00 0:00
Order408	L408	p3	1/14/00 8:00	2/18/00 0:00
Order409	L409	p4	1/14/00 8:00	1/25/00 0:00
Order410	L410	p5	1/18/00 8:00	1/25/00 0:00
Order411	L411	p5	11/29/99 8:00	12/19/99 0:00
Order412	L412	p2	12/6/99 8:00	12/22/99 0:00
Order413	L413	p5	12/6/99 8:00	12/15/99 0:00
Order414	L414	p5	12/6/99 8:00	12/24/99 0:00
Order415	L415	p5	12/8/99 8:00	1/4/00 0:00
Order416	L416	p5	12/8/99 8:00	1/4/00 0:00
Order417	L417	p5	12/9/99 8:00	1/18/00 0:00
Order418	L418	p2	12/9/99 8:00	1/17/00 0:00
Order419	L419	p2	12/9/99 8:00	1/17/00 0:00
Order420	L420	p2	12/9/99 8:00	1/17/00 0:00
Order421	L421	p2	12/9/99 8:00	1/17/00 0:00
Order422	L422	p2	12/9/99 8:00	1/17/00 0:00
Order423	L423	p2	12/9/99 8:00	1/17/00 0:00
Order424	L424	p4	12/9/99 8:00	1/17/00 0:00
Order425	L425	p4	12/9/99 8:00	1/17/00 0:00
Order426	L426	p4	12/9/99 8:00	1/17/00 0:00
Order427	L427	p3	12/9/99 8:00	1/17/00 0:00
Order428	L428	p3	12/10/99 8:00	12/23/99 0:00
Order429	L429	p5	12/14/99 8:00	1/18/00 0:00
Order430	L430	p5	12/17/99 8:00	1/28/00 0:00
Order431	L431	p2	12/20/99 8:00	1/10/99 0:00
Order432	L432	p5	12/20/99 8:00	1/18/00 0:00
Order433	L433	p3	12/21/99 8:00	1/19/00 0:00
Order434	L434	p6	12/21/99 8:00	1/20/00 0:00
Order435	L435	p6	12/21/99 8:00	1/20/00 0:00

ORDER	LOT	PART	START	DUE
Order440	L440	p3	12/21/99 8:00	2/1/00 0:00
Order441	L441	p4	12/21/99 8:00	1/25/00 0:00
Order442	L442	p3	12/21/99 8:00	1/24/00 0:00
Order443	L443	p3	12/21/99 8:00	1/25/00 0:00
Order444	L444	p3	12/21/99 8:00	2/20/00 0:00
Order445	L445	p3	12/21/99 8:00	2/21/00 0:00
Order446	L446	p3	12/21/99 8:00	2/22/00 0:00
Order447	L447	p3	12/21/99 8:00	2/23/00 0:00
Order448	L448	p5	12/21/99 8:00	1/31/00 0:00
Order449	L449	p2	12/22/99 8:00	1/31/00 0:00
Order450	L450	p1	12/22/99 8:00	2/1/00 0:00
Order451	L451	p1	12/22/99 8:00	2/2/00 0:00
Order452	L452	p1	12/22/99 8:00	2/3/00 0:00
Order453	L453	p6	12/22/99 8:00	1/24/00 0:00
Order454	L454	p6	12/22/99 8:00	1/25/00 0:00
Order455	L455	p6	12/22/99 8:00	1/26/00 0:00

Appendix B: The Stations Worksheet

STNFAM	IGNORE	STN	FWLRANK	RULE
tablehptest	HP Test area Table	tablehptest_1	rank_LBA	rule_FIRST
tabletest	Test area Table	tabletest_1	rank_LBA	rule_FIRST
scopephoto	Photo Microscope	scopephoto_1	rank_LBA	rule_FIRST
		scopephoto_2	rank_LBA	rule_FIRST
scopeproc	Process scope	scopeproc_1	rank_LBA	rule_FIRST
		scopeproc_2	rank_LBA	rule_FIRST
scopemtrls	Materials Microscope	scopemtrls_1	rank_LBA	rule_FIRST
scopetower	tower Inspection microscope	scopetower_1	rank_LBA	rule_FIRST
termphoto	photo terminal	termphoto_1	rank_LBA	rule_FIRST
termproca	process 1 terminal	termproca_1	rank_LBA	rule_FIRST
termprocb	process 2 terminal	termprocb_1	rank_LBA	rule_FIRST
termmtrl	materials terminal	termmtrl_1	rank_LBA	rule_FIRST
termimp	implant terminal	termimp_1	rank_LBA	rule_FIRST
termtower	Ivory tower terminal	termtower_1	rank_LBA	rule_FIRST
	Ivory tower terminal	termtower_2	rank_LBA	rule_FIRST
tapetest	tape test	tapetest_1	rank_LBA	rule_FIRST
impfour	001000	impfour_1	rank_LBA	rule_FIRST
imptwo	001010	imptwo_1	rank_LBA	rule_FIRST
step	002000	step_1	rank_LBA	rule_FIRST
	002010	step_2	rank_LBA	rule_FIRST
suss	002100	suss_1	rank_LBA	rule_FIRST
fusion	002130	fusion_1	rank_LBA	rule_FIRST
	002132	fusion_2	rank_LBA	rule_FIRST
maskcln	002140	maskcln_1	rank_LBA	rule_FIRST
polycoat	004040	polycoat_1	rank_LBA	rule_FIRST
adhsvcoat	004042	adhsvcoat_1	rank_LBA	rule_FIRST
developA	004050	developA_1	rank_LBA	rule_FIRST
	004055	developA_2	rank_LBA	rule_FIRST
developB	004057 backend coat track	developB_1	rank_LBA	rule_FIRST
coat	004060	coat_1	rank_LBA	rule_FIRST
	004070	coat_2	rank_LBA	rule_FIRST
	004073	coat_3	rank_LBA	rule_FIRST
liftoffever	006900 ever	liftoffever_1	rank_LBA	rule_FIRST
liftoffsemi	005000 semi	liftoffsemi_1	rank_LBA	rule_FIRST
branson	005010	branson_1	rank_LBA	rule_FIRST
	005020	branson_2	rank_LBA	rule_FIRST
	005025	branson_3	rank_LBA	rule_FIRST
technics	005030	technics_1	rank_LBA	rule_FIRST
goldplate	005041, 005042	goldplate_1	rank_LBA	rule_FIRST
bonder	005050	bonder_1	rank_LBA	rule_FIRST
	005052	bonder_2	rank_LBA	rule_FIRST
rdA	005070 materials	rdA_1	rank_LBA	rule_FIRST

STNFAM	IGNORE	STN	FWLRANK	RULE
	005100 109mm	rdC_2	rank_LBA	rule_FIRST
wetbenchA	005075	wetbenchA_1	rank_LBA	rule_FIRST
wetbenchB	005076	wetbenchB_1	rank_LBA	rule_FIRST
wetbenchC	005077	wetbenchC_1	rank_LBA	rule_FIRST
sput	006000 perkin elmer	sput_1	rank_LBA	rule_FIRST
	006060 cvc	sput_2	rank_LBA	rule_FIRST
sputB	006080 XM-8 in materials lab	sputB_1	rank_LBA	rule_FIRST
plasmaetchA	006011, 006012	plasmaetchA_1	rank_LBA	rule_FIRST
plasmaetchB	006021, 006022	plasmaetchB_1	rank_LBA	rule_FIRST
evapA	006040 nickel	evapA_1	rank_LBA	rule_FIRST
evapB	006030 tmscal	evapB_1	rank_LBA	rule_FIRST
	006045 cha	evapB_2	rank_LBA	rule_FIRST
depos	006050 pt2411	depos_1	rank_LBA	rule_FIRST
omegaetch	006090	omegaetch_1	rank_LBA	rule_FIRST
	006092	omegaetch_2	rank_LBA	rule_FIRST
alphastepA	007000	alphastepA_1	rank_LBA	rule_FIRST
alphastepB	007002	alphastepB_1	rank_LBA	rule_FIRST
osi	007010	osi_1	rank_LBA	rule_FIRST
thicknessA	007020 ellipsometer	thicknessA_1	rank_LBA	rule_FIRST
thicknessB	007025 ellipsometer	thicknessB_1	rank_LBA	rule_FIRST
	007030 tylan	thicknessB_2	rank_LBA	rule_FIRST
sheetres	007040 4-point probe	sheetres_1	rank_LBA	rule_FIRST
	007045 4-point probe	sheetres_2	rank_LBA	rule_FIRST
wafertransfer	007055 mgi	wafertransfer_1	rank_LBA	rule_FIRST
mgage	007060	mgage_1	rank_LBA	rule_FIRST
lehighton	007085	lehighton_1	rank_LBA	rule_FIRST
rftest	007100	rftest_1	rank_LBA	rule_FIRST
hpink	007110 hp#1 dc test	hpink_1	rank_LBA	rule_FIRST
hptest	007120 hp#2 dctest	hptest_1	rank_LBA	rule_FIRST
digind	007310 digimatic indicator #1	digind_1	rank_LBA	rule_FIRST
	007320 digimatic indicator #2	digind_2	rank_LBA	rule_FIRST
anneal	008010 rta 2106	anneal_1	rank_LBA	rule_FIRST
alloy	008020 heatpulse 2101	alloy_1	rank_LBA	rule_FIRST
prime	008030	prime_1	rank_LBA	rule_FIRST
	008035	prime_2	rank_LBA	rule_FIRST
polybake	008040 polymide oven (yes pb 450)	polybake_1	rank_LBA	rule_FIRST
scribe	008050 waferrmarkII lumonics	scribe_1	rank_LBA	rule_FIRST
adhesionprom	008060 yes-5 vacuum oven	adhesionprom_1	rank_LBA	rule_FIRST
bakematerials	008070 air bake oven in materials	bakematerials_1	rank_LBA	rule_FIRST
bakeproc	008080 Blue M Bake Oven in Process	bakeproc_1	rank_LBA	rule_FIRST
wetbenchgrinder	008092 wetbench at grinder	wetbenchgrinder_1	rank_LBA	rule_FIRST
wetbenchmtrls	008093 wetbench at materilas lab	wetbenchmtrls_1	rank_LBA	rule_FIRST
dicingsaw	009000 dicing saw ma (backup)	dicingsaw_1	rank_LBA	rule_FIRST
	009005 dicing saw k&s	dicingsaw_2	rank_LBA	rule_FIRST
scribebreak	009010	scribebreak_1	rank_LBA	rule_FIRST

Appendix C: The Parts Worksheet

PART	ROUTEFILE	ROUTE	STDLTSZ
p1	route1.txt	route-p1	8
~			
p2	route2.txt	route-p2	8
~			
p3	route3.txt	route-p3	8
~			
p4	route4.txt	route-p4	8
~			
p5	route5.txt	route-p5	8
~			
p6	route6.txt	route-p6	8

Appendix D : Route for product 1

Data Set I

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r1000	a	1	termmtrl	0.848	min	
	b	2	scribe	0.977	min	piece
	c	3	rdA	8.09	min	
	d	4	termmtrl	2.157	min	
~						
r1001	a	6	termmtrl	2	min	
	b	7	termmtrl	2	min	
~						
r1010	a	8	termphoto	1	min	
	b	9	prime	35.41	min	
	c	10	coat	1.45	min	piece
	d	11	step	0.431	min	piece
	e	12	developA	1.828	min	piece
	f	13	rdB	8.576	min	
	g	14	fusion	3.543	min	
	h	15	scopephoto	0.329	min	piece
	i	16	termphoto	0.788	min	
~						
r1020	a	17	termproca	0.736	min	
	b	18	branson	6.895	min	
	c	19	wetbenchC	2.567	min	
	d	20	rdB	4.923	min	
	e	21	scopeproc	0.916	min	
	f	22	branson	35.763	min	
	g	23	wetbenchC	3.91	min	
	h	24	rdB	5.17	min	
	i	25	alphastepA	4.54	min	
	j	26	scopeproc	0.92	min	
	k	27	termproca	1.59	min	
~						
r1810	a	28	termphoto	1	min	
	b	29	prime	35.41	min	
	c	30	coat	1.45	min	piece
	d	31	step	0.431	min	piece
	e	32	developA	1.828	min	piece
	f	33	rdB	8.576	min	
	g	34	fusion	3.543	min	
	h	35	scopephoto	0.329	min	piece
	i	36	termphoto	0.788	min	
~						
r1820	a	37	termimp	0.802	min	
	b	38	imptwo	24	min	
	c	39	impfour	40.456	min	
	d	40	termimp	1.5	min	
~						
r1830	a	41	termproca	2	min	
	b	42	branson	35	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	c	43	rdB	6	min	
	d	44	scopeproc	3	min	
	e	45	termproca	2	min	
~						
r2010	a	46	termphoto	1	min	
	b	47	prime	35.41	min	
	c	48	coat	1.45	min	piece
	d	49	step	0.431	min	piece
	e	50	developA	1.828	min	piece
	f	51	rdB	8.576	min	
	g	52	fusion	3.543	min	
	h	53	scopephoto	0.329	min	piece
	i	54	termphoto	0.788	min	
~						
r2020	a	55	termimp	0.802	min	
	b	56	imptwo	24	min	
	c	57	impfour	40.456	min	
	d	58	termimp	1.5	min	
~						
r2030	a	59	termproca	2	min	
	b	60	branson	35	min	
	c	61	rdB	6	min	
	d	62	scopeproc	3	min	
	e	63	termproca	2	min	
~						
r2400	a	64	termmtrl	0.63	min	
	b	65	wetbenchmtrls	4.58	min	
	c	66	wetbenchmtrls	6.26	min	
	d	67	rdA	7.67	min	
	e	68	sput	52.8	min	
	f	69	scopemtrls	1.7	min	
	g	70	mgage	0.41	min	piece
	h	71	termmtrl	1.7	min	
~						
r2410	a	72	termphoto	0.64	min	
	b	73	wetbenchB	1.58	min	
	c	74	rdB	4.91	min	
	d	75	prime	37	min	
	e	76	coat	2.09	min	piece
	f	77	step	2.13	min	piece
	g	78	developA	2.01	min	piece
	h	79	rdB	9.09	min	
	i	80	scopephoto	3.2	min	
	j	81	termphoto	0.57	min	
~						
r2420	a	82	termproca	1.31	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	e	86	evapA	68.5	min	
	f	87	alphastepA	1.17	min	
	g	88	sheetres	1.39	min	
	h	89	liftoffever	20.1	min	
	i	90	scopeproc	8.19	min	
	j	91	branson	15.2	min	
	k	92	termproca	1.49	min	
~						
r2430	a	93	termproca	0.96	min	
	b	94	osi	8.89	min	
	c	95	termproca	1.26	min	
~						
r2440	a	96	termproca	0.68	min	
	b	97	rdB	16.8	min	
	c	98	omegaetch	11.9	min	piece
	d	99	rdB	8.05	min	
	e	100	scopeproc	2.27	min	
	f	101	termproca	0.57	min	
~						
r3010	a	102	termphoto	1	min	
	b	103	prime	35.41	min	
	c	104	coat	1.45	min	piece
	d	105	step	0.431	min	piece
	e	106	developA	1.828	min	piece
	f	107	rdB	8.576	min	
	g	108	fusion	3.543	min	
	h	109	scopephoto	0.329	min	piece
	i	109a	termphoto	0.788	min	
~						
r3020	a	110	termimp	0.802	min	
	b	111	imptwo	24	min	
	c	112	impfour	40.456	min	
	d	113	termimp	1.5	min	
~						
r3030	a	114	termproca	2	min	
	b	115	branson	35	min	
	c	116	rdB	6	min	
	d	117	scopeproc	3	min	
	e	118	termproca	2	min	
r3340	a	119	termmtrl	1.25	min	
	b	120	wetbenchmtrls	11	min	
	c	121	rdA	8.28	min	
	d	122	scopemtrls	1.98	min	
	e	123	osi	16	min	
	f	124	bakematerials	5.17	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	j	128	termmtrl	2.07	min	
~						
r3410	a	129	termphoto	0.95	min	
	b	130	prime	34.3	min	
	c	131	coat	1.82	min	piece
	d	132	step	1.11	min	piece
	e	133	developA	1.89	min	piece
	f	134	rdB	5.28	min	
	g	135	fusion	3.53	min	piece
	h	136	scopephoto	1.07	min	
	i	137	termphoto	0.76	min	
~						
r3420	a	138	termimp	0.802	min	
	b	139	imptwo	24	min	
	c	140	impfour	40.456	min	
	d	141	termimp	1.5	min	
~						
r3430	a	142	termproca	2	min	
	b	143	branson	35.594	min	
	c	144	rdB	14.139	min	
	d	145	scopeproc	3.93	min	
	e	146	termproca	1.585	min	
~						
r3500	a	147	termmtrl	1.25	min	
	b	148	wetbenchmtrls	11	min	
	c	149	rdA	8.28	min	
	d	150	bakematerials	5.17	min	
	e	151	depos	14.6	min	
	f	152	thicknessB	1.32	min	
	g	153	scopemtrls	1.98	min	
	h	154	termmtrl	2.07	min	
~						
r3510	a	155	termphoto	3	min	
	b	156	coat	1.373	min	piece
	c	157	coat	1.718	min	piece
	d	158	developA	2.918	min	piece
	e	159	suss	2.918	min	piece
	f	160	fusion	6.493	min	
	g	161	scopeproc	6.126	min	
	h	162	termphoto	0.172	min	
~						
r3520	a	163	termproca	0.96	min	
	b	164	plasmaetchA	76.7	min	
	c	165	thicknessB	1.04	min	piece
	d	166	scopeproc	1.12	min	
	e	167	alphastepA	1.56	min	piece

