Intelligent Shop Scheduling for Semiconductor Manufacturing

By

Amr Arisha, B.Eng. (Hons.)

This thesis is submitted to Dublin City University as the fulfilment of the requirement for the award of the degree of

Doctor of Philosophy

Supervisors:

Paul Young Ph.D.

Professor M. EL Baradie, Ph.D. D.Sc.

Professor M.S.J. Hashmi, Ph.D. D.Sc.

School of Mechanical and Manufacturing Engineering

Dublin City University

DCU

2003

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ______________ ID No: 99145685
Amr Arisha

Date: 09-06-2003

Acknowledgement

First, I am so grateful to my God for everything I have in my entire life.

There are many who deserve special thanks and acknowledgement for their help. I express my gratitude and appreciation to my supervisor, **Prof. Mohie El Baradie** for his guidance and encouragement.

I may not be able to find words to express my gratitude but I hope you know how much I appreciate the help and support you gave me in that long way for PhD. I feel proud that I had the luck such a kind and helpful friend to be my supervisor. Thanks **Paul.**

I would like to thank **Dr. Cathal Heavey** for taking his time to be my extern examiner; it is an honor to me. Thanks also to **Dr. Jeremiah Murphy**, my intern examiner for his constructive remarks.

Thanks to Intel-Ireland for their sponsorship of this research project. Many thanks to Intel-Ireland staff, especially **Ger Ryan**, **Mike O'Dwyer**, and **Prof. James Ignizio** for their help and sincere support they gave during this research.

I am grateful to my university AASTMT, especially **Prof. Omar Abdel Aziz**, and **Prof. Essam Rouchdy** for their support during all these years of work and research.

Special Thanks also to **Prof. M.S.J. Hashmi**, Head of the school for his fatherhood encouragement and support. I would like to express my appreciation for **Martina Reddy**, and my colleagues and friends **David**, **Ahmed Imhamed**, **Toan**, **Clint**, **Nicolas**, **Antje**, and **Martina**. Thanks guys. Thanks to our library staff especially ILL staff, **Marie** and **Margo**, for their sincere help during last years.

Special Thanks to my teacher **Prof. Hamdy Elwany**, Head of the production engineering department, Alexandria University for his continuous support and sincere advices. I owe thanks to my friends, **Khaled El-Kilany**, **Nagy**, **Safy**, **Myra**, **Regis**, **Sandra**, **Maksude** and **Amira**, for their support.

At last but never the least, My **parents**, my brother **Dr. Khaled Arisha**, my sister **Dr. Kholoud Arisha**, I hope you know how often you inspired me, and how much I look up to you for the values you added to my life with being my family. I wish for you and all the love that is in my heart all the happiness and health.

Dedicated to my parents

with all my love

Intelligent Shop Scheduling for Semiconductor Manufacturing

Amr Arisha **B.Eng. (Hons.)**

ABSTRACT

Semiconductor market sales have expanded massively to more than 200 billion dollars annually accompanied by increased pressure on the manufacturers to provide higher quality products at lower cost to remain competitive. Scheduling of semiconductor manufacturing is one of the keys to increasing productivity, however the complexity of manufacturing high capacity semiconductor devices and the cost considerations mean that it is impossible to experiment within the facility. There is an immense need for effective decision support models, characterizing and analyzing the manufacturing process, allowing the effect of changes in the production environment to be predicted in order to increase utilization and enhance system performance. Although many simulation models have been developed within semiconductor manufacturing very little research on the simulation of the photolithography process has been reported even though semiconductor manufacturers have recognized that the scheduling of photolithography is one of the most important and challenging tasks due to complex nature of the process.

Traditional scheduling techniques and existing approaches show some benefits for solving small and medium sized, straightforward scheduling problems. However, they have had limited success in solving complex scheduling problems with stochastic elements in an economic timeframe. This thesis presents a new methodology combining advanced solution approaches such as simulation, artificial intelligence, system modeling and Taguchi methods, to schedule a photolithography toolset. A new structured approach was developed to effectively support building the simulation models. A single tool and complete toolset model were developed using this approach and shown to have less than 4% deviation from actual production values. The use of an intelligent scheduling agent for the toolset model shows an average of 15% improvement in simulated throughput time and is currently in use for scheduling the photolithography toolset in a manufacturing plant.

Table of Contents

List of Figures

List of Tables

Glossary of Acronyms

Acronym	Definition
AI	Artificial Intelligence
AM	Agile Manufacturing
AMT	Advanced Manufacturing Technology
ANN	Artificial Neural Network
ANOM	Analysis of Means
ANOVA	Analysis of Variance
AS/AR	Automatic Storage and Auomatic Retrieval
CAD	Computer-Aided Design/Drafting
CADSS	Computer Aided Decision Support System
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CAPM	Computer-Aided Production Management
CAPP	Computer-Aided Process Planning
CBR	Case-Based Reasoning
CE/CS	Concurrent/Simultaneous Engineering
CIM	Computer Integrated Manufacturing
CNC	Computer Numerical Control
DNC	Direct Numerical Contro
DSS	Decision Support System
EDI	Electronic Data Interchange
ERP	Enterprise Resources Planning
ES	Expert System
FCFS	Fist-Come First-Served
FFE	Full Factorial Experimentation
FIFO	First-In First-Out
FMC	Flexible Manufacturing Cell
FMS	Flexible Manufacturing System
FzL	Fuzzy Logic
GA	Genetic Algorithms
GCE	Global Concurrent Engineering
GT	Group Technology
HIS	Hybrid Intelligent Systems
ICAM	Integrated Computer-Aided Manufacturing
ICAM	Integrated Computer Aided Manufacturing

IDEF	Integrated DEFinition
IIS	Intelligent Information Systems
IMS	Intelligent Manufacturing Systems
IPS	Intelligent Photolithography Scheduling
JIT	Just In Time
JSS	Job Shop Scheduling
KBS	Knowledge-Based System
LAN	Local Area Network
LM	Lean Manufacturing
LPT	Longest Processing Time
LRPT	Longest Remaining Processing Time
MHS	Material Handling Systems
MPS	Master Production Scheduling
MRP	Material Requirements Planning
MRPII	Manufacturing Resource Planning
NC	Numerical Control
NN	Neural Network
OPT	Optimized Production Technology
PDM	Product Data Management
PPC	Production Planning and Control
QA	Quality Assurance
QC	Quality Control
SASM	Schematic Approach for Simulation Modeling
SPT	Shortest Processing Time
SRPT	Shortest Remaining Processing Time
SS	Shop Scheduling
TOC	Theory of Constraints
TPM	Total Productive Maintenance
TQM