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r4010	a	170	termphoto	0.54	min	
	b	171	prime	37.4	min	
	c	172	coat	1.71	min	piece
	d	173	step	1.94	min	piece
	e	174	developA	1.89	min	piece
	f	175	rdB	5	min	
	g	176	scopephoto	10.5	min	
	h	177	termphoto	2.06	min	
~						
r4020	a	178	termproca	0.72	min	
	b	179	plasmaetchA	18	min	
	c	180	scopeproc	1.12	min	
	d	181	termproca	1.01	min	
~						
r4030	a	182	termproca	0.92	min	
	b	183	wetbenchC	3.88	min	
	c	184	rdB	4.98	min	
	d	185	evapB	76.3	min	
	e	186	alphastepA	1.4	min	
	f	187	sheetres	1.4	min	
	g	188	liftoffsemi	34.5	min	
	h	189	scopeproc	2.48	min	
	i	190	branson	32.3	min	
	j	191	termproca	2.06	min	
~						
r4040	a	192	termproca	0.26	min	
	b	193	alloy	5.691	min	piece
	c	194	scopeproc	24.7	min	
	d	195	termproca	1.57	min	
~						
r4350	a	196	termproca	2	min	
	b	197	hptest	100	min	
	c	198	termproca	2	min	
~						
r4400	a	199	termproca	2	min	
	b	200	tabletest	30	min	
	c	201	termproca	2	min	
~						
r5010	a	202	termphoto	2	min	
	b	203	wetbenchB	1.58	min	
	c	204	rdB	4.91	min	
	d	205	prime	37	min	
	e	206	coat	2.09	min	piece
	f	207	step	2.13	min	piece
	g	208	developA	2.01	min	piece
	h	209	wetbenchC	4	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r5020	a	213	termproca	1.27	min	
	b	214	branson	6.53	min	
	c	215	evapB	55.2	min	
	d	216	alphastepA	1.03	min	
	e	217	sheetres	1.06	min	
	f	218	liftoffever	20.1	min	
	g	219	rdB	6	min	
	h	220	scopeproc	7.62	min	
	i	221	branson	15.8	min	
	j	222	termproca	3.84	min	
~						
r6100	a	223	termmtrl	0.34	min	
	b	224	depos	33.5	min	
	c	225	lehighton	1.73	min	
	d	226	scopemtrls	2.19	min	
	e	227	termmtrl	1.11	min	
~						
r6110	a	228	termphoto	0.95	min	
	b	229	prime	34.3	min	
	c	230	coat	1.82	min	piece
	d	231	step	1.11	min	piece
	e	232	developA	1.89	min	piece
	f	233	rdB	5.28	min	
	g	234	fusion	3.53	min	piece
	h	235	scopephoto	1.07	min	
	i	236	termphoto	0.76	min	
~						
r6120	a	237	termproca	0.91	min	
	b	238	plasmaetchB	78.7	min	
	c	239	branson	35.5	min	
	d	240	liftoffever	20.1	min	
	e	241	rdB	14.6	min	
	f	242	scopeproc	3.84	min	piece
	g	243	termproca	1.22	min	
~						
r7900	a	244	termphoto	0.77	min	
	b	245	adhesionprom	62.8	min	
	c	246	polycoat	2.69	min	piece
	d	247	scopephoto	5	min	piece
	e	248	polybake	363	min	
	f	249	termphoto	0.89	min	
~						
r7910	a	250	termphoto	0.85	min	
	b	251	coat	2.23	min	piece
	c	252	step	5.29	min	piece
	d	253	developA	3.83	min	piece

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
~						
r7920	a	257	termproca	1.67	min	
	b	258	plasmaetchB	87.8	min	
	c	259	scopeproc	14.5	min	
	d	260	liftoffever	20.1	min	
	e	261	rdB	1.33	min	
	f	262	scopeproc	3.81	min	
	g	263	termproca	1.81	min	
~					min	
r8020	a	264	termproca	0.75	min	
	b	265	branson	4.74	min	
	c	266	sput	50.6	min	
	d	267	evapB	35.6	min	
	e	268	alphastepA	4.15	min	
	f	269	sheetres	3.16	min	
	g	270	termproca	1.12	min	
~						
r8110	a	271	termphoto	0.71	min	
	b	272	prime	38.2	min	
	c	273	coat	2.28	min	piece
	d	274	step	3.91	min	piece
	e	275	step	0	min	piece
	f	276	developA	2.19	min	piece
	g	277	rdB	5.34	min	
	h	278	coat	2.25	min	piece
	i	279	scopephoto	5.98	min	
	j	280	termphoto	1.29	min	
~						
r8120	a	281	termproca	0.66	min	
	b	282	branson	11.4	min	
	c	283	alphastepA	1.26	min	
	d	284	goldplate	82	min	
	e	285	rdB	9.71	min	
	f	286	alphastepA	0.64	min	
	g	287	termproca	0.98	min	
~						
r8130	a	288	termproca	0.6	min	
	b	289	liftoffsemi	4.41	min	
	c	290	technics	44	min	
	d	291	wetbenchB	4.42	min	
	e	292	wetbenchB	2.25	min	
	f	293	wetbenchB	4.36	min	
	g	294	rdB	12.6	min	
	h	295	scopeproc	10.3	min	
	i	296	alphastepA	1.04	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	c	300	polycoat	2.91	min	piece
	d	301	polycoat	0	min	
	e	302	scopephoto	2	min	
	f	303	polybake	363	min	
	g	304	termphoto	1.164	min	
~						
r8210	a	305	termphoto	0.71	min	
	b	306	coat	2.71	min	piece
	c	307	coat	2.68	min	piece
	d	308	suss	2.46	min	piece
	e	309	developA	30.6	min	piece
	f	310	rdB	5.58	min	
	g	311	scopephoto	0.942	min	piece
	h	312	termphoto	1.23	min	
~						
r8220	a	313	termproca	1.56	min	
	b	314	plasmaetchB	173	min	
	c	315	scopeproc	1.84	min	
	d	316	liftoffever	27.8	min	
	e	317	rdB	5.1	min	
	f	318	branson	5.72	min	
	g	319	scopeproc	1.96	min	piece
	h	320	termproca	0.87	min	
~						
r8350	a	321	termproca	2	min	
	b	322	hptest	10	min	piece
	c	323	termproca	2	min	
~						
r8400	a	324	termproca	2	min	
	b	325	tablehptest	12	min	
	c	326	termproca	2	min	
~					min	
r9000	a	327	termmtrl	0.64	min	
	b	328	scopeproc	1.32	min	piece
	c	329	rdB	5	min	
	d	330	adhsvcoat	1.97	min	piece
	e	331	bonder	167	min	
	f	332	termmtrl	0.76	min	
~						
r9020	a	333	termmtrl	0.17	min	
	b	334	grinder	73	min	
	c	335	digind	2.26	min	piece
	d	336	wetbenchgrinder	18.1	min	
	e	337	digind	0.57	min	piece
	f	338	scopeproc	2	min	
	g	339	termmtrl	2.01	min	
~						

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	c	342	polycoat	1.5	min	piece
	d	343	rdC	7	min	
	e	344	polycoat	1.5	min	piece
	f	345	suss	25	sec	piece
	g	346	developB	2.5	min	piece
	h	347	wetbenchC	10	min	
	i	348	rdC	7	min	
	j	349	scopephoto	3	min	piece
	k	350	scopephoto	1	min	piece
	l	351	termphoto	2	min	
~						
r9130	a	352	termphoto	2	min	
	b	353	polycoat	1.5	min	
	c	354	suss	25	sec	
	d	355	plasmaetchA	18	min	
	e	356	scopeproc	2	min	
	f	357	wetbenchC	10	min	
	g	358	rdC	7	min	
	h	359	scopeproc	2	min	
	i	360	termphoto	2	min	
~					min	
r9140	a	361	termproca	2	min	
	b	362	branson	7.5	min	
	c	363	wetbenchC	10	min	
	d	364	rdC	7	min	
	e	365	sputB	10	min	
	f	366	goldplate	82	min	
	g	367	rdC	7	min	
	h	368	evapB	35.6	min	
	i	369	sheetres	10	min	
	j	370	sputB	10	min	
	k	371	tapetest	10	min	
	m	372	termproca	2	min	
~						
r9142	a	373	termphoto	2	min	
	b	374	polycoat	10	min	
	c	375	suss	25	min	
	d	376	developB	2.5	min	
	e	377	scopephoto	10	min	
	f	378	termphoto	2	min	
~						
r9144	a	379	termproca	2	min	
	b	380	plasmaetchA	18	min	
	c	381	scopeproc	2	min	
	d	382	wetbenchC	10	min	
	e	383	rdC	7	min	
	f	384	liftoffsemi	20	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r9160	a	386	termmtrl	0.76	min	
	b	387	demount	150	min	
	c	388	digind	0.74	min	piece
	d	389	technics	33.5	min	
	e	390	technics	26.2	min	
	f	391	termmtrl	3.2	min	
~						
r9250	a	392	termtower	0.76	min	
	b	393	rftest	4.05	min	piece
	c	394	termtower	1.27	min	
~						
r9260	a	395	termtower	0.28	min	
	b	396	termtower	14.83	min	
~						
r9300	a	397	termtower	0.8	min	
	b	398	scopetower	6.21	min	piece
	c	399	termtower	0.72	min	
~						
r9999	a	400	termtower	13	min	

Data Set II

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
r2400	a	64	termmtrl	0	min
	b	65	wetbenchmtrls	15.96	min
	d	67	rdA	7.67	min
	e	68	sput	60.6	min
	f	69	scopemtrls	4.5	min
	g	70	mgage	4.73	min
	h	71	termmtrl	0	min
~					
r2410	a	72	termphoto	0	min
	b	73	wetbenchB	11.7	min
	c	74	rdB	4.91	min
	d	75	prime	43.2	min
	e	76	coat	16.9	min
	f	77	step	16.4	min
	g	78	developA	16.6	min
	h	79	rdB	9.8	min
	i	80	scopephoto	1	min
	j	81	termphoto	0	min
~					
r2420	a	82	termproca	1.31	min
	b	83	plasmaetchA	15.1	min
	c	84	wetbenchC	3.66	min
	d	85	rdB	5.14	min
	e	86	evapA	68.5	min
	f	87	alphastepA	1.17	min
	g	88	sheetres	1.39	min
	h	89	liftoffever	20.1	min
	i	90	scopeproc	8.19	min
	j	91	branson	15.2	min
	k	92	termproca	1.49	min
~					
r3520	a	163	termproca	0	min
	b	164	plasmaetchA	69.63	min
	c	165	thicknessB	11.2	min
	d	166	scopeproc	9.86	min
	e	167	alphastepA	15.3	min
	f	168	branson	8.4	min
	g	169	termproca	0	min
~					
r4010	a	170	termphoto	0	min
	b	171	prime	42.5	min
	c	172	coat	17.15	min
	d	173	step	49.7	min
	e	174	developA	3.95	min
	f	175	rdB	15.75	min
	g	176	scopephoto	10.35	min

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
	h	177	termphoto	0	min
~					
r4020	a	178	termproca	0	min
	b	179	plasmaetchA	68.5	min
	c	180	scopeproc	3.2	min
	d	181	termproca	0	min
~					
r4040	a	192	termproca	0	min
	b	193	alloy	68.5	min
	c	194	scopeproc	3.2	min
	d	195	termproca	0	min
~					
r6120	a	237	termproca	0	min
	b	238	plasmaetchB	43.83	min
	c	239	branson	30.2	min
	d	240	liftoffever	20.93	min
	e	241	rdB	12.7	min
	f	242	scopeproc	3.4	min
	g	243	termproca	0	min

Appendix E: Route for product 2

Data Set I for product 2

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r1000	a	1	termmtrl	0.848	min	
	b	2	scribe	0.977	min	piece
	c	3	rdA	8.09	min	
	d	4	termmtrl	2.157	min	
~						
r1001	a	6	termmtrl	2	min	
	b	7	termmtrl	2	min	
~						
r1010	a	8	termphoto	1	min	
	b	9	prime	35.41	min	
	c	10	coat	1.45	min	piece
	d	11	step	0.431	min	piece
	e	12	developA	1.828	min	piece
	f	13	rdB	8.576	min	
	g	14	fusion	3.543	min	
	h	15	scopephoto	0.329	min	piece
	i	16	termphoto	0.788	min	
~						
r1020	a	17	termproca	0.736	min	
	b	18	branson	6.895	min	
	c	19	wetbenchC	2.567	min	
	d	20	rdB	4.923	min	
	e	21	scopeproc	0.916	min	
	f	22	branson	35.763	min	
	g	23	wetbenchC	3.91	min	
	h	24	rdB	5.17	min	
	i	25	alphastepA	4.54	min	
	j	26	scopeproc	0.92	min	
	k	27	termproca	1.59	min	
~						
r1325	a	28	termimp	0.802	min	
	b	29	imptwo	24	min	
	c	30	impfour	40.456	min	
	d	31	termimp	1.5	min	
~						
r2400	a	32	termmtrl	0.63	min	
	b	33	wetbenchmtrls	4.58	min	
	c	34	wetbenchmtrls	6.26	min	
	d	35	rdA	7.67	min	
	e	36	sput	52.8	min	
	f	37	scopemtrls	1.7	min	
	g	38	mgage	0.41	min	piece
	h	39	termmtrl	1.7	min	
~						
r2410	a	40	termphoto	0.64	min	
	b	41	wetbenchB	1.58	min	
	c	42	rdB	4.91	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	43	prime	37	min	
	e	44	coat	2.09	min	piece
	f	45	step	2.13	min	piece
	g	46	developA	2.01	min	piece
	h	47	rdB	9.09	min	
	i	48	scopephoto	3.2	min	
	j	49	termphoto	0.57	min	
~						
r2420	a	50	termproca	1.31	min	
	b	51	plasmaetchA	15.1	min	
	c	52	wetbenchC	3.66	min	
	d	53	rdB	5.14	min	
	e	54	evapA	68.5	min	
	f	55	alphastepA	1.17	min	
	g	56	sheetres	1.39	min	
	h	57	liftoffever	20.1	min	
	i	58	scopeproc	8.19	min	
	j	59	branson	15.2	min	
	k	60	termproca	1.49	min	
~						
r2430	a	61	termproca	0.96	min	
	b	62	osi	8.89	min	
	c	63	termproca	1.26	min	
~						
r2440	a	64	termproca	0.68	min	
	b	65	rdB	16.8	min	
	c	66	omegaetch	11.9	min	piece
	d	67	rdB	8.05	min	
	e	68	scopeproc	2.27	min	
	f	69	termproca	0.57	min	
~						
r3020	a	70	termimp	0.802	min	
	b	71	imptwo	24	min	
	c	72	impfour	40.456	min	
	d	73	termimp	1.5	min	
~						
r3340	a	74	termmtrl	1.25	min	
	b	75	wetbenchmtrls	11	min	
	c	76	rdA	8.28	min	
	d	77	scopemtrls	1.98	min	
	e	78	osi	16	min	
	f	79	bakematerials	5.17	min	
	g	80	depos	14.6	min	
	h	81	thicknessB	1.32	min	
	i	82	anneal	60	min	
	j	83	termmtrl	2.07	min	
~						
r3410	a	84	termphoto	0.95	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	85	prime	34.3	min	
	c	86	coat	1.82	min	piece
	d	87	step	1.11	min	piece
	e	88	developA	1.89	min	piece
	f	89	rdB	5.28	min	
	g	90	fusion	3.53	min	piece
	h	91	scopephoto	1.07	min	
	i	92	termphoto	0.76	min	
~						
r3420	a	93	termimp	0.802	min	
	b	94	imptwo	24	min	
	c	95	impfour	40.456	min	
	d	96	termimp	1.5	min	
~						
r3430	a	97	termproca	2	min	
	b	98	branson	35.594	min	
	c	99	rdB	14.139	min	
	d	100	scopeproc	3.93	min	
	e	101	termproca	1.585	min	
~						
r4010	a	102	termphoto	0.54	min	
	b	103	prime	37.4	min	
	c	104	coat	1.71	min	piece
	d	105	step	1.94	min	piece
	e	106	developA	1.89	min	piece
	f	107	rdB	5	min	
	g	108	scopephoto	10.5	min	
	h	109	termphoto	2.06	min	
~						
r4020	a	110	termproca	0.72	min	
	b	111	plasmaetchA	18	min	
	c	112	scopeproc	1.12	min	
	d	113	termproca	1.01	min	
~						
r4030	a	114	termproca	0.92	min	
	b	115	wetbenchC	3.88	min	
	c	116	rdB	4.98	min	
	d	117	evapB	76.3	min	
	e	118	alphastepA	1.4	min	
	f	119	sheetres	1.4	min	
	g	120	liftoffsemi	34.5	min	
	h	121	scopeproc	2.48	min	
	i	122	branson	32.3	min	
	j	123	termproca	2.06	min	
~						
r4040	a	124	termproca	0.26	min	
	b	125	alloy	5.691	min	piece
	c	126	scopeproc	24.7	min	
	d	127	termproca	1.57	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
~						
r7210	a	128	termphoto	0.64	min	
	b	129	wetbenchB	1.58	min	
	c	130	rdB	4.91	min	
	d	131	prime	37	min	
	e	132	coat	2.09	min	piece
	f	133	step	2.13	min	piece
	g	134	developA	2.01	min	piece
	h	135	wetbenchC	1.25	min	
	i	136	rdB	9.09	min	
	j	137	scopephoto	3.2	min	
	k	138	termphoto	0.57	min	
~						
r7220	a	139	termproca	1.27	min	
	b	140	branson	6	min	
	c	141	evapB	55.2	min	
	d	142	alphastepA	1.03	min	
	e	143	sheetres	1.06	min	
	f	144	liftoffever	20.1	min	
	g	145	rdB	6	min	
	h	146	scopeproc	5	min	
	i	147	branson	15.8	min	
	j	148	termproca	3.84	min	
~						
r7500	a	149	termproca	2	min	
	b	150	depos	14	min	
	c	151	thicknessB	3	min	
	d	152	termproca	2	min	
~					min	
r7510	a	153	termphoto	0.64	min	
	b	154	wetbenchB	1.58	min	
	c	155	rdB	4.91	min	
	d	156	prime	37	min	
	e	157	coat	2.09	min	piece
	f	158	step	2.13	min	piece
	g	159	developA	2.01	min	piece
	h	160	developB	2	min	piece
	i	161	rdB	9.09	min	
	j	162	coat	2.09	min	piece
	k	163	scopephoto	3.2	min	
	l	164	termphoto	0.57	min	
~						
r7520	a	165	termproca	2	min	
	b	166	branson	6	min	
	c	167	wetbenchC	6	min	
	d	168	rdB	7	min	
	e	169	scopeproc	2	min	
	f	170	branson	35	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	h	172	scopeproc	2	min	
	i	173	bakeproc	15	min	
	j	174	termproca	2	min	
~						
r8350	a	175	termproca	2	min	
	b	176	hptest	10	min	piece
	c	177	termproca	2	min	
~						
r8400	a	178	termproca	2	min	
	b	179	tablehptest	12	min	
	c	180	termproca	2	min	
~					min	
r9000	a	181	termmtrl	0.64	min	
	b	182	scopeproc	1.32	min	piece
	c	183	rdB	5	min	
	d	184	adhsvcoat	1.97	min	piece
	e	185	bonder	167	min	
	f	186	termmtrl	0.76	min	
~						
r9020	a	187	termmtrl	0.17	min	
	b	188	grinder	73	min	
	c	189	digind	2.26	min	piece
	d	190	wetbenchgrinder	18.1	min	
	e	191	digind	0.57	min	piece
	f	192	scopeproc	2	min	
	g	193	termmtrl	2.01	min	
~						
r9160	a	194	termmtrl	0.76	min	
	b	195	demount	150	min	
	c	196	digind	0.74	min	piece
	d	197	technics	33.5	min	
	e	198	technics	26.2	min	
	f	199	termmtrl	3.2	min	
~						
r9300	a	200	termtower	0.8	min	
	b	201	scopetower	6.21	min	piece
	c	202	termtower	0.72	min	
~						
r9999	a	203	termtower	13	min	

Data Set II for product 2

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
r2400	a	32	termmtrl	0	min
	b	33	wetbenchmtrls	15.96	min
	d	35	rdA	7.67	min
	e	36	sput	60.6	min
	f	37	scopemtrls	4.5	min
	g	38	mgage	4.73	min
	h	39	termmtrl	0	min
~					
r2410	a	40	termphoto	0	min
	b	41	wetbenchB	11.7	min
	c	42	rdB	4.91	min
	d	43	prime	43.2	min
	e	44	coat	16.9	min
	f	45	step	16.4	min
	g	46	developA	16.6	min
	h	47	rdB	9.8	min
	i	48	scopephoto	1	min
	j	49	termphoto	0	min
~					
r2420	a	50	termproca	1.31	min
	b	51	plasmaetchA	15.1	min
	c	52	wetbenchC	3.66	min
	d	53	rdB	5.14	min
	e	54	evapA	68.5	min
	f	55	alphastepA	1.17	min
	g	56	sheetres	1.39	min
	h	57	liftoffever	20.1	min
	i	58	scopeproc	8.19	min
	j	59	branson	15.2	min
	k	60	termproca	1.49	min
~					
r4010	a	102	termphoto	0	min
	b	103	prime	42.5	min
	c	104	coat	17.15	min
	d	105	step	49.7	min
	e	106	developA	3.95	min
	f	107	rdB	15.75	min
	g	108	scopephoto	10.35	min
	h	109	termphoto	0	min
~					
r4020	a	110	termproca	0	min
	b	111	plasmaetchA	31.33	min
	c	112	scopeproc	6.33	min
	d	113	termproca	0	min
~					
r4040	a	124	termproca	0	min
	b	125	alloy	68.5	min
	c	126	scopeproc	3.2	min

Appendix F: Route for product 3

Data Set II

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r1000	a	1	termmtrl	0.848	min	
	b	2	scribe	0.977	min	piece
	c	3	rdA	8.09	min	
	d	4	termmtrl	2.157	min	
~						
r1001	a	6	termmtrl	2	min	
	b	7	termmtrl	2	min	
~						
r1010	a	8	termphoto	1	min	
	b	9	prime	35.41	min	
	c	10	coat	1.45	min	piece
	d	11	step	0.431	min	piece
	e	12	developA	1.828	min	piece
	f	13	rdB	8.576	min	
	g	14	fusion	3.543	min	
	h	15	scopephoto	0.329	min	piece
	i	16	termphoto	0.788	min	
~						
r1020	a	17	termproca	0.736	min	
	b	18	branson	6.895	min	
	c	19	wetbenchC	2.567	min	
	d	20	rdB	4.923	min	
	e	21	scopeproc	0.916	min	
	f	22	branson	35.76	min	
	g	23	wetbenchC	3.91	min	
	h	24	rdB	5.17	min	
	i	25	alphastepA	4.54	min	
	j	26	scopeproc	0.92	min	
	k	27	termproca	1.59	min	
~						
r1325	a	28	termimp	0.802	min	
	b	29	imptwo	24	min	
	c	30	impfour	40.46	min	
	d	31	termimp	1.5	min	
~						
r2400	a	32	termmtrl	0.63	min	
	b	33	wetbenchmtrls	4.58	min	
	c	34	wetbenchmtrls	6.26	min	
	d	35	rdA	7.67	min	
	e	36	sput	52.8	min	
	f	37	scopemtrls	1.7	min	
	g	38	mgage	0.41	min	piece
	h	39	termmtrl	1.7	min	
~						
r2410	a	40	termphoto	0.64	min	
	b	41	wetbenchB	1.58	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	43	prime	37	min	
	e	44	coat	2.09	min	piece
	f	45	step	2.13	min	piece
	g	46	developA	2.01	min	piece
	h	47	rdB	9.09	min	
	i	48	scopephoto	3.2	min	
	j	49	termphoto	0.57	min	
~						
r2420	a	50	termproca	1.31	min	
	b	51	plasmaetchA	15.1	min	
	c	52	wetbenchC	3.66	min	
	d	53	rdB	5.14	min	
	e	54	evapA	68.5	min	
	f	55	alphastepA	1.17	min	
	g	56	sheetres	1.39	min	
	h	57	liftoffever	20.1	min	
	i	58	scopeproc	8.19	min	
	j	59	branson	15.2	min	
	k	60	termproca	1.49	min	
~						
r2430	a	61	termproca	0.96	min	
	b	62	osi	8.89	min	
	c	63	termproca	1.26	min	
~						
r2440	a	64	termproca	0.68	min	
	b	65	rdB	16.8	min	
	c	66	omegaetch	11.9	min	piece
	d	67	rdB	8.05	min	
	e	68	scopeproc	2.27	min	
	f	69	termproca	0.57	min	
~						
r3020	a	70	termimp	0.802	min	
	b	71	imptwo	24	min	
	c	72	impfour	40.46	min	
	d	73	termimp	1.5	min	
~						
r3340	a	74	termmtrl	1.25	min	
	b	75	wetbenchmtrls	11	min	
	c	76	rdA	8.28	min	
	d	77	scopemtrls	1.98	min	
	e	78	osi	16	min	
	f	79	bakematerials	5.17	min	
	g	80	depos	14.6	min	
	h	81	thicknessB	1.32	min	
	i	82	anneal	60	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	85	prime	34.3	min	
	c	86	coat	1.82	min	piece
	d	87	step	1.11	min	piece
	e	88	developA	1.89	min	piece
	f	89	rdB	5.28	min	
	g	90	fusion	3.53	min	piece
	h	91	scopephoto	1.07	min	
	i	92	termphoto	0.76	min	
~						
r3420	a	93	termimp	0.802	min	
	b	94	imptwo	24	min	
	c	95	impfour	40.456	min	
	d	96	termimp	1.5	min	
~						
r3430	a	97	termproca	2	min	
	b	98	branson	35.594	min	
	c	99	rdB	14.139	min	
	d	100	scopeproc	3.93	min	
	e	101	termproca	1.585	min	
~						
r3510	a	102	termphoto	3	min	
	b	103	coat	1.373	min	piece
	c	104	coat	1.718	min	piece
	d	105	developA	2.918	min	piece
	e	106	suss	2.918	min	piece
	f	107	fusion	6.493	min	
	g	108	scopeproc	6.126	min	
	h	109	termphoto	0.172	min	
~						
r3520	a	110	termproca	0.96	min	
	b	111	plasmaetchA	76.7	min	
	c	112	thicknessB	1.04	min	piece
	d	113	scopeproc	1.12	min	
	e	114	alphastepA	1.56	min	piece
	f	115	branson	10.6	min	
	g	116	termproca	2.05	min	
~						
r4010	a	117	termphoto	0.54	min	
	b	118	prime	37.4	min	
	c	119	coat	1.71	min	piece
	d	120	step	1.94	min	piece
	e	121	developA	1.89	min	piece
	f	122	rdB	5	min	
	g	123	scopephoto	10.5	min	
	h	124	termphoto	2.06	min	
~						
r4020	a	125	termproca	0.72	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	128	termproca	1.01	min	
~						
r4030	a	129	termproca	0.92	min	
	b	130	wetbenchC	3.88	min	
	c	131	rdB	4.98	min	
	d	132	evapB	76.3	min	
	e	133	alphastepA	1.4	min	
	f	134	sheetres	1.4	min	
	g	135	liftoffsemi	34.5	min	
	h	136	scopeproc	2.48	min	
	i	137	branson	32.3	min	
	j	138	termproca	2.06	min	
~						
r4040	a	139	termproca	0.26	min	
	b	140	alloy	5.691	min	piece
	c	141	scopeproc	24.7	min	
	d	142	termproca	1.57	min	
~						
r5010	a	143	termphoto	2	min	
	b	144	wetbenchB	1.58	min	
	c	145	rdB	4.91	min	
	d	146	prime	37	min	
	e	147	coat	2.09	min	piece
	f	148	step	2.13	min	piece
	g	149	developA	2.01	min	piece
	h	150	wetbenchC	4	min	
	i	151	rdB	9.09	min	
	j	152	scopephoto	0.4	min	piece
	k	153	termphoto	0.57	min	
~						
r5020	a	154	termproca	1.27	min	
	b	155	branson	6.53	min	
	c	156	evapB	55.2	min	
	d	157	alphastepA	1.03	min	
	e	158	sheetres	1.06	min	
	f	159	liftoffever	20.1	min	
	g	160	rdB	6	min	
	h	161	scopeproc	7.62	min	
	i	162	branson	15.8	min	
	j	163	termproca	3.84	min	
~						
~						
r6100	a	164	termmtrl	0.34	min	
	b	165	depos	33.5	min	
	c	166	lehighton	1.73	min	
	d	167	scopemtrls	2.19	min	
	e	168	termmtrl	1.11	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	170	prime	34.3	min	
	c	171	coat	1.82	min	piece
	d	172	step	1.11	min	piece
	e	173	developA	1.89	min	piece
	f	174	rdB	5.28	min	
	g	175	fusion	3.53	min	piece
	h	176	scopephoto	1.07	min	
	i	177	termphoto	0.76	min	
~						
r6120	a	178	termproca	0.91	min	
	b	179	plasmaetchB	78.7	min	
	c	180	branson	35.5	min	
	d	181	liftoffever	20.1	min	
	e	182	rdB	14.6	min	
	f	183	scopeproc	3.84	min	piece
	g	184	termproca	1.22	min	
~						
r7900	a	185	termphoto	0.77	min	
	b	186	adhesionprom	62.8	min	
	c	187	polycoat	2.69	min	piece
	d	188	scopephoto	5	min	piece
	e	189	polybake	363	min	
	f	190	termphoto	0.89	min	
~						
r7910	a	191	termphoto	0.85	min	
	b	192	coat	2.23	min	piece
	c	193	step	5.29	min	piece
	d	194	developA	3.83	min	piece
	e	195	rdB	5.22	min	
	f	196	scopephoto	12.9	min	
	g	197	termphoto	0.17	min	
~						
r7920	a	198	termproca	1.67	min	
	b	199	plasmaetchB	87.8	min	
	c	200	scopeproc	14.5	min	
	d	201	liftoffever	20.1	min	
	e	202	rdB	1.33	min	
	f	203	scopeproc	3.81	min	
	g	204	termproca	1.81	min	
~					min	
r8020	a	205	termproca	0.75	min	
	b	206	branson	4.74	min	
	c	207	sput	50.6	min	
	d	208	evapB	35.6	min	
	e	209	alphastepA	4.15	min	
	f	210	sheetres	3.16	min	
	g	211	termproca	1.12	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	213	prime	38.2	min	
	c	214	coat	2.28	min	piece
	d	215	step	3.91	min	piece
	e	216	step	0	min	piece
	f	217	developA	2.19	min	piece
	g	218	rdB	5.34	min	
	h	219	coat	2.25	min	piece
	i	220	scopephoto	5.98	min	
	j	221	termphoto	1.29	min	
~						
r8120	a	222	termproca	0.66	min	
	b	223	branson	11.4	min	
	c	224	alphastepA	1.26	min	
	d	225	goldplate	82	min	
	e	226	rdB	9.71	min	
	f	227	alphastepA	0.64	min	
	g	228	termproca	0.98	min	
~						
r8130	a	229	termproca	0.6	min	
	b	230	liftoffsemi	4.41	min	
	c	231	technics	44	min	
	d	232	wetbenchB	4.42	min	
	e	233	wetbenchB	2.25	min	
	f	234	wetbenchB	4.36	min	
	g	235	rdB	12.6	min	
	h	236	scopeproc	10.3	min	
	i	237	alphastepA	1.04	min	
	j	238	termproca	0.23	min	
~						
r8200	a	239	termphoto	0.78	min	
	b	240	adhesionprom	62.8	min	
	c	241	polycoat	2.91	min	piece
	d	242	polycoat	0	min	
	e	243	scopephoto	2	min	
	f	244	polybake	363	min	
	g	245	termphoto	1.164	min	
~						
r8210	a	246	termphoto	0.71	min	
	b	247	coat	2.71	min	piece
	c	248	coat	2.68	min	piece
	d	249	suss	2.46	min	piece
	e	250	developA	30.6	min	piece
	f	251	rdB	5.58	min	
	g	252	scopephoto	0.942	min	piece
	h	253	termphoto	1.23	min	
~						

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	257	liftoffever	27.8	min	
	e	258	rdB	5.1	min	
	f	259	branson	5.72	min	
	g	260	scopeproc	1.96	min	piece
	h	261	termproca	0.87	min	
~						
r8350	a	262	termproca	2	min	
	b	263	hptest	10	min	piece
	c	264	termproca	2	min	
~						
r8400	a	265	termproca	2	min	
	b	266	tablehptest	12	min	
	c	267	termproca	2	min	
~					min	
r9000	a	268	termmtrl	0.64	min	
	b	269	scopeproc	1.32	min	piece
	c	270	rdB	5	min	
	d	271	adhsvcoat	1.97	min	piece
	e	272	bonder	167	min	
	f	273	termmtrl	0.76	min	
~						
r9020	a	274	termmtrl	0.17	min	
	b	275	grinder	73	min	
	c	276	digind	2.26	min	piece
	d	277	wetbenchgrinder	18.1	min	
	e	278	digind	0.57	min	piece
	f	279	scopeproc	2	min	
	g	280	termmtrl	2.01	min	
~						
r9160	a	281	termmtrl	0.76	min	
	b	282	demount	150	min	
	c	283	digind	0.74	min	piece
	d	284	technics	33.5	min	
	e	285	technics	26.2	min	
	f	286	termmtrl	3.2	min	
~						
r9250	a	287	termtower	0.76	min	
	b	288	rftest	4.05	min	piece
	c	289	termtower	1.27	min	
~						
r9260	a	290	termtower	0.28	min	
	b	291	termtower	14.83	min	
~						
r9300	a	292	termtower	0.8	min	

Data Set II for product 3

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
r2400	a	32	termmtrl	0	min
	b	33	wetbenchmtrls	15.96	min
	d	35	rdA	7.67	min
	e	36	sput	60.6	min
	f	37	scopemtrls	4.5	min
	g	38	mgage	4.73	min
	h	39	termmtrl	0	min
~					
r2410	a	40	termphoto	0	min
	b	41	wetbenchB	11.7	min
	c	42	rdB	4.91	min
	d	43	prime	43.2	min
	e	44	coat	16.9	min
	f	45	step	16.4	min
	g	46	developA	16.6	min
	h	47	rdB	9.8	min
	i	48	scopephoto	1	min
	j	49	termphoto	0	min
~					
r2420	a	50	termproca	1.31	min
	b	51	plasmaetchA	15.1	min
	c	52	wetbenchC	3.66	min
	d	53	rdB	5.14	min
	e	54	evapA	68.5	min
	f	55	alphastepA	1.17	min
	g	56	sheetres	1.39	min
	h	57	liftoffever	20.1	min
	i	58	scopeproc	8.19	min
	j	59	branson	15.2	min
	k	60	termproca	1.49	min
~					
r3520	a	110	termproca	0	min
	b	111	plasmaetchA	69.63	min
	c	112	thicknessB	11.2	min
	d	113	scopeproc	9.86	min
	e	114	alphastepA	15.3	min
	f	115	branson	8.4	min
	g	116	termproca	0	min
~					
r4010	a	102	termphoto	0	min
	b	103	prime	42.5	min
	c	104	coat	17.15	min
	d	105	step	49.7	min
	e	106	developA	3.95	min

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
	g	108	scopephot	10.35	min
	h	109	termphoto	0	min
~					
r4020	a	110	termproca	0	min
	b	111	plasmaetc	31.33	min
	c	112	scopeproc	6.33	min
	d	113	termproca	0	min
~					
r4040	a	124	termproca	0	min
	b	125	alloy	68.5	min
	c	126	scopeproc	3.2	min
	d	127	termproca	0	min
r6120	a	178	termproca	0	min
	b	179	plasmaetc	43.83	min
	c	180	branson	30.2	min
	d	181	liftoffever	20.93	min
	e	182	rdB	12.7	min
	f	183	scopeproc	3.4	min
	g	184	termproca	0	min

Appendix G: Route for product 4

Data Set I

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r1000	a	1	termmtrl	0.848	min	
	b	2	scribe	0.977	min	piece
	c	3	rdA	8.09	min	
	d	4	termmtrl	2.157	min	
~						
r1001	a	6	termmtrl	2	min	
	b	7	termmtrl	2	min	
~						
r1010	a	8	termphoto	1	min	
	b	9	prime	35.41	min	
	c	10	coat	1.45	min	piece
	d	11	step	0.431	min	piece
	e	12	developA	1.828	min	piece
	f	13	rdB	8.576	min	
	g	14	fusion	3.543	min	
	h	15	scopephoto	0.329	min	piece
	i	16	termphoto	0.788	min	
~						
r1020	a	17	termproca	0.736	min	
	b	18	branson	6.895	min	
	c	19	wetbenchC	2.567	min	
	d	20	rdB	4.923	min	
	e	21	scopeproc	0.916	min	
	f	22	branson	35.763	min	
	g	23	wetbenchC	3.91	min	
	h	24	rdB	5.17	min	
	i	25	alphastepA	4.54	min	
	j	26	scopeproc	0.92	min	
	k	27	termproca	1.59	min	
~						
r1420	a	28	termimp	2	min	
	b	29	imptwo	35	min	
	c	30	impfour	40	min	
	d	31	termimp	2	min	
~						
r2400	a	32	termmtrl	0.63	min	
	b	33	wetbenchmtrls	4.58	min	
	c	34	wetbenchmtrls	6.26	min	
	d	35	rdA	7.67	min	
	e	36	sput	52.8	min	
	f	37	scopemtrls	1.7	min	
	g	38	mgage	0.41	min	piece
	h	39	termmtrl	1.7	min	
~						
r2410	a	40	termphoto	0.64	min	
	b	41	wetbenchB	1.58	min	
	c	42	rdB	4.91	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	43	prime	37	min	
	e	44	coat	2.09	min	piece
	f	45	step	2.13	min	piece
	g	46	developA	2.01	min	piece
	h	47	rdB	9.09	min	
	i	48	scopephoto	3.2	min	
	j	49	termphoto	0.57	min	
~						
r2420	a	50	termproca	1.31	min	
	b	51	plasmaetchA	15.1	min	
	c	52	wetbenchC	3.66	min	
	d	53	rdB	5.14	min	
	e	54	evapA	68.5	min	
	f	55	alphastepA	1.17	min	
	g	56	sheetres	1.39	min	
	h	57	liftoffever	20.1	min	
	i	58	scopeproc	8.19	min	
	j	59	branson	15.2	min	
	k	60	termproca	1.49	min	
~						
r2430	a	61	termproca	0.96	min	
	b	62	osi	8.89	min	
	c	63	termproca	1.26	min	
~						
r2440	a	64	termproca	0.68	min	
	b	65	rdB	16.8	min	
	c	66	omegaetch	11.9	min	piece
	d	67	rdB	8.05	min	
	e	68	scopeproc	2.27	min	
	f	69	termproca	0.57	min	
~						
r3020	a	70	termimp	0.802	min	
	b	71	imptwo	24	min	
	c	72	impfour	40.456	min	
	d	73	termimp	1.5	min	
~						
r3340	a	74	termmtrl	1.25	min	
	b	75	wetbenchmtrls	11	min	
	c	76	rdA	8.28	min	
	d	77	scopemtrls	1.98	min	
	e	78	osi	16	min	
	f	79	bakematerials	5.17	min	
	g	80	depos	14.6	min	
	h	81	thicknessB	1.32	min	
	i	82	anneal	60	min	
	j	83	termmtrl	2.07	min	
~						
r3410	a	84	termphoto	0.95	min	
	b	85	prime	34.3	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	c	86	coat	1.82	min	piece
	d	87	step	1.11	min	piece
	e	88	developA	1.89	min	piece
	f	89	rdB	5.28	min	
	g	90	fusion	3.53	min	piece
	h	91	scopephoto	1.07	min	
	i	92	termphoto	0.76	min	
~						
r3420	a	93	termimp	0.802	min	
	b	94	imptwo	24	min	
	c	95	impfour	40.456	min	
	d	96	termimp	1.5	min	
~						
r3430	a	97	termproca	2	min	
	b	98	branson	35.594	min	
	c	99	rdB	14.139	min	
	d	100	scopeproc	3.93	min	
	e	101	termproca	1.585	min	
~						
r4010	a	102	termphoto	0.54	min	
	b	103	prime	37.4	min	
	c	104	coat	1.71	min	piece
	d	105	step	1.94	min	piece
	e	106	developA	1.89	min	piece
	f	107	rdB	5	min	
	g	108	scopephoto	10.5	min	
	h	109	termphoto	2.06	min	
~						
r4020	a	110	termproca	0.72	min	
	b	111	plasmaetchA	18	min	
	c	112	scopeproc	1.12	min	
	d	113	termproca	1.01	min	
~						
r4030	a	114	termproca	0.92	min	
	b	115	wetbenchC	3.88	min	
	c	116	rdB	4.98	min	
	d	117	evapB	76.3	min	
	e	118	alphastepA	1.4	min	
	f	119	sheetres	1.4	min	
	g	120	liftoffsemi	34.5	min	
	h	121	scopeproc	2.48	min	
	i	122	branson	32.3	min	
	j	123	termproca	2.06	min	
~						
r4040	a	124	termproca	0.26	min	
	b	125	alloy	5.691	min	piece
	c	126	scopeproc	24.7	min	
	d	127	termproca	1.57	min	
~						

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r7210	a	128	termphoto	0.64	min	
	b	129	wetbenchB	1.58	min	
	c	130	rdB	4.91	min	
	d	131	prime	37	min	
	e	132	coat	2.09	min	piece
	f	133	step	2.13	min	piece
	g	134	developA	2.01	min	piece
	h	135	wetbenchC	1.25	min	
	i	136	rdB	9.09	min	
	j	137	scopephoto	3.2	min	
	k	138	termphoto	0.57	min	
~						
r7220	a	139	termproca	1.27	min	
	b	140	branson	6	min	
	c	141	evapB	55.2	min	
	d	142	alphastepA	1.03	min	
	e	143	sheetres	1.06	min	
	f	144	liftoffever	20.1	min	
	g	145	rdB	6	min	
	h	146	scopeproc	5	min	
	i	147	branson	15.8	min	
	j	148	termproca	3.84	min	
~						
r7500	a	149	termproca	2	min	
	b	150	depos	14	min	
	c	151	thicknessB	3	min	
	d	152	termproca	2	min	
~					min	
r7510	a	153	termphoto	0.64	min	
	b	154	wetbenchB	1.58	min	
	c	155	rdB	4.91	min	
	d	156	prime	37	min	
	e	157	coat	2.09	min	piece
	f	158	step	2.13	min	piece
	g	159	developA	2.01	min	piece
	h	160	developB	2	min	piece
	i	161	rdB	9.09	min	
	j	162	coat	2.09	min	piece
	k	163	scopephoto	3.2	min	
	l	164	termphoto	0.57	min	
~						
r7520	a	165	termproca	2	min	
	b	166	branson	6	min	
	c	167	wetbenchC	6	min	
	d	168	rdB	7	min	
	e	169	scopeproc	2	min	
	f	170	branson	35	min	
	g	171	rdB	7	min	
	h	172	scopeproc	2	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	i	173	bakeproc	15	min	
	j	174	termproca	2	min	
~						
r8350	a	175	termproca	2	min	
	b	176	hptest	10	min	piece
	c	177	termproca	2	min	
~						
r8400	a	178	termproca	2	min	
	b	179	tablehptest	12	min	
	c	180	termproca	2	min	
~					min	
r9000	a	181	termmtrl	0.64	min	
	b	182	scopeproc	1.32	min	piece
	c	183	rdB	5	min	
	d	184	adhsvcoat	1.97	min	piece
	e	185	bonder	167	min	
	f	186	termmtrl	0.76	min	
~						
r9020	a	187	termmtrl	0.17	min	
	b	188	grinder	73	min	
	c	189	digind	2.26	min	piece
	d	190	wetbenchgrinder	18.1	min	
	e	191	digind	0.57	min	piece
	f	192	scopeproc	2	min	
	g	193	termmtrl	2.01	min	
~						
r9112	a	194	termphoto	2	min	
	b	195	developB	2.5	min	piece
	c	196	polycoat	1.5	min	piece
	d	197	suss	25	sec	piece
	e	198	developB	2.5	min	piece
	f	199	scopephoto	3	min	piece
	g	200	scopephoto	1	min	piece
	h	201	termphoto	2	min	
~						
r9122	a	202	termproca	1.27	min	
	b	203	branson	10	min	
	c	204	wetbenchC	6	min	
	d	205	wetbenchC	2.5	min	
	e	206	rdC	7	min	
	f	207	evapB	55.2	min	
	g	208	alphastepA	1.03	min	
	h	209	sheetres	1.06	min	
	i	210	liftoffsemi	20.1	min	
	j	211	scopeproc	6	min	
	k	212	tapetest	5	min	
	l	213	termproca	3.84	min	
~						
r9160	a	214	termmtrl	0.76	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	215	demount	150	min	
	c	216	digind	0.74	min	piece
	d	217	technics	33.5	min	
	e	218	technics	26.2	min	
	f	219	termmtrl	3.2	min	
~						
r9400	a	220	termtower		min	
	b	221	scopetower		min	
	c	222	termtower		min	
~						
r9999	a	223	termtower	13	min	

Data Set II for product 4

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
r2400	a	32	termmtrl	0	min
	b	33	wetbenchmtrls	15.96	min
	d	35	rdA	7.67	min
	e	36	sput	60.6	min
	f	37	scopemtrls	4.5	min
	g	38	mgage	4.73	min
	h	39	termmtrl	0	min
~					
r2410	a	40	termphoto	0	min
	b	41	wetbenchB	11.7	min
	c	42	rdB	4.91	min
	d	43	prime	43.2	min
	e	44	coat	16.9	min
	f	45	step	16.4	min
	g	46	developA	16.6	min
	h	47	rdB	9.8	min
	i	48	scopephoto	1	min
	j	49	termphoto	0	min
~					
r2420	a	50	termproca	1.31	min
	b	51	plasmaetchA	15.1	min
	c	52	wetbenchC	3.66	min
	d	53	rdB	5.14	min
	e	54	evapA	68.5	min
	f	55	alphastepA	1.17	min
	g	56	sheetres	1.39	min
	h	57	liftoffever	20.1	min
	i	58	scopeproc	8.19	min
	j	59	branson	15.2	min
	k	60	termproca	1.49	min
~					
r4010	a	102	termphoto	0	min
	b	103	prime	42.5	min
	c	104	coat	17.15	min
	d	105	step	49.7	min
	e	106	developA	3.95	min
	f	107	rdB	15.75	min
	g	108	scopephoto	10.35	min
	h	109	termphoto	0	min
~					
r4020	a	110	termproca	0	min
	b	111	plasmaetchA	31.33	min
	c	112	scopeproc	6.33	min
	d	113	termproca	0	min
~					
r4040	a	124	termproca	0	min
	b	125	alloy	68.5	min
	c	126	scopeproc	3.2	min

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
r4040	a	124	termproca	0	min
	b	125	alloy	68.5	min
	c	126	scopeproc	3.2	min
	d	127	termproca	0	min

Appendix H: Route for product 5

Data Set I

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r1000	a	1	termmtrl	0.848	min	
	b	2	scribe	0.977	min	piece
	c	3	rdA	8.09	min	
	d	4	termmtrl	2.157	min	
~						
r1001	a	6	termmtrl	2	min	
	b	7	termmtrl	2	min	
~						
r1010	a	8	termphoto	1	min	
	b	9	prime	35.41	min	
	c	10	coat	1.45	min	piece
	d	11	step	0.431	min	piece
	e	12	developA	1.828	min	piece
	f	13	rdB	8.576	min	
	g	14	fusion	3.543	min	
	h	15	scopephoto	0.329	min	piece
	i	16	termphoto	0.788	min	
~						
r1020	a	17	termproca	0.736	min	
	b	18	branson	6.895	min	
	c	19	wetbenchC	2.567	min	
	d	20	rdB	4.923	min	
	e	21	scopeproc	0.916	min	
	f	22	branson	35.763	min	
	g	23	wetbenchC	3.91	min	
	h	24	rdB	5.17	min	
	i	25	alphastepA	4.54	min	
	j	26	scopeproc	0.92	min	
	k	27	termproca	1.59	min	
~						
r1511	a	28	termphoto	1	min	
	b	29	prime	35.41	min	
	c	30	coat	1.45	min	piece
	d	31	step	0.431	min	piece
	e	32	developA	1.828	min	piece
	f	33	rdB	8.576	min	
	g	34	fusion	3.543	min	
	h	35	scopephoto	0.329	min	piece
	i	36	termphoto	0.788	min	
~						
r1521	a	37	termimp	2	min	
	b	38	imptwo	35	min	
	c	39	impfour	40	min	
	d	40	termimp	2	min	
~						
r1531	a	41	termproca	2	min	
	b	42	branson	35	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	c	43	rdB	6	min	
	d	44	scopeproc	3	min	
	e	45	termproca	2	min	
~						
r1710	a	46	termphoto	1	min	
	b	47	prime	35.41	min	
	c	48	coat	1.45	min	piece
	d	49	step	0.431	min	piece
	e	50	developA	1.828	min	piece
	f	51	rdB	8.576	min	
	g	52	fusion	3.543	min	
	h	53	scopephoto	0.329	min	piece
	i	54	termphoto	0.788	min	
~						
r1720	a	55	termimp	0.802	min	
	b	56	imptwo	24	min	
	c	57	impfour	40.456	min	
	d	58	termimp	1.5	min	
~						
r1730	a	59	termproca	2	min	
	b	60	branson	35	min	
	c	61	rdB	6	min	
	d	62	scopeproc	3	min	
	e	63	termproca	2	min	
~						
r2400	a	64	termmtrl	0.63	min	
	b	65	wetbenchmtrls	4.58	min	
	c	66	wetbenchmtrls	6.26	min	
	d	67	rdA	7.67	min	
	e	68	sput	52.8	min	
	f	69	scopemtrls	1.7	min	
	g	70	mgage	0.41	min	piece
	h	71	termmtrl	1.7	min	
~						
r2410	a	72	termphoto	0.64	min	
	b	73	wetbenchB	1.58	min	
	c	74	rdB	4.91	min	
	d	75	prime	37	min	
	e	76	coat	2.09	min	piece
	f	77	step	2.13	min	piece
	g	78	developA	2.01	min	piece
	h	79	rdB	9.09	min	
	i	80	scopephoto	3.2	min	
	j	81	termphoto	0.57	min	
~						
r2420	a	82	termproca	1.31	min	
	b	83	plasmaetchA	15.1	min	
	c	84	wetbenchC	3.66	min	
	d	85	rdB	5.14	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	e	86	evapA	68.5	min	
	f	87	alphastepA	1.17	min	
	g	88	sheetres	1.39	min	
	h	89	liftoffever	20.1	min	
	i	90	scopeproc	8.19	min	
	j	91	branson	15.2	min	
	k	92	termproca	1.49	min	
~						
r2430	a	93	termproca	0.96	min	
	b	94	osi	8.89	min	
	c	95	termproca	1.26	min	
~						
r2440	a	96	termproca	0.68	min	
	b	97	rdB	16.8	min	
	c	98	omegaetch	11.9	min	piece
	d	99	rdB	8.05	min	
	e	100	scopeproc	2.27	min	
	f	101	termproca	0.57	min	
~						
r3010	a	97	termphoto	1	min	
	b	98	prime	35.41	min	
	c	99	coat	1.45	min	piece
	d	100	step	0.431	min	piece
	e	101	developA	1.828	min	piece
	f	102	rdB	8.576	min	
	g	103	fusion	3.543	min	
	h	104	scopephoto	0.329	min	piece
	i	105a	termphoto	0.788	min	
~						
r3020	a	105	termimp	0.802	min	
	b	106	imptwo	24	min	
	c	107	impfour	40.456	min	
	d	108	termimp	1.5	min	
~						
r3030	a	109	termproca	2	min	
	b	110	branson	35	min	
	c	111	rdB	6	min	
	d	112	scopeproc	3	min	
	e	113	termproca	2	min	
~						
r3200	a	114	termmtrl	1.25	min	
	b	115	wetbenchmtrls	11	min	
	c	116	rdA	8.28	min	
	d	117	scopeproc	1.98	min	
	e	118	osi	16	min	
	f	119	bakematerials	5.17	min	
	g	120	depos	14.6	min	
	h	121	thicknessB	1.32	min	
	i	122	scopemtrls	0.76	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	j	123	termmtrl	2.07	min	
~						
r3210	a	124	termphoto	0.95	min	
	b	125	prime	34.3	min	
	c	126	coat	1.82	min	piece
	d	127	step	1.11	min	piece
	e	128	developA	1.82	min	piece
	f	129	rdB	5.28	min	
	g	130	fusion	3.53	min	piece
	h	131	termphoto	0.76	min	
~						
r3220	a	132	termimp	0.802	min	
	b	133	imptwo	24	min	
	c	134	impfour	40.456	min	
	d	135	termimp	1.5	min	
~						
r3230	a	136	termproca	0.89	min	
	b	137	branson	35.6	min	
	c	138	rdB	4.36	min	piece
	d	139	termproca	1.59	min	
~					min	
r3240	a	140	termmtrl	2	min	
	b	141	anneal	5	min	piece
	c	142	termmtrl	2	min	
~						
r3410	a	143	termphoto	0.95	min	
	b	144	prime	34.3	min	
	c	145	coat	1.82	min	piece
	d	146	step	1.11	min	piece
	e	147	developA	1.89	min	piece
	f	148	rdB	5.28	min	
	g	149	fusion	3.53	min	piece
	h	150	scopephoto	1.07	min	
	i	151	termphoto	0.76	min	
~						
r3420	a	152	termimp	0.802	min	
	b	153	imptwo	24	min	
	c	154	impfour	40.456	min	
	d	155	termimp	1.5	min	
~						
r3430	a	156	termproca	2	min	
	b	157	branson	35.594	min	
	c	158	rdB	14.139	min	
	d	159	scopeproc	3.93	min	
	e	160	termproca	1.585	min	
~						
r3500	a	161	termmtrl	1.25	min	
	b	162	wetbenchmtrls	11	min	
	c	163	rdA	8.28	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	164	bakematerials	5.17	min	
	e	165	depos	14.6	min	
	f	166	thicknessB	1.32	min	
	g	167	scopemtrls	1.98	min	
	h	168	termmtrl	2.07	min	
~						
r3510	a	169	termphoto	3	min	
	b	170	coat	1.373	min	piece
	c	171	coat	1.718	min	piece
	d	172	developA	2.918	min	piece
	e	173	suss	2.918	min	piece
	f	174	fusion	6.493	min	
	g	175	scopeproc	6.126	min	
	h	176	termphoto	0.172	min	
~						
r3520	a	177	termproca	0.96	min	
	b	178	plasmaetchA	76.7	min	
	c	179	thicknessB	1.04	min	piece
	d	180	scopeproc	1.12	min	
	e	181	alphastepA	1.56	min	piece
	f	182	branson	10.6	min	
	g	183	termproca	2.05	min	
~						
r4010	a	184	termphoto	0.54	min	
	b	185	prime	37.4	min	
	c	186	coat	1.71	min	piece
	d	187	step	1.94	min	piece
	e	188	developA	1.89	min	piece
	f	189	rdB	5	min	
	g	190	scopephoto	10.5	min	
	h	191	termphoto	2.06	min	
~						
r4020	a	192	termproca	0.72	min	
	b	193	plasmaetchA	18	min	
	c	194	scopeproc	1.12	min	
	d	195	termproca	1.01	min	
~						
r4030	a	196	termproca	0.92	min	
	b	197	wetbenchC	3.88	min	
	c	198	rdB	4.98	min	
	d	199	evapB	76.3	min	
	e	200	alphastepA	1.4	min	
	f	201	sheetres	1.4	min	
	g	202	liftoffsemi	34.5	min	
	h	203	scopeproc	2.48	min	
	i	204	branson	32.3	min	
	j	205	termproca	2.06	min	
~						
r4040	a	206	termproca	0.26	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	207	alloy	5.691	min	piece
	c	208	scopeproc	24.7	min	
	d	209	termproca	1.57	min	
~						
r4350	a	210	termproca	2	min	
	b	211	hptest	100	min	
	c	212	termproca	2	min	
~						
r4400	a	213	termproca	2	min	
	b	214	tabletest	30	min	
	c	215	termproca	2	min	
~						
r5012	a	216	termphoto	2	min	
	b	217	coat	1.6	min	piece
	c	218	coat	1.71	min	piece
	d	219	step	3	min	piece
	e	220	developA	1.89	min	piece
	f	221	rdB	9.09	min	
	g	222	scopephoto	0.4	min	
	h	223	termphoto	0.57	min	
~						
r5022	a	224	termproca	1.27	min	
	b	225	plasmaetchB	40	min	
	c	226	evapB	55.2	min	
	d	227	alphastepA	1.03	min	
	e	228	sheetres	1.06	min	
	f	229	liftoffever	20.1	min	
	g	230	rdB	6	min	
	h	231	scopeproc	7.62	min	
	i	232	plasmaetchB	35	min	
	j	233	branson	15.8	min	
	k	234	termproca	3.84	min	
~						
r6100	a	235	termmtrl	0.34	min	
	b	236	depos	33.5	min	
	c	237	lehighton	1.73	min	
	d	238	scopemtrls	2.19	min	
	e	239	termmtrl	1.11	min	
~						
r6110	a	240	termphoto	0.95	min	
	b	241	prime	34.3	min	
	c	242	coat	1.82	min	piece
	d	243	step	1.11	min	piece
	e	244	developA	1.89	min	piece
	f	245	rdB	5.28	min	
	g	246	fusion	3.53	min	piece
	h	247	scopephoto	1.07	min	
	i	248	termphoto	0.76	min	
~						

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r6120	a	249	termproca	0.91	min	
	b	250	plasmaetchB	78.7	min	
	c	251	branson	35.5	min	
	d	252	liftoffever	20.1	min	
	e	253	rdB	14.6	min	
	f	254	scopeproc	3.84	min	piece
	g	255	termproca	1.22	min	
~						
r7210	a	256	termphoto	0.64	min	
	b	257	wetbenchB	1.58	min	
	c	258	rdB	4.91	min	
	d	259	prime	37	min	
	e	260	coat	2.09	min	piece
	f	261	step	2.13	min	piece
	g	262	developA	2.01	min	piece
	h	263	wetbenchC	1.25	min	
	i	264	rdB	9.09	min	
	j	265	scopephoto	3.2	min	
	k	266	termphoto	0.57	min	
~						
r7220	a	267	termproca	1.27	min	
	b	268	branson	6	min	
	c	269	evapB	55.2	min	
	d	270	alphastepA	1.03	min	
	e	271	sheetres	1.06	min	
	f	272	liftoffever	20.1	min	
	g	273	rdB	6	min	
	h	274	scopeproc	5	min	
	i	275	branson	15.8	min	
	j	276	termproca	3.84	min	
~						
r7900	a	277	termphoto	0.77	min	
	b	278	adhesionprom	62.8	min	
	c	279	polycoat	2.69	min	piece
	d	280	scopephoto	5	min	piece
	e	281	polybake	363	min	
	f	282	termphoto	0.89	min	
~						
r7910	a	283	termphoto	0.85	min	
	b	284	coat	2.23	min	piece
	c	285	step	5.29	min	piece
	d	286	developA	3.83	min	piece
	e	287	rdB	5.22	min	
	f	288	scopephoto	12.9	min	
	g	289	termphoto	0.17	min	
~						
r7920	a	290	termproca	1.67	min	
	b	291	plasmaetchB	87.8	min	
	c	292	scopeproc	14.5	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	293	liftoffever	20.1	min	
	e	294	rdB	1.33	min	
	f	295	scopeproc	3.81	min	
	g	296	termproca	1.81	min	
~					min	
r8020	a	297	termproca	0.75	min	
	b	298	branson	4.74	min	
	c	299	sput	50.6	min	
	d	300	evapB	35.6	min	
	e	301	alphastepA	4.15	min	
	f	302	sheetres	3.16	min	
	g	303	termproca	1.12	min	
~						
r8110	a	304	termphoto	0.71	min	
	b	305	prime	38.2	min	
	c	306	coat	2.28	min	piece
	d	307	step	3.91	min	piece
	e	308	step	0	min	piece
	f	309	developA	2.19	min	piece
	g	310	rdB	5.34	min	
	h	311	coat	2.25	min	piece
	i	312	scopephoto	5.98	min	
	j	313	termphoto	1.29	min	
~						
r8120	a	314	termproca	0.66	min	
	b	315	branson	11.4	min	
	c	316	alphastepA	1.26	min	
	d	317	goldplate	82	min	
	e	318	rdB	9.71	min	
	f	319	alphastepA	0.64	min	
	g	320	termproca	0.98	min	
~						
r8130	a	321	termproca	0.6	min	
	b	322	liftoffsemi	4.41	min	
	c	323	technics	44	min	
	d	324	wetbenchB	4.42	min	
	e	325	wetbenchB	2.25	min	
	f	326	wetbenchB	4.36	min	
	g	327	rdB	12.6	min	
	h	328	scopeproc	10.3	min	
	i	329	alphastepA	1.04	min	
	j	330	termproca	0.23	min	
~						
r8200	a	331	termphoto	0.78	min	
	b	332	adhesionprom	62.8	min	
	c	333	polycoat	2.91	min	piece
	d	334	polycoat	0	min	
	e	335	scopephoto	2	min	
	f	336	polybake	363	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	g	337	termphoto	1.164	min	
~						
r8210	a	338	termphoto	0.71	min	
	b	339	coat	2.71	min	piece
	c	340	coat	2.68	min	piece
	d	341	suss	2.46	min	piece
	e	342	developA	30.6	min	piece
	f	343	rdB	5.58	min	
	g	344	scopephoto	0.942	min	piece
	h	345	termphoto	1.23	min	
~						
r8220	a	346	termproca	1.56	min	
	b	347	plasmaetchB	173	min	
	c	348	scopeproc	1.84	min	
	d	349	liftoffever	27.8	min	
	e	350	rdB	5.1	min	
	f	351	branson	5.72	min	
	g	352	scopeproc	1.96	min	piece
	h	353	termproca	0.87	min	
~						
r8350	a	354	termproca	2	min	
	b	355	hptest	10	min	piece
	c	356	termproca	2	min	
~						
r8400	a	357	termproca	2	min	
	b	358	tablehptest	12	min	
	c	359	termproca	2	min	
~					min	
r9000	a	360	termmtrl	0.64	min	
	b	361	scopeproc	1.32	min	piece
	c	362	rdB	5	min	
	d	363	adhsvcoat	1.97	min	piece
	e	364	bonder	167	min	
	f	365	termmtrl	0.76	min	
~						
r9020	a	366	termmtrl	0.17	min	
	b	367	grinder	73	min	
	c	368	digind	2.26	min	piece
	d	369	wetbenchgrinder	18.1	min	
	e	370	digind	0.57	min	piece
	f	371	scopeproc	2	min	
	g	372	termmtrl	2.01	min	
~						
r9110	a	373	termphoto	2	min	
	b	374	developB	2.5	min	piece
	c	375	polycoat	1.5	min	piece
	d	376	rdC	7	min	
	e	377	polycoat	1.5	min	piece
	f	378	suss	25	sec	piece

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	g	379	developB	2.5	min	piece
	h	380	wetbenchC	10	min	
	i	381	rdC	7	min	
	j	382	scopephoto	3	min	piece
	k	383	scopephoto	1	min	piece
	l	384	termphoto	2	min	
~						
r9130	a	385	termphoto	2	min	
	b	386	polycoat	1.5	min	
	c	387	suss	25	sec	
	d	388	plasmaetchA	18	min	
	e	389	scopeproc	2	min	
	f	390	wetbenchC	10	min	
	g	391	rdC	7	min	
	h	392	scopeproc	2	min	
	i	393	termphoto	2	min	
~						
r9140	a	394	termproca	2	min	
	b	395	branson	7.5	min	
	c	396	wetbenchC	10	min	
	d	397	rdC	7	min	
	e	398	sputB	10	min	
	f	399	goldplate	82	min	
	g	400	rdC	7	min	
	h	401	evapB	35.6	min	
	i	402	sheetres	10	min	
	j	403	sputB	10	min	
	k	404	tapetest	10	min	
	m	405	termproca	2	min	
~						
r9142	a	406	termphoto	2	min	
	b	407	polycoat	10	min	
	c	408	suss	25	sec	
	d	409	developB	2.5	min	
	e	410	scopephoto	10	min	
	f	411	termphoto	2	min	
~					min	
r9144	a	412	termproca	2	min	
	b	413	plasmaetchA	18	min	
	c	414	scopeproc	2	min	
	d	415	wetbenchC	10	min	
	e	416	rdC	7	min	
	f	417	liftoffsemi	20	min	
	g	418	termproca	2	min	
~					min	
r9160	a	419	termmtrl	0.76	min	
	b	420	demount	150	min	
	c	421	digind	0.74	min	piece
	d	422	technics	33.5	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	e	423	technics	26.2	min	
	f	424	termmtrl	3.2	min	
~						
r9250	a	425	termtower	0.76	min	
	b	426	rftest	4.05	min	piece
	c	427	termtower	1.27	min	
~						
r9260	a	428	termtower	0.28	min	
	b	429	termtower	14.83	min	
~						
r9300	a	430	termtower	0.8	min	
	b	431	scopetower	6.21	min	piece
	c	432	termtower	0.72	min	
~						
r9999	a	433	termtower	13	min	

Data Set II for product 5

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
r2400	a	32	termmtrl	0	min
	b	33	wetbenchmtrls	15.96	min
	d	35	rdA	7.67	min
	e	36	sput	60.6	min
	f	37	scopemtrls	4.5	min
	g	38	mgage	4.73	min
	h	39	termmtrl	0	min
~					
r2410	a	40	termphoto	0	min
	b	41	wetbenchB	11.7	min
	c	42	rdB	4.91	min
	d	43	prime	43.2	min
	e	44	coat	16.9	min
	f	45	step	16.4	min
	g	46	developA	16.6	min
	h	47	rdB	9.8	min
	i	48	scopephoto	1	min
	j	49	termphoto	0	min
~					
r2420	a	50	termproca	1.31	min
	b	51	plasmaetchA	15.1	min
	c	52	wetbenchC	3.66	min
	d	53	rdB	5.14	min
	e	54	evapA	68.5	min
	f	55	alphastepA	1.17	min
	g	56	sheetres	1.39	min
	h	57	liftoffever	20.1	min
	i	58	scopeproc	8.19	min
	j	59	branson	15.2	min
	k	60	termproca	1.49	min
~					
r3520	a	110	termproca	0	min
	b	111	plasmaetchA	69.63	min
	c	112	thicknessB	11.2	min
	d	113	scopeproc	9.86	min
	e	114	alphastepA	15.3	min
	f	115	branson	8.4	min
	g	116	termproca	0	min
~					
r4010	a	102	termphoto	0	min
	b	103	prime	42.5	min
	c	104	coat	17.15	min
	d	105	step	49.7	min
	e	106	developA	3.95	min
	f	107	rdB	15.75	min
	g	108	scopephoto	10.35	min

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
	h	109	termphoto	0	min
~					
r4020	a	110	termproca	0	min
	b	111	plasmaetc	31.33	min
	c	112	scopeproc	6.33	min
	d	113	termproca	0	min
~					
r4040	a	124	termproca	0	min
	b	125	alloy	68.5	min
	c	126	scopeproc	3.2	min
	d	127	termproca	0	min
r6120	a	178	termproca	0	min
	b	179	plasmaetc	43.83	min
	c	180	branson	30.2	min
	d	181	liftoffever	20.93	min
	e	182	rdB	12.7	min
	f	183	scopeproc	3.4	min
	g	184	termproca	0	min

Appendix I: Route for product 6

Data Set I

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r1000	a	1	termmtrl	0.848	min	
	b	2	scribe	0.977	min	piece
	c	3	rdA	8.09	min	
	d	4	termmtrl	2.157	min	
~						
r1001	a	6	termmtrl	2	min	
	b	7	termmtrl	2	min	
~						
r1010	a	8	termphoto	1	min	
	b	9	prime	35.41	min	
	c	10	coat	1.45	min	piece
	d	11	step	0.431	min	piece
	e	12	developA	1.828	min	piece
	f	13	rdB	8.576	min	
	g	14	fusion	3.543	min	
	h	15	scopephoto	0.329	min	piece
	i	16	termphoto	0.788	min	
~						
r1020	a	17	termproca	0.736	min	
	b	18	branson	6.895	min	
	c	19	wetbenchC	2.567	min	
	d	20	rdB	4.923	min	
	e	21	scopeproc	0.916	min	
	f	22	branson	35.763	min	
	g	23	wetbenchC	3.91	min	
	h	24	rdB	5.17	min	
	i	25	alphastepA	4.54	min	
	j	26	scopeproc	0.92	min	
	k	27	termproca	1.59	min	
~						
r1820	a	28	termimp	0.802	min	
	b	29	imptwo	24	min	
	c	30	impfour	40.456	min	
	d	31	termimp	1.5	min	
~						
r2400	a	32	termmtrl	0.63	min	
	b	33	wetbenchmtrls	4.58	min	
	c	34	wetbenchmtrls	6.26	min	
	d	35	rdA	7.67	min	
	e	36	sput	52.8	min	
	f	37	scopemtrls	1.7	min	
	g	38	mgage	0.41	min	piece
	h	39	termmtrl	1.7	min	
~						
r2410	a	40	termphoto	0.64	min	
	b	41	wetbenchB	1.58	min	
	c	42	rdB	4.91	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	d	43	prime	37	min	
	e	44	coat	2.09	min	piece
	f	45	step	2.13	min	piece
	g	46	developA	2.01	min	piece
	h	47	rdB	9.09	min	
	i	48	scopephoto	3.2	min	
	j	49	termphoto	0.57	min	
~						
r2420	a	50	termproca	1.31	min	
	b	51	plasmaetchA	15.1	min	
	c	52	wetbenchC	3.66	min	
	d	53	rdB	5.14	min	
	e	54	evapA	68.5	min	
	f	55	alphastepA	1.17	min	
	g	56	sheetres	1.39	min	
	h	57	liftoffever	20.1	min	
	i	58	scopeproc	8.19	min	
	j	59	branson	15.2	min	
	k	60	termproca	1.49	min	
~						
r2430	a	61	termproca	0.96	min	
	b	62	osi	8.89	min	
	c	63	termproca	1.26	min	
~						
r2440	a	64	termproca	0.68	min	
	b	65	rdB	16.8	min	
	c	66	omegaetch	11.9	min	piece
	d	67	rdB	8.05	min	
	e	68	scopeproc	2.27	min	
	f	69	termproca	0.57	min	
~						
r3020	a	70	termimp	0.802	min	
	b	71	imptwo	24	min	
	c	72	impfour	40.456	min	
	d	73	termimp	1.5	min	
~						
r3200	a	74	termmtrl	1.25	min	
	b	75	wetbenchmtrls	11	min	
	c	76	rdA	8.28	min	
	d	77	scopeproc	1.98	min	
	e	78	osi	16	min	
	f	79	bakematerials	5.17	min	
	g	80	depos	14.6	min	
	h	81	thicknessB	1.32	min	
	i	82	scopemtrls	0.76	min	
	j	83	termmtrl	2.07	min	
~						
r3210	a	84	termphoto	0.95	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	85	prime	34.3	min	
	c	86	coat	1.82	min	piece
	d	87	step	1.11	min	piece
	e	88	developA	1.82	min	piece
	f	89	rdB	5.28	min	
	g	90	fusion	3.53	min	piece
	h	91	termphoto	0.76	min	
~						
r3220	a	92	termimp	0.802	min	
	b	93	imptwo	24	min	
	c	94	impfour	40.456	min	
	d	95	termimp	1.5	min	
~						
r3230	a	96	termproca	0.89	min	
	b	97	branson	35.6	min	
	c	98	rdB	4.36	min	piece
	d	99	termproca	1.59	min	
~					min	
r3240	a	100	termmtrl	2	min	
	b	101	anneal	5	min	piece
	c	102	termmtrl	2	min	
~						
r3410	a	103	termphoto	0.95	min	
	b	104	prime	34.3	min	
	c	105	coat	1.82	min	piece
	d	106	step	1.11	min	piece
	e	107	developA	1.89	min	piece
	f	108	rdB	5.28	min	
	g	109	fusion	3.53	min	piece
	h	110	scopephoto	1.07	min	
	i	111	termphoto	0.76	min	
~						
r3420	a	112	termimp	0.802	min	
	b	113	imptwo	24	min	
	c	114	impfour	40.456	min	
	d	115	termimp	1.5	min	
~						
r3430	a	116	termproca	2	min	
	b	117	branson	35.594	min	
	c	118	rdB	14.139	min	
	d	119	scopeproc	3.93	min	
	e	120	termproca	1.585	min	
~						
r3510	a	121	termphoto	3	min	
	b	122	coat	1.373	min	piece
	c	123	coat	1.718	min	piece
	d	124	developA	2.918	min	piece
	e	125	suss	2.918	min	piece
	f	126	fusion	6.493	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	g	127	scopeproc	6.126	min	
	h	128	termphoto	0.172	min	
~						
r3520	a	129	termproca	0.96	min	
	b	130	plasmaetchA	76.7	min	
	c	131	thicknessB	1.04	min	piece
	d	132	scopeproc	1.12	min	
	e	133	alphastepA	1.56	min	piece
	f	134	branson	10.6	min	
	g	135	termproca	2.05	min	
~						
r4010	a	136	termphoto	0.54	min	
	b	137	prime	37.4	min	
	c	138	coat	1.71	min	piece
	d	139	step	1.94	min	piece
	e	140	developA	1.89	min	piece
	f	141	rdB	5	min	
	g	142	scopephoto	10.5	min	
	h	143	termphoto	2.06	min	
~						
r4020	a	144	termproca	0.72	min	
	b	145	plasmaetchA	18	min	
	c	146	scopeproc	1.12	min	
	d	147	termproca	1.01	min	
~						
r4030	a	148	termproca	0.92	min	
	b	149	wetbenchC	3.88	min	
	c	150	rdB	4.98	min	
	d	151	evapB	76.3	min	
	e	152	alphastepA	1.4	min	
	f	153	sheetres	1.4	min	
	g	154	liftoffsemi	34.5	min	
	h	155	scopeproc	2.48	min	
	i	156	branson	32.3	min	
	j	157	termproca	2.06	min	
~						
r5010	a	158	termphoto	2	min	
	b	159	wetbenchB	1.58	min	
	c	160	rdB	4.91	min	
	d	161	prime	37	min	
	e	162	coat	2.09	min	piece
	f	163	step	2.13	min	piece
	g	164	developA	2.01	min	piece
	h	165	wetbenchC	4	min	
	i	166	rdB	9.09	min	
	j	167	scopephoto	0.4	min	piece
	k	168	termphoto	0.57	min	
~						
r5020	a	169	termproca	1.27	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	b	170	branson	6.53	min	
	c	171	evapB	55.2	min	
	d	172	alphastepA	1.03	min	
	e	173	sheetres	1.06	min	
	f	174	liftoffever	20.1	min	
	g	175	rdB	6	min	
	h	176	scopeproc	7.62	min	
	i	177	branson	15.8	min	
	j	178	termproca	3.84	min	
~						
r6100	a	179	termmtrl	0.34	min	
	b	180	depos	33.5	min	
	c	181	lehighton	1.73	min	
	d	182	scopemtrls	2.19	min	
	e	183	termmtrl	1.11	min	
~						
r6110	a	184	termphoto	0.95	min	
	b	185	prime	34.3	min	
	c	186	coat	1.82	min	piece
	d	187	step	1.11	min	piece
	e	188	developA	1.89	min	piece
	f	189	rdB	5.28	min	
	g	190	fusion	3.53	min	piece
	h	191	scopephoto	1.07	min	
	i	192	termphoto	0.76	min	
~						
r6120	a	193	termproca	0.91	min	
	b	194	plasmaetchB	78.7	min	
	c	195	branson	35.5	min	
	d	196	liftoffever	20.1	min	
	e	197	rdB	14.6	min	
	f	198	scopeproc	3.84	min	piece
	g	199	termproca	1.22	min	
~						
r7900	a	200	termphoto	0.77	min	
	b	201	adhesionprom	62.8	min	
	c	202	polycoat	2.69	min	piece
	d	203	scopephoto	5	min	piece
	e	204	polybake	363	min	
	f	205	termphoto	0.89	min	
~						
r7910	a	206	termphoto	0.85	min	
	b	207	coat	2.23	min	piece
	c	208	step	5.29	min	piece
	d	209	developA	3.83	min	piece
	e	210	rdB	5.22	min	

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
r7920	a	213	termproca	1.67	min	
	b	214	plasmaetchB	87.8	min	
	c	215	scopeproc	14.5	min	
	d	216	liftoffever	20.1	min	
	e	217	rdB	1.33	min	
	f	218	scopeproc	3.81	min	
	g	219	termproca	1.81	min	
~					min	
r8020	a	220	termproca	0.75	min	
	b	221	branson	4.74	min	
	c	222	sput	50.6	min	
	d	223	evapB	35.6	min	
	e	224	alphastepA	4.15	min	
	f	225	sheetres	3.16	min	
	g	226	termproca	1.12	min	
~						
r8110	a	227	termphoto	0.71	min	
	b	228	prime	38.2	min	
	c	229	coat	2.28	min	piece
	d	230	step	3.91	min	piece
	e	231	step	0	min	piece
	f	232	developA	2.19	min	piece
	g	233	rdB	5.34	min	
	h	234	coat	2.25	min	piece
	i	235	scopephoto	5.98	min	
	j	236	termphoto	1.29	min	
~						
r8120	a	237	termproca	0.66	min	
	b	238	branson	11.4	min	
	c	239	alphastepA	1.26	min	
	d	240	goldplate	82	min	
	e	241	rdB	9.71	min	
	f	242	alphastepA	0.64	min	
	g	243	termproca	0.98	min	
~						
r8130	a	244	termproca	0.6	min	
	b	245	liftoffsemi	4.41	min	
	c	246	technics	44	min	
	d	247	wetbenchB	4.42	min	
	e	248	wetbenchB	2.25	min	
	f	249	wetbenchB	4.36	min	
	g	250	rdB	12.6	min	
	h	251	scopeproc	10.3	min	
	i	252	alphastepA	1.04	min	
	j	253	termproca	0.23	min	
~						

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	e	258	scopephoto	2	min	
	f	259	polybake	363	min	
	g	260	termphoto	1.164	min	
~						
r8210	a	261	termphoto	0.71	min	
	b	262	coat	2.71	min	piece
	c	263	coat	2.68	min	piece
	d	264	suss	2.46	min	piece
	e	265	developA	30.6	min	piece
	f	266	rdB	5.58	min	
	g	267	scopephoto	0.942	min	piece
	h	268	termphoto	1.23	min	
~						
r8220	a	269	termproca	1.56	min	
	b	270	plasmaetchB	173	min	
	c	271	scopeproc	1.84	min	
	d	272	liftoffever	27.8	min	
	e	273	rdB	5.1	min	
	f	274	branson	5.72	min	
	g	275	scopeproc	1.96	min	piece
	h	276	termproca	0.87	min	
~						
r8350	a	277	termproca	2	min	
	b	278	hptest	10	min	piece
	c	279	termproca	2	min	
~						
r8400	a	280	termproca	2	min	
	b	281	tablehptest	12	min	
	c	282	termproca	2	min	
~					min	
r9000	a	283	termmtrl	0.64	min	
	b	284	scopeproc	1.32	min	piece
	c	285	rdB	5	min	
	d	286	adhsvcoat	1.97	min	piece
	e	287	bonder	167	min	
	f	288	termmtrl	0.76	min	
~						
r9020	a	289	termmtrl	0.17	min	
	b	290	grinder	73	min	
	c	291	digind	2.26	min	piece
	d	292	wetbenchgrinder	18.1	min	
	e	293	digind	0.57	min	piece
	f	294	scopeproc	2	min	
	g	295	termmtrl	2.01	min	
~						

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS	PTPER
	e	300	technics	26.2	min	
	f	301	termmtrl	3.2	min	
~						
r9250	a	302	termtower	0.76	min	
	b	303	rftest	4.05	min	piece
	c	304	termtower	1.27	min	
~						
r9260	a	305	termtower	0.28	min	
	b	306	termtower	14.83	min	
~						
r9300	a	307	termtower	0.8	min	
	b	308	scopetower	6.21	min	piece
	c	309	termtower	0.72	min	

Data Set II for product 6

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
r2400	a	32	termmtrl	0	min
	b	33	wetbenchmtrls	15.96	min
	d	35	rdA	7.67	min
	e	36	sput	60.6	min
	f	37	scopemtrls	4.5	min
	g	38	mgage	4.73	min
	h	39	termmtrl	0	min
~					
r2410	a	40	termphoto	0	min
	b	41	wetbenchB	11.7	min
	c	42	rdB	4.91	min
	d	43	prime	43.2	min
	e	44	coat	16.9	min
	f	45	step	16.4	min
	g	46	developA	16.6	min
	h	47	rdB	9.8	min
	i	48	scopephoto	1	min
	j	49	termphoto	0	min
~					
r2420	a	50	termproca	1.31	min
	b	51	plasmaetchA	15.1	min
	c	52	wetbenchC	3.66	min
	d	53	rdB	5.14	min
	e	54	evapA	68.5	min
	f	55	alphastepA	1.17	min
	g	56	sheetres	1.39	min
	h	57	liftoffever	20.1	min
	i	58	scopeproc	8.19	min
	j	59	branson	15.2	min
	k	60	termproca	1.49	min
~					
r3520	a	110	termproca	0	min
	b	111	plasmaetchA	69.63	min
	c	112	thicknessB	11.2	min
	d	113	scopeproc	9.86	min
	e	114	alphastepA	15.3	min
	f	115	branson	8.4	min
	g	116	termproca	0	min
~					
r4010	a	102	termphoto	0	min
	b	103	prime	42.5	min
	c	104	coat	17.15	min
	d	105	step	49.7	min
	e	106	developA	3.95	min
	f	107	rdB	15.75	min
	g	108	scopephoto	10.35	min

IGNORE	IGNORE	STEP	STNFAM	PTIME	PTUNITS
	h	109	termphoto	0	min
~					
r4020	a	110	termproca	0	min
	b	111	plasmaetchA	31.33	min
	c	112	scopeproc	6.33	min
	d	113	termproca	0	min
~					
r6120	a	178	termproca	0	min
	b	179	plasmaetchB	43.83	min
	c	180	branson	30.2	min
	d	181	liftoffever	20.93	min
	e	182	rdB	12.7	min
	f	183	scopeproc	3.4	min
	g	184	termproca	0	min

Appendix J: The Down Calendar Worksheet

DOWNCALNAME	DOWNCALTYPE	MTTF	MTTFUNITS	MTTR	MTTRUNITS
impfour_down	mttf_by_cal	113.17	hr	8.5	hr
imptwo_down	mttf_by_cal	259.2	hr	13	hr
step_1_down	mttf_by_cal	280	hr	22	hr
step_2_down	mttf_by_cal	386	hr	12	hr
suss_down	mttf_by_cal	313.9	hr	7	hr
fusion_1_down	mttf_by_cal	507.96	hr	7.4	hr
fusion_2_down	mttf_by_cal	1299.2	hr	7.4	hr
polycoat_down	mttf_by_cal	178.26	hr	7.37	hr
adhsvcoat_down	mttf_by_cal	374.94	hr	7.48	hr
developA_1_down	mttf_by_cal	598.92	hr	5.5	hr
developA_2_down	mttf_by_cal	560.73	hr	6.15	hr
developB_down	mttf_by_cal	865.23	hr	6.55	hr
coat_1_down	mttf_by_cal	200.66	hr	6.68	hr
coat_2_down	mttf_by_cal	641.76	hr	7.61	hr
coat_3_down	mttf_by_cal	186.95	hr	5.75	hr
liftoffsemi_down	mttf_by_cal	356.43	hr	3.4	hr
branson_1_down	mttf_by_cal	613.72	hr	2.43	hr
branson_2_down	mttf_by_cal	383.48	hr	3.94	hr
branson_3_down	mttf_by_cal	383.48	hr	3.94	hr
technics_down	mttf_by_cal	1018.845	hr	1.47	hr
goldplate_down	mttf_by_cal	738.56	hr	10.37	hr
bonder_1_down	mttf_by_cal	582.22	hr	2	hr
bonder_2_down	mttf_by_cal	685.2	hr	5.12	hr
rdA_down	mttf_by_cal	801	hr	4.69	hr
wetbenchB_down	mttf_by_cal	612	hr	4.19	hr
wetbenchC_down	mttf_by_cal	550	hr	0.03	hr
rdB_1_down	mttf_by_cal	994.24	hr	5	hr
rdB_2_down	mttf_by_cal	880	hr	4	hr
rdB_3_down	mttf_by_cal	663	hr	5	hr
rdC_1_down	mttf_by_cal	625	hr	10	hr
rdC_2_down	mttf_by_cal	771	hr	14	hr
sput_1_down	mttf_by_cal	231.83	hr	16.43	hr
plasmaetchA_down	mttf_by_cal	293.04	hr	7.34	hr
plasmaetchB_down	mttf_by_cal	356.97	hr	2.02	hr
evapB_1_down	mttf_by_cal	86.86	hr	3.62	hr
evapA_down	mttf_by_cal	156.13	hr	3.07	hr
evapB_2_down	mttf_by_cal	127.96	hr	8.37	hr
depos_down	mttf_by_cal	126.13	hr	5.16	hr
sput_2_down	mttf_by_cal	201.18	hr	7.72	hr
sputB_down	mttf_by_cal	346.42	hr	4.64	hr
omegaetch_1_down	mttf_by_cal	308.009	hr	9.27	hr
omegaetch_2_down	mttf_by_cal	293.49	hr	11.12	hr
liftoffever_down	mttf_by_cal	531.42	hr	0.545	hr

DOWNCALNAME	DOWNCALTYPE	MTTF	MTTFUNITS	MTTR	MTTRUNITS
thicknessB_2_down	mttf_by_cal	827.24	hr	11.95	hr
sheetres_1_down	mttf_by_cal	912	hr	2.54	hr
sheetres_2_down	mttf_by_cal	395	hr	2.54	hr
wafertransfer_down	mttf_by_cal	774	hr	3.62	hr
mgage_down	mttf_by_cal	106.12	hr	8	hr
lehighton_down	mttf_by_cal	329.24	hr	4.43	hr
hpink_down	mttf_by_cal	1809.47	hr	6	hr
hptest_down	mttf_by_cal	1407.486	hr	3.1	hr
anneal_down	mttf_by_cal	249.88	hr	5.1	hr
alloy_down	mttf_by_cal	582.32	hr	3.595	hr
prime_1_down	mttf_by_cal	826.8	hr	4.34	hr
prime_2_down	mttf_by_cal	826.8	hr	4.34	hr
scribe_down	mttf_by_cal	408.85	hr	5.88	hr
wetbenchmtrls_down	mttf_by_cal	600	hr	0.05	hr
dicingsaw_1_down	mttf_by_cal	485	hr	0.01	hr
dicingsaw_2_down	mttf_by_cal	595	hr	0.01	hr
scribebreak_1_down	mttf_by_cal	885	hr	9	hr
scribebreak_2_down	mttf_by_cal	667	hr	0.34	hr
grinder_down	mttf_by_cal	392.49	hr	6.64	hr
demount_down	mttf_by_cal	449.94	hr	8.53	hr

Appendix K: The Preventive Maintenance Calendar Worksheet

PMCALNAME	PMCALTYPE	MTBPM	MTBPMUNITS	MTTR	MTTRUNITS
impfour_30	mtbpm_by_proctime	30	day	3.39	hr
impfour_90	mtbpm_by_proctime	90	day	3.39	hr
impfour_120	mtbpm_by_proctime	120	day	3.39	hr
impfour_365	mtbpm_by_proctime	365	day	3.39	hr
~	mtbpm_by_proctime				
imptwo_30	mtbpm_by_proctime	30	day	5.61	hr
imptwo_60	mtbpm_by_proctime	60	day	5.61	hr
imptwo_90	mtbpm_by_proctime	90	day	5.61	hr
imptwo_180	mtbpm_by_proctime	180	day	5.61	hr
imptwo_365	mtbpm_by_proctime	365	day	5.61	hr
~	mtbpm_by_proctime				
step_7	mtbpm_by_proctime	7	day	2.73	hr
step_30	mtbpm_by_proctime	30	day	2.73	hr
step_90	mtbpm_by_proctime	90	day	2.73	hr
~	mtbpm_by_proctime				
suss_30	mtbpm_by_proctime	30	day	2	hr
~	mtbpm_by_proctime				
fusion_30	mtbpm_by_proctime	30	day	1	hr
~	mtbpm_by_proctime				
maskcln_30	mtbpm_by_proctime	30	day	3	hr
maskcln_90	mtbpm_by_proctime	90	day	3	hr
maskcln_180	mtbpm_by_proctime	180	day	3	hr
~	mtbpm_by_proctime				
polycoat_7	mtbpm_by_proctime	7	day	1.25	hr
polycoat_30	mtbpm_by_proctime	30	day	1.25	hr
polycoat_60	mtbpm_by_proctime	60	day	1.25	hr
~	mtbpm_by_proctime				
adhsvcoat_30	mtbpm_by_proctime	30	day	0.02	hr
~	mtbpm_by_proctime				
developA_7	mtbpm_by_proctime	7	day	4	hr
developA_30	mtbpm_by_proctime	30	day	4	hr
~	mtbpm_by_proctime				
developB_7	mtbpm_by_proctime	7	day	2.5	hr
developB_30	mtbpm_by_proctime	30	day	2.5	hr
~	mtbpm_by_proctime				
coat_7	mtbpm_by_proctime	7	day	4.824	hr
coat_14	mtbpm_by_proctime	14	day	4.824	hr
coat_30	mtbpm_by_proctime	30	day	4.824	hr
coat_180	mtbpm_by_proctime	180	day	4.824	hr
coat_365	mtbpm_by_proctime	365	day	4.824	hr
~	mtbpm_by_proctime				
liftoffsemi_7	mtbpm_by_proctime	7	day	0.04	hr
liftoffsemi_90	mtbpm_by_proctime	90	day	0.04	hr
liftoffsemi_180	mtbpm_by_proctime	180	day	0.04	hr

PMCALNAME	PMCALTYPE	MTBPM	MTBPMUNITS	MTTR	MTTRUNITS
branson_365	mtbpm_by_proctime	365	day	0.4	hr
~	mtbpm_by_proctime				
goldplate_90	mtbpm_by_proctime	180	day	10	hr
goldplate_180	mtbpm_by_proctime	180	day	10	hr
~	mtbpm_by_proctime				
bonder_90	mtbpm_by_proctime	90	day	3	hr
~	mtbpm_by_proctime				
sput_90	mtbpm_by_proctime	90	day	12.59	hr
sput_120	mtbpm_by_proctime	120	day	12.59	hr
sput_180	mtbpm_by_proctime	180	day	12.59	hr
sput_365	mtbpm_by_proctime	365	day	12.59	hr
~	mtbpm_by_proctime				
plasmaetchA_30	mtbpm_by_proctime	30	day	5.35	hr
plasmaetchA_90	mtbpm_by_proctime	90	day	5.35	hr
plasmaetchA_365	mtbpm_by_proctime	365	day	5.35	hr
~	mtbpm_by_proctime				
plasmaetchB_30	mtbpm_by_proctime	30	day	6.07	hr
plasmaetchB_90	mtbpm_by_proctime	90	day	6.07	hr
plasmaetchB_365	mtbpm_by_proctime	365	day	6.07	hr
~	mtbpm_by_proctime				
evapB_90	mtbpm_by_proctime	90	day	5.62	hr
evapB_365	mtbpm_by_proctime	365	day	5.62	hr
~	mtbpm_by_proctime				
evapA_90	mtbpm_by_proctime	90	day	5.67	hr
evapA_365	mtbpm_by_proctime	365	day	5.67	hr
~	mtbpm_by_proctime				
depos_7	mtbpm_by_proctime	7	day	2	hr
depos_30	mtbpm_by_proctime	30	day	2	hr
depos_180	mtbpm_by_proctime	180	day	2	hr
depos_270	mtbpm_by_proctime	270	day	2	hr
depos_365	mtbpm_by_proctime	365	day	2	hr
~	mtbpm_by_proctime				
sputB_90	mtbpm_by_proctime	90	day	9.97	hr
sputB_365	mtbpm_by_proctime	365	day	9.97	hr
~	mtbpm_by_proctime				
omegaetch_7	mtbpm_by_proctime	7	day	4	hr
omegaetch_30	mtbpm_by_proctime	30	day	4	hr
omegaetch_90	mtbpm_by_proctime	90	day	4	hr
omegaetch_180	mtbpm_by_proctime	180	day	4	hr
omegaetch_365	mtbpm_by_proctime	365	day	4	hr
~	mtbpm_by_proctime				
liftoffever_90	mtbpm_by_proctime	90	day	6	hr
~	mtbpm_by_proctime				
thicknessA_90	mtbpm_by_proctime	90	day	5	hr
~	mtbpm_by_proctime				
thicknessB_30	mtbpm_by_proctime	30	day	6	hr

PMCALNAME	PMCALTYPE	MTBPM	MTBPMUNITS	MTTR	MTTRUNITS
alloy_180	mtbpm_by_proctime	180	day	3.2	hr
~	mtbpm_by_proctime				
prime_90	mtbpm_by_proctime	90	day	3.5	hr
prime_365	mtbpm_by_proctime	365	day	3.5	hr
~	mtbpm_by_proctime				
polybake_90	mtbpm_by_proctime	90	day	3	hr
~	mtbpm_by_proctime				
scribe_90	mtbpm_by_proctime	90	day	1.8	hr
scribe_180	mtbpm_by_proctime	180	day	1.8	hr
~	mtbpm_by_proctime				
adhesionprom_90	mtbpm_by_proctime	90	day	3	hr
adhesionprom_180	mtbpm_by_proctime	180	day	3	hr
~	mtbpm_by_proctime				
bakematerials_90	mtbpm_by_proctime	90	day	2	hr
~	mtbpm_by_proctime				
grinder_120	mtbpm_by_proctime	120	day	2	hr
~	mtbpm_by_proctime				
demount_30	mtbpm_by_proctime	30	day	5.18	hr
~	mtbpm_by_proctime				
technics_7	mtbpm_by_proctime	7	day	0.515	hr
~	mtbpm_by_proctime				
dicingsaw_90	mtbpm_by_proctime	90	day	2	hr
dicingsaw_180	mtbpm_by_proctime	180	day	1	hr
~	mtbpm_by_proctime				
hpink_90	mtbpm_by_proctime	90	day	2	hr
~	mtbpm_by_proctime				
hptest_90	mtbpm_by_proctime	90	day	10	hr

Appendix L: Order files

Order file for experiment 3 with Data Set I

ORDER	LOT	PART	START	REPEAT	RUNITS	RPT#
Order 1	L1	p1	11/29/99 8:00	12	hr	100
Order 2	L2	p2	11/29/99 8:00	12	hr	100
Order 3	L3	p3	11/29/99 8:00	12	hr	100
Order 4	L4	p4	11/29/99 8:00	12	hr	100
Order 5	L5	p5	11/29/99 8:00	12	hr	100
Order 6	L6	p6	11/29/99 8:00	12	hr	100

Order file for experiment 4 with Data Set I

ORDER	LOT	PART	START	REPEAT	RUNITS	RPT#
Order 1	L1	p1	11/29/99 8:00	12	hr	150
Order 2	L2	p2	11/29/99 8:00	12	hr	150
Order 3	L3	p3	11/29/99 8:00	12	hr	150
Order 4	L4	p4	11/29/99 8:00	12	hr	150
Order 5	L5	p5	11/29/99 8:00	12	hr	150
Order 6	L6	p6	11/29/99 8:00	12	hr	150

Appendix M: Developed Heuristic in C++

```
#include "asapdefs.h"
#include "newonlybottleneckdown.h"

BEGIN_EXPORT_SECTION
        // Functions prototyped in this export section make the corresponding
        // function visible to programs outside the extension DLL.  This is
        // necessary for INIT FUNCTIONS, ACTIONS, RULES, etc.
        DEFINE_RULE_FUNCTION(rule_newonlybottleneckdown)
END_EXPORT_SECTION

BEGIN_RULE_FUNCTION(rule_newonlybottleneckdown)
{       //Passed: FIRes* theRes, FIFilterList& theSourceList, FIBoolean skipExecution

 /* Define variables to get the resource family and the factory pointer */
   FIFamily* fam = theRes->family();
        FIResFam* rfam = (FIResFam*)fam;
        FISmInt state;
        FIFactory* factory = theRes->factory();
        FIInt ent =0;

 /* Find the number of jobs on the resource's worklist */
        ent = rfam->entitiesOnWorkList();
 /* Define variables to store the entity which is chosen. Also define pointers to the rank
and rule functions that will be used later in the program. */
   FIFilterList  dolist,dolist1, dolist2, dolist3, dolist4, dolist5, dolist6, dolist7, dolist8,
dolist9, dolist10, dolist11;
        FIInt skiptofilter = 1;
        static FIRankFuncPtr aLSRankPtr = ((FIRankDef*)theRes->factory()-
>findEntity("rank_LS", RANKDEF))->funcPtr();
```

```
        static FIRuleFuncPtr aFirstRulePtr = ((FIRuleDef*)theRes->factory()-
>findEntity("rule_FIRST",RULEDEF))->funcPtr();
        static FIRankFuncPtr aLPRRankPtr = ((FIRankDef*)theRes->factory()-
>findEntity("rank_LPR", RANKDEF))->funcPtr();
   static FIRankFuncPtr aLTRRankPtr = ((FIRankDef*)theRes->factory()-
>findEntity("rank_LTR", RANKDEF))->funcPtr();
        FIBoolean skipExecution = FALSE;
 /* Define the standard lotsize. */
        FIInt lotsize = 8;

 /* If no jobs on the resource family's worklist, return FALSE. */
        if (ent ==0) {
                return FALSE;
        }

 /* If the resource family is a batching family and if the number of entities on the worklist
is less than the batch maximum, form a batch of all the products on the worklist and
schedule them next. */
        if (rfam->batchFamily() && ent <= theRes->batchData()->batchMax()) {
                theRes->doTask(theSourceList,dolist,skiptofilter);
           if (dolist.entries()) {
                   theRes->selectTask(dolist);
                   return TRUE;
                }
                else { return FALSE;}
        }
 /* If number of jobs on the family worklist is 1, then schedule it. */
        if (ent == 1) {
                theRes->doTask(theSourceList,dolist,skiptofilter);
           if (dolist.entries()) {
                   theRes->selectTask(dolist);
                   return TRUE;
```

```
                }
                else {return FALSE;}
        }
        FIInt Presentfamstations = rfam->entities().entries();


/* Next step is to identify the bottleneck. Define a attribute to the station families, to store
the time required by all the jobs in the system, at that time, at a particular station family.
*/

   static FIAttIndex tempai;
  static FIAttIndex ptimeai;
{FINameGroupIterator  aStnFamIterator(((FIFactory*)theRes->factory())-
>allStnFams());
          FIResFam*     aResFam;
          while ((aResFam=(FIResFam*)aStnFamIterator())!=0) {
                  aResFam->addUserAttribute("Ptime","0.0", TYPEREAL);
          }
         }

   FIResFam* astnfam;
        FIReal time ;
        static FIAttIndex aPtimeai;
        static FIAttIndex fPtimeai;
        static FIAttIndex rPtimeai;

 /* Iterate through all the orders and all the lots within the order, for the remainder of the
lots' route. Add the time required by each lot at a station family in the station family's
attribute, define above. */

{FINameGroupIterator          aOrderIterator(((FIFactory*)theRes->factory())-
>allOrders());
```

```
FIOrder*    aOrder;
while ((aOrder = (FIOrder*)aOrderIterator())!=0) {
        FINameGroupIterator      aLotIterator(aOrder->entities());
        FILot*       aLot;
            while ((aLot=(FILot*)aLotIterator())!=0) {
                    if ( aLot->started()  && !aLot->finished()) {
                       FIStep* anxtstep;
                            anxtstep = aLot->step();
                            while (anxtstep){
                               astnfam = anxtstep->firstRequiredStationFam();
          FIGenData& aPtime = astnfam->userAttribute("Ptime",aPtimeai);
                                    time = aPtime.real() + anxtstep-
>procDuration(lotsize,1,CALC_MEAN,1,aLot) + anxtstep-
>setupDuration(1.0,CALC_MEAN,aLot);
                                    aPtime.real(time);
                                    anxtstep = anxtstep->nextStep();
                            }
                    }
            }
     }
     }
```

/* Identify the bottleneck. This is done by dividing the total time required by all the lots in the system, at a given time, at a station family, by the number of stations in the station family. The station family having the highest ratio is the bottleneck. If the station is a batching family then the average batch size till the present time is used as an additional divisor. */

```
FIResFam* maxutiliz;
maxutiliz = astnfam;
FIInt Maxutilizstations = maxutiliz->entities().entries();
```

```
   FIRes *maxutilizres1 = (FIRes*)maxutiliz->entities().first();
        FIReal maxutilizmx;
   if (maxutiliz->batchFamily()) {
                maxutilizmx = maxutiliz->statistics()-
>batchSize().average(STAT_ABSOLUTE);
        }
        else {maxutilizmx = 1;}
        {FINameGroupIterator        aStnFamIterator(((FIFactory*)theRes->factory())-
>allStnFams());
         FIResFam*    aResFam;
         while ((aResFam=(FIResFam*)aStnFamIterator())!=0) {
                 FIInt Noofstations = aResFam->entities().entries();
                FIReal aResFammx;
                 FIRes *aResFamres = (FIRes*)aResFam->entities().first();
                 if (aResFam->batchFamily()) {
                 aResFammx = aResFam->statistics()-
>batchSize().average(STAT_ABSOLUTE);
                 }
                 else {aResFammx = 1;}

                FIGenData& rPtime = maxutiliz->userAttribute("Ptime", rPtimeai);
                FIGenData& fPtime = aResFam->userAttribute("Ptime", fPtimeai);
                if ((fPtime.real()/(Noofstations * aResFammx)) >
rPtime.real()/(Maxutilizstations * maxutilizmx))) {
                maxutiliz = aResFam;
                Maxutilizstations = Noofstations;
                maxutilizmx = aResFammx;
                }
         }
        }
```

```
 /* The bottleneck is pointed to by the pointer maxutiliz. After the bottleneck is identfied,
remove the attribute Ptime. */
        {FINameGroupIterator          aStnFamIterator(((FIFactory*)theRes->factory())-
>allStnFams());
         FIResFam*     aResFam;
         while ((aResFam=(FIResFam*)aStnFamIterator())!=0) {
                  aResFam->removeUserAttribute("Ptime");
         }
        }

 /* If the station family at which the decision is being made, is the bottleneck, arrange lots
in the order of Least Percent Processing Time Remaining, and select the first one. */
        if (maxutiliz == rfam) {
           FIFilterList rank1 ;
                 rank1.copyFilterListEntsSorted(theSourceList,aLPRRankPtr);
                 theRes->doTask(rank1,dolist5,skiptofilter);
           if (dolist5.entries()) {
                    theRes->selectTask(dolist5);
                    return TRUE;
                 }
                 else { return FALSE; }
        }

/* Iterate through all the remaining routes of the lots in the system. Identify which lots are
going to the bottleneck along the remainder of the routes and assign them the attribute
BottleFlag = Y. If the lots visit the bottleneck, then store the time required to reach the
bottleneck in the attribute Timetobottleneck. If any of the stations along the lots route to
the bottleneck or along the remainder of their route, are down, then the attribute
DownFlag = D. Store the lot with the minimum time to the bottleneck in the pointer
minbottleneck. */
        FIStep *nxtstep;
        FIResFam *stnfam;
```

```
        FIFilterList Source;
        FIFilterList bottleneck, nonbottleneck;
 /* Arrange all the lots in the worklist according to the rank LPR. */
        Source.copyFilterListEntsSorted(theSourceList,aLPRRankPtr);
        FIReal  minboproctime=0, boproctime =0, slack=0;
        FIInt   bocounter1 = 0, dcounter =0;
        FIFilterListIterator sourceiter(Source);
        FILot *Lot, *boLot, *boLot1;
        static FIAttIndex flagai, btimeai, dflagai, aflagai, adflagai;

        while ((Lot = (FILot*)sourceiter()) != 0) {
                Lot->addUserAttribute("BottleFlag","N", TYPESTRING);
                Lot->addUserAttribute("Timetobottleneck", "0.0", TYPEREAL);
                Lot->addUserAttribute("DownFlag", "U", TYPESTRING);
           FIReal proctime = 0;
                nxtstep = Lot->step();
                FIGenData& flag = Lot->userAttribute("BottleFlag", flagai);
                FIGenData& btime = Lot->userAttribute("Timetobottleneck", btimeai);
                FIGenData& dflag = Lot->userAttribute("DownFlag", dflagai);
                while (nxtstep !=0 || nxtstep != NULL) {
                   stnfam = nxtstep->firstRequiredStationFam();
                        FINameGroupIterator aR_it4(stnfam->entities());
                        FIRes* aRes4 =0;
                        FIInt sflag = 0;

                        while ((aRes4 = (FIRes*)aR_it4())!= 0) {
                                if (aRes4->down() || aRes4->userDefinedUC()-
>qtyToNextOccurence() < proctime + nxtstep-
>procDuration(lotsize,1.0,CALC_MEAN,1,Lot) + nxtstep->setupDuration(1.0,
CALC_MEAN, Lot) + stnfam->statistics()-
>workListCycle().average(STAT_ABSOLUTE)) {
                                        sflag = 1;
```

```
            }
        }

        if (sflag == 1) {
            dflag.string("D");
        }
        if (stnfam == maxutiliz) {
            flag.string("Y");
           boproctime = proctime;
            btime.real(boproctime);
            if (dflag.string() == "U") {
                bocounter1= bocounter1 + 1;
               if(bocounter1 == 1 || boproctime < minboproctime) {
                 boLot = Lot;
                 minboproctime = boproctime;
                    }

               if(bocounter1 != 1 && boproctime == minboproctime) {
                 FIReal slack(aLSRankPtr(Lot)),
                        slack1(aLSRankPtr(boLot));
                 if (slack < slack1) {
                     boLot = Lot;
                   }
                 }
            }
            nxtstep = 0;
        }
        else {
           proctime = proctime + nxtstep-
>procDuration(lotsize,1.0,CALC_MEAN,1,Lot) + nxtstep->setupDuration(1.0,
CALC_MEAN, Lot) + stnfam->statistics()-
>workListCycle().average(STAT_ABSOLUTE);
```

```
                    nxtstep = nxtstep->nextStep();
                }
            }
        }
    FIInt bottle = 0 , nonbottle = 0;
        FIFilterList minbottleneck;
        FILot *bLot =0;
        FIFilterListIterator sourceiter1(Source);
    if(bocounter1 >=1) {
          minbottleneck.append(boLot);
        }
```

/* Iterate through all the lots on the station family's worklist. If the lot is visiting the bottleneck, without any down stations along its route to the bottleneck, append it to the list bottleneck. If a lot is not visiting the bottleneck but has no down stations along its remaining route, append it to the list nonbottleneck. */

```
        while ((bLot = (FILot*)sourceiter1()) != 0) {
            FIGenData& aflag = bLot->userAttribute("BottleFlag", aflagai);
            FIGenData& adflag = bLot->userAttribute("DownFlag", adflagai);

            if (aflag.string() == "Y" && adflag.string() == "U") {
                bottleneck.append(bLot);
                bottle =  bottle + 1;
            }
            if (aflag.string() == "N" && adflag.string() == "U") {
                nonbottleneck.append(bLot);
                nonbottle =  nonbottle + 1;
            }
        }
```

/* Check the status of the bottleneck machine (i.e. if it is idle or processing.) */

```
        FIInt bottleidleness =0;
```

```
        FIReal timeonmaxutiliz = 0;
        FINameGroupIterator aR_it3(maxutiliz->entities());
    FIRes* aRes3 = 0;
    while ((aRes3 = (FIRes*)aR_it3())!= 0) {
                if (!aRes3->down() && aRes3->claimers().entries() == 0) {
                        bottleidleness =1;
                }
        }

 /* If the bottleneck is processing some lot, find the time after which the bottleneck will
be idle. This is considering the lots on the bottleneck's worklist and also the lot which the
bottleneck is processing presently. */
    if (bottleidleness == 0) {
            FINameGroupIterator aR_it5(maxutiliz->entities());
            FIRes *aRes5 = 0;
            double secforfinish1, minsecforfinish = 100000000;
            FIInt iteration = 0, bottlestations = 0;
            AutoTime& Presenttime1 = modelTime();
            AutoTime Finishtime1, testtime;

            while ((aRes5 = (FIRes*)aR_it5())!= 0) {
                    if (!aRes5->down() && aRes5->claimers().entries() > 0) {
            FISchedEntity* aSE1 = (FISchedEntity*)aRes5->claimers().first();

                            iteration = iteration + 1;
                            if (aSE1->actionDuration() >= 0.0) {
                Finishtime1 =  aSE1->factory()->modelStartTime() + (aSE1-
>actionStartSeconds() + aSE1->actionDuration());
                                }
            secforfinish1 = Finishtime1.secondsFrom(Presenttime1);
                        if (secforfinish1 < minsecforfinish || iteration ==1) {
                                minsecforfinish = secforfinish1;
```

```
                    }
                }
            }
        timeonmaxutiliz = (maxutiliz->estPTimeOnWorkList()/(maxutiliz-
>entities().entries())) + minsecforfinish;
    }

 /* If there are lots going to the bottleneck on the present station family's worklist and if
the bottleneck is idle, select the lot with the smallest time to the bottleneck. */

    if (bottle >= 1) {
        if (bottleidleness == 1) {
            FIFilterListIterator sourceiter2(Source);
            FILot *cLot;
            while ((cLot = (FILot*)sourceiter2()) != 0) {
                cLot->removeUserAttribute("BottleFlag");
                cLot->removeUserAttribute("Timetobottleneck");
                cLot->removeUserAttribute("DownFlag");
            }
            if(bottle>=2) {
                bottleneck.remove(boLot);
                boLot1 = (FILot*)bottleneck.first();
                Source.remove(boLot1);
                Source.prepend(boLot1);
            }
            Source.remove(boLot);
            Source.prepend(boLot);
            theRes->doTask(Source,dolist1,skiptofilter);
        if (dolist1.entries()) {
            theRes->selectTask(dolist1);
        return TRUE;
            }
```

```
                else { return FALSE ; }
        }

 /* If there are lots going to the bottleneck and the bottleneck is busy, iterate through the lots not going to the bottleneck and find if any can be processed without causing idleness at the bottleneck machine. */
        if (nonbottle >= 1) {
           FIFilterListIterator sourceiter3(nonbottleneck);
                double secforfinish;
                FILot *dLot;

                while ((dLot = (FILot*)sourceiter3()) != 0) {
                        FIReal stepproc = dLot->step()->procDuration(lotsize,1.0,CALC_MEAN,1,dLot) + dLot->step()->setupDuration(1.0, CALC_MEAN, dLot);;
        FINameGroupIterator aR_it4(rfam->entities());
        FIRes* aRes4 = 0;
                        AutoTime& Presenttime = modelTime();
                        AutoTime Finishtime;
        while ((aRes4 = (FIRes*)aR_it4())!= 0 && aRes4 != theRes) {
                                if (!aRes4->down() && aRes4->claimers().entries() > 0) {
                FISchedEntity* aSE = (FISchedEntity*)aRes4->claimers().first();
                if (aSE->actionDuration() >= 0.0) {
                   Finishtime=  aSE->factory()->modelStartTime() + (aSE->actionStartSeconds() + aSE->actionDuration());
                                        }
                secforfinish = Finishtime.secondsFrom(Presenttime);
                 msg() << "The time2 is " << secforfinish << endl;
                                if (secforfinish < stepproc) {
                                    stepproc = secforfinish;
                                }
```

```
                        }
                    }
                    slack = timeonmaxutiliz - minboproctime - stepproc;

                    if (slack >=0) {
                        FIFilterListIterator sourceiter4(Source);
                        FILot *eLot;
                        while ((eLot = (FILot*)sourceiter4()) != 0) {
                          eLot->removeUserAttribute("BottleFlag");
                          eLot-
>removeUserAttribute("Timetobottleneck");
                                eLot->removeUserAttribute("DownFlag");
                        }
                        Source.remove(dLot);
                      Source.prepend(dLot);
                        theRes->doTask(Source,dolist2,skiptofilter);
                  if (dolist2.entries()) {
                      theRes->selectTask(dolist2);
                   return TRUE;
                        }
                        else { return FALSE; }
                    }
                  }
              }

 /* If there are lots going to the bottleneck, then iterate among them in the order of LPR,
to find if any can be scheduled without causing idleness at the bottleneck. */
              FIFilterListIterator sourceiter5(bottleneck);
              FILot *fLot;
              static FIAttIndex ttimeai;

              while ((fLot = (FILot*)sourceiter5()) != 0) {
```

```
                    FIGenData& ttime = fLot->userAttribute("Timetobottleneck",
ttimeai);
                    if(ttime.isValid()) {

                    if (timeonmaxutiliz  >= ttime.real() ) {
                            FIFilterListIterator sourceiter6(Source);
                            FILot *gLot;

                            while ((gLot = (FILot*)sourceiter6()) != 0) {
                                    gLot->removeUserAttribute("BottleFlag");
                                    gLot->removeUserAttribute("Timetobottleneck");
                                    gLot->removeUserAttribute("DownFlag");
                            }
                            Source.remove(fLot);
                            Source.prepend(fLot);
                            theRes->doTask(Source,dolist3,skiptofilter);
                if (dolist3.entries()) {
                        theRes->selectTask(dolist3);
                   return TRUE;
                            }
                            else { return FALSE; }
                    }
                    }
                    else { msg() << " The lot " << fLot->name() << " does not have a
real attribute " << ttime.real() << endl; }
                }
 /* If all the above fails, schedule the lot with the minimum time to the bottleneck
machine. */
                FIFilterListIterator sourceiter7(Source);
                FILot *hLot;
                while ((hLot = (FILot*)sourceiter7()) != 0) {
                        hLot->removeUserAttribute("BottleFlag");
```

```
                hLot->removeUserAttribute("Timetobottleneck");
                hLot->removeUserAttribute("DownFlag");
            }
            Source.remove(boLot);
            Source.prepend(boLot);

            theRes->doTask(Source,dolist4,skiptofilter);
        if (dolist4.entries()) {
              theRes->selectTask(dolist4);
          return TRUE;
            }
            else { return FALSE; }
    }
 /* If there are no lots going to the bottleneck and if there are some lots in the
nonbottleneck list, schedule the first one
   according to LPR. */

    if(nonbottle >=1) {
            FIFilterListIterator sourceiter8(Source);
            FILot *iLot;
            while ((iLot = (FILot*)sourceiter8()) != 0) {
                iLot->removeUserAttribute("BottleFlag");
                iLot->removeUserAttribute("Timetobottleneck");
                iLot->removeUserAttribute("DownFlag");
            }
      theRes->doTask(Source,dolist6,skiptofilter);
      if (dolist6.entries()) {
            theRes->selectTask(dolist6);
        return TRUE;
      }
      else { return FALSE;}
    }
```

```
 /* If all the above fails, then if a lot visits the bottleneck along the remainder of its route
append it to the dbottleneck list else append it to the dnonbottleneck list. i.e. disregard if
the attribute DownFlag. */
        FIInt dbottle = 0 , dnonbottle = 0, dcounter1 = 0;
        FIFilterList  dbottleneck, dnonbottleneck;
        FIReal dtimemin =0;
        FILot *kLot, *downLot, *downLot1;
        FIFilterListIterator sourceiter10(Source);
        static FIAttIndex dtimeai, botimeai, botflagai;

        while ((kLot = (FILot*)sourceiter10()) != 0) {
                FIGenData& botflag = kLot->userAttribute("BottleFlag", botflagai);
                if (botflag.string() == "Y") {
                        dbottleneck.append(kLot);
                        dbottle =  dbottle + 1;

                   FIGenData& dtime = kLot->userAttribute("Timetobottleneck",
dtimeai);
                        if (dtime.real() < dtimemin || dcounter1 ==0) {
                                downLot = kLot;
                           dtimemin = dtime.real();
                                dcounter1 = dcounter1 + 1;
                        }
                }

                if (botflag.string() == "N") {
                        dnonbottleneck.append(kLot);
                        dnonbottle =  dnonbottle + 1;
                }
        }
```

```
 /* Repeat the same procedure as before, with the list bottleneck replaced by dbottleneck
and the list nonbottleneck replaced by dnonbottleneck. */

        if (bottleidleness ==1) {
                if (dbottle >=1) {
                   FIFilterListIterator sourceiter9(Source);
                   FILot *jLot;
                   while ((jLot = (FILot*)sourceiter9()) != 0) {
                           jLot->removeUserAttribute("BottleFlag");
                           jLot->removeUserAttribute("Timetobottleneck");
                                   jLot->removeUserAttribute("DownFlag");
                        }
                        if (dbottle >=2) {
                                   dbottleneck.remove(downLot);
                                   downLot1 = (FILot*)dbottleneck.first();
                                   Source.remove(downLot1);
                                   Source.prepend(downLot1);
                        }
                   Source.remove(downLot);
                   Source.prepend(downLot);
                   theRes->doTask(Source,dolist7,skiptofilter);
                   if (dolist7.entries()) {
                      theRes->selectTask(dolist7);
                return TRUE;
                           }
                   else { return FALSE ; }
                }
        }

        if (bottleidleness == 0) {
      if (dbottle >= 1) {
                   if (dnonbottle >= 1) {
```

```
                    FIFilterListIterator sourceiter11(dnonbottleneck);
                        double secforfinish =0;
                        FILot *lLot=0;

                        while ((lLot = (FILot*)sourceiter11()) != 0) {
                                FIReal stepproc = lLot->step()-
>procDuration(lotsize,1.0,CALC_MEAN,1,lLot) + lLot->step()->setupDuration(1.0,
CALC_MEAN, lLot);
            FINameGroupIterator aR_it6(rfam->entities());
            FIRes* aRes6 = 0;
                                AutoTime& Presenttime = modelTime();
                                AutoTime Finishtime =0;
            while ((aRes6 = (FIRes*)aR_it6())!= 0 && aRes6 != theRes) {
                                    if (!aRes6->down() && aRes6-
>claimers().entries() > 0) {
                    FISchedEntity* aSE = (FISchedEntity*)aRes6->claimers().first();
                    if (aSE->actionDuration() >= 0.0) {
                       Finishtime=  aSE->factory()->modelStartTime() + (aSE-
>actionStartSeconds() + aSE->actionDuration());
                                                }
                    secforfinish = Finishtime.secondsFrom(Presenttime);
                                                msg() << "The time3 is " <<
secforfinish<< endl;

                                        if (secforfinish < stepproc) {
                                                stepproc = secforfinish;
                                                }
                                    }
                                }

                        slack = timeonmaxutiliz - dtimemin - stepproc;
```

```
                        if (slack >=0) {
                                FIFilterListIterator sourceiter12(Source);
                                FILot *eLot;
                                while ((eLot = (FILot*)sourceiter12()) != 0) {
                                   eLot->removeUserAttribute("BottleFlag");
                                   eLot-
>removeUserAttribute("Timetobottleneck");
                                                eLot-
>removeUserAttribute("DownFlag");
                                                }
                                Source.remove(lLot);
                             Source.prepend(lLot);
                                theRes->doTask(Source,dolist8,skiptofilter);
                    if (dolist8.entries()) {
                          theRes->selectTask(dolist8);
                       return TRUE;
                                                }
                                else { return FALSE; }
                             }
                        }
                    }

                FIFilterListIterator sourceiter13(dbottleneck);
                FILot *mLot =0;
                static FIAttIndex ttimeai;

                while ((mLot = (FILot*)sourceiter13()) != 0) {
                        FIGenData& ttime = mLot->userAttribute("Timetobottleneck",
ttimeai);

                        if(ttime.isValid()) {
```

```
                    if (timeonmaxutiliz  >= ttime.real() ) {
                        FIFilterListIterator sourceiter14(Source);
                        FILot *gLot;

                        while ((gLot = (FILot*)sourceiter13()) != 0) {
                            gLot->removeUserAttribute("BottleFlag");
                            gLot-
>removeUserAttribute("Timetobottleneck");
                                gLot-
>removeUserAttribute("DownFlag");
                            }
                        Source.remove(mLot);
                        Source.prepend(mLot);

                       theRes->doTask(Source,dolist9,skiptofilter);
                if (dolist9.entries()) {
                    theRes->selectTask(dolist9);
                return TRUE;
                        }
                    else { return FALSE; }
                  }
                }

                else { msg() << " The lot " << mLot->name() << " does not have
a real attribute " << ttime.real() << endl; }
                }

            FIFilterListIterator sourceiter15(Source);
            FILot *hLot;

            while ((hLot = (FILot*)sourceiter15()) != 0) {
                hLot->removeUserAttribute("BottleFlag");
```

```
                        hLot->removeUserAttribute("Timetobottleneck");
                            hLot->removeUserAttribute("DownFlag");
                    }
                Source.remove(downLot);
                Source.prepend(downLot);

                theRes->doTask(Source,dolist10,skiptofilter);
            if (dolist10.entries()) {
                    theRes->selectTask(dolist10);
                return TRUE;
                    }
                else { return FALSE; }
    }
      }
      if(dnonbottle >=1) {
        FIFilterListIterator sourceiter16(Source);
        FILot *iLot;
        while ((iLot = (FILot*)sourceiter16()) != 0) {
                iLot->removeUserAttribute("BottleFlag");
                iLot->removeUserAttribute("Timetobottleneck");
                iLot->removeUserAttribute("DownFlag");
        }
        theRes->doTask(Source,dolist11,skiptofilter);
        if (dolist11.entries()) {
                theRes->selectTask(dolist11);
            return TRUE;
        }
        else { return FALSE;}
      }
      return FALSE;
}       //Returns: FIBoolean ("return TRUE;" or "return FALSE;")
END_RULE_FUNCTION
```

Appendix N: Branch and Bound program in C

```
/* Reading a continous LP problem and reoptimizing it to obtain an integer solution */

/* Bring in the CPLEX function declarations and the C library
   header file stdio.h with the following single include. */
#include "cplex.h"

/* Bring in the declarations for the string and character functions
   and malloc */
#include <ctype.h>
#include <stdlib.h>
#include <string.h>

/* Include declarations for functions in this program. */
#ifdef CPX_PROTOTYPE_ANSI
static void
   free_and_null (char **ptr);
#else
static void
   free_and_null ();

#endif
#ifdef CPX_PROTOTYPE_ANSI
int
main (void)
#else
int
main ()
#endif
{
/* Declare and allocate space for the variables and arrays that will contain
   the data which defines the LP problem. */
```

```
char     file[8] = "semi.lp";
CPXENVptr  env = NULL;
CPXLPptr   lp  = NULL;
int        status;
int        cur_numrows, cur_numcols;
double     *x  = NULL;
int        j;
char **cur_colname = NULL;
char *cur_colnamestore = NULL;
int  surplus;
int cur_colnamespace;
int lpstat =0;
int minindex =0, maxindex =0, mincount =1, maxcount =1;
double min =0, max=0;
int flag =0;
double objval, ubound = 0;
int itcnt =0;
int i =0, solcounter =0;
int colindex;
int maxlevel = 100000;
int levelcnt;
int level[100000];
int branchflag[100000];
int cntall = 2;
int indexall[2] = {0,0};
char luall[3] = {'L','U','\0'};
double bdfirst[2] = {1,1};
double bdsecond[2] = {0,0};
double bdreset[2] = {0,1};

   for (levelcnt =0; levelcnt < maxlevel; levelcnt ++) {
      level[levelcnt] = -1;
```

```
    branchflag[levelcnt] = 0;
  }
  /* Initialize the CPLEX environment. */
/*line 77 */
  env = CPXopenCPLEX (&status);

  /* If an error occurs, the status value indicates the reason for
     failure. A call to CPXgeterrorstring will produce the text of
     the error message. Note that CPXopenCPLEX produces no output,
     so the only way to see the cause of the error is to use
     CPXgeterrorstring. For other CPLEX routines, the errors will
     be seen if the CPX_PARAM_SCRIND indicator is set to CPX_ON. */

  if (env == NULL) {
    char errmsg[1024];
    fprintf (stderr, "Could not open CPLEX environment.\n");
    CPXgeterrorstring (env, status, errmsg);
    fprintf (stderr, "%s", errmsg);
    goto TERMINATE;
  }
  /* Turn on the output to the screen */
  status = CPXsetintparam(env, CPX_PARAM_SCRIND, CPX_ON);
  if (status !=0) {
    fprintf (stderr,
             "Failure to turn on screen indicator, error %d. \n",status);
    goto TERMINATE;
  }
  /* Fill in the data for the problem, We read from the file semi.lp. */

  lp = CPXcreateprob (env, &status, file);

  if ( lp == NULL ) {
```

```
    fprintf (stderr, "Failed to create LP.\n");
    goto TERMINATE;
  }
  /* Load the problem */
  status = CPXreadcopyprob (env, lp, file, NULL);
  if ( status ) {
    fprintf (stderr, "Failed to read and copy the problem data.\n");
    goto TERMINATE;
  }


  /* Optimize the problem and obtain the solution. */
OPTIMIZE:
  status = CPXoptimize (env, lp);
/* line 128 */
  if (status) {
    fprintf (stderr, "Failed to optimize LP. \n");
    goto TERMINATE;
  }
  lpstat = CPXgetstat(env,lp);
  if (lpstat ==2 ){
    if (itcnt ==0) {
      fprintf(stderr, "Problem infeasible or unbounded. \n ");
      goto TERMINATE;
    }
    else {
      i = i - 1;
      if (branchflag[i] != 0) {
        do {
          if (i ==0) {
            fprintf (stderr, "Search completed. \n");
            goto TERMINATE;
```

```
            }
            indexall[0] = level[i];
            indexall[1] = level[i];
            status = CPXchgbds (env,lp,cntall,indexall,luall,bdreset);

            if (status) {
               fprintf (stderr, "Failed to change bounds1\n");
               goto TERMINATE;
            }
            branchflag[i] = 0;
            i = i -1;
          }while(branchflag[i] ==1);
        }

        indexall[0] = level[i];
        indexall[1] = level[i];
        status = CPXchgbds (env,lp,cntall,indexall,luall,bdsecond);

        if (status) {
          fprintf (stderr, "Failed to change bounds2\n");
          goto TERMINATE;
        }
        branchflag[i] = 1;
        i = i + 1;
        goto OPTIMIZE;
      }
      }
    itcnt = itcnt + 1;
   /* Code to change the bounds of a variable to either 0 or 1 */
    cur_numcols = CPXgetnumcols (env,lp);
    cur_numrows = CPXgetnumrows (env,lp);
```

```
status = CPXgetobjval(env,lp,&objval);
if (status) {
   fprintf(stderr, "Failed to obtain objective value.\n");
   goto TERMINATE;
}
if (objval > ubound && solcounter == 1) {
   i = i - 1;
   if (branchflag[i] != 0) {
        do {
          if (i ==0) {
             fprintf (stderr, "Search completed. \n");
             goto TERMINATE;
          }
          indexall[0] = level[i];
          indexall[1] = level[i];
          status = CPXchgbds (env,lp,cntall,indexall,luall,bdreset);
          if (status) {
             fprintf (stderr, "Failed to change bounds3\n");
             goto TERMINATE;
          }
          branchflag[i] = 0;
          i = i -1;
        }while(branchflag[i] ==1);
   }
   indexall[0] = level[i];
   indexall[1] = level[i];
   status = CPXchgbds (env,lp,cntall,indexall,luall,bdsecond);
   if (status) {
      fprintf (stderr, "Failed to change bounds4\n");
      goto TERMINATE;
   }
   branchflag[i] = 1;
```

```
    i = i + 1;
    goto OPTIMIZE;
  }
/* Allocate space for solution */
  x = (double *) malloc (cur_numcols*sizeof(double));

  if ( x == NULL) {
    fprintf (stderr, "Out of memory for solution at x. \n");
    goto TERMINATE;
  }
  status = CPXgetx (env, lp, x, 0, cur_numcols-1);

/* Now get the column names for the problem.  First we determine how
     much space is used to hold the names, and then do the allocation.
    Then we call CPXgetcolname() to get the actual names. */
  status = CPXgetcolname (env, lp, NULL, NULL, 0, &surplus, 0,
                          cur_numcols-1);
  if (( status != CPXERR_NEGATIVE_SURPLUS ) &&
     ( status != 0 )                      )  {
    fprintf (stderr,
             "Could not determine amount of space for column names.\n");
    goto TERMINATE;
  }
  cur_colnamespace = - surplus;
  if ( cur_colnamespace > 0 ) {
    cur_colname      = (char **) malloc (sizeof(char *)*cur_numcols);
  /*   cur_colname1     = (char**) malloc  (sizeof(char *)*cur_numcols */
    cur_colnamestore = (char *)  malloc (cur_colnamespace);
  /*   cur_colnamestore1 = (char *) malloc (cur_colnamespace); */
    if ( cur_colname      == NULL ||
         cur_colnamestore == NULL   ) {
       fprintf (stderr, "Failed to get memory for column names.\n");
```

```
      status = -1;
      goto TERMINATE;
    }
    status = CPXgetcolname (env, lp, cur_colname, cur_colnamestore,
                            cur_colnamespace, &surplus, 0, cur_numcols-1);
    if ( status ) {
      fprintf (stderr, "CPXgetcolname failed.\n");
      goto TERMINATE;
    }
  }
  else {
    fprintf (stderr,"No names associated with problem.  Using Fake names.\n");
  }

  for (j=0; j< cur_numcols; j++) {
    if (*cur_colname[j] == 'X') {
      if (x[j] != 1 && x[j] !=0) {
        if (x[j] > max || maxcount ==1) {
          max = x[j];
          maxindex = j;
          maxcount = maxcount + 1;
          flag =1;
        }
      }
    }
  }
  if (flag == 1) {
     level[i] = maxindex;
     indexall[0] = maxindex;
     indexall[1] = maxindex;

     status = CPXchgbds (env,lp,cntall,indexall,luall,bdfirst);
```

```
      if (status) {
         fprintf (stderr, "Failed to change bounds5\n");
         goto TERMINATE;
      }
      maxcount =1;
      maxindex =0;
      max = 0;
      i = i + 1;
      flag = 0;
      free_and_null ((char **) &x);
      free_and_null ((char **) &cur_colname);
      free_and_null ((char **) &cur_colnamestore);
      goto OPTIMIZE;
   }
   if (solcounter ==0 || objval < ubound) {
     fprintf(stderr,"The optimal solution is %.10g\n",objval);
     for (j=0; j< cur_numcols; j++) {
       if (*cur_colname[j] == 'X') {
          fprintf (stderr, " The variable %-16s has a value of %17.10g. \n",cur_colname[j],
x[j]);
       }
     }
      ubound = objval;
   }
   i = i - 1;
   if (branchflag[i] != 0) {
     do {
       if (i ==0) {
          fprintf (stderr, "Search completed. \n");
          goto TERMINATE;
       }
       indexall[0] = level[i];
```

```
      indexall[1] = level[i];
      status = CPXchgbds (env,lp,cntall,indexall,luall,bdreset);
      if (status) {
         fprintf (stderr, "Failed to change bounds6\n");
         goto TERMINATE;
      }
      branchflag[i] = 0;
      i = i -1;
   }while(branchflag[i] ==1);
}
indexall[0] = level[i];
indexall[1] = level[i];
status = CPXchgbds (env,lp,cntall,indexall,luall,bdsecond);
if (status) {
   fprintf (stderr, "Failed to change bounds7\n");
   goto TERMINATE;
}
branchflag[i] = 1;
i = i + 1;
solcounter = 1;
free_and_null ((char **) &x);
free_and_null ((char **) &cur_colname);
free_and_null ((char **) &cur_colnamestore);
goto OPTIMIZE;

TERMINATE:
if (solcounter ==1) {
   fprintf(stderr,"The optimal objective value is %.10g. \n ",ubound);
}

/*Free the data read in by the readers */
free_and_null ((char **) &x);
```

```
   free_and_null ((char **) &cur_colname);
   free_and_null ((char **) &cur_colnamestore);

   if (lp != NULL) {
      status = CPXfreeprob (env, &lp);
      if (status) {
         fprintf (stderr, "CPXunloadprob failed, error code %d. \n", status);
      }
   }
   /* Free the CPLEX environment if necessary. */
   if (env != NULL) {
      status = CPXcloseCPLEX (&env);

      /* The CPXcloseCPLEX produces no output,
         so the only way to see the cause of the error is to use
         CPXgeterrorstring. For other CPLEX routines, the errors will
         be seen if the CPX_PARAM_SCRIND is set to CPX_ON */
      if ( status ) {
         char errmsg [1024];
         fprintf (stderr, "Could not close CPLEX environment. \n");
         CPXgeterrorstring (env, status, errmsg);
         fprintf (stderr, "%s", errmsg);
      }
   }
   return (status);
} /* END main */
/* This routine frees up the pointer *ptr, and sets *ptr to NULL */
#ifdef CPX_PROTOTYPE_ANSI
static void
free_and_null (char **ptr)
#else
static void
```

```
free_and_null (ptr)
char **ptr;
#endif
{
  if (*ptr !=NULL) {
    free (*ptr);
    *ptr = NULL;
  }
}
```

Vita

Sameer T. Shikalgar was born on 3rd August 1976 in Mumbai, India. He earned his Bachelor of Engineering degree from Fr. Conceicao Rodrigues College of Engineering in Mumbai. During his work with Larsen and Turbo, India, he gained a greater appreciation of manufacturing engineering theories and principles. He enrolled for graduate studies in the Industrial and Systems engineering program at Virginia Polytechnic and State University (VPI&SU) in the Fall of 2000. Upon graduation from VPI&SU he accepted a job with IBM, E. Fishkill as an Industrial Engineer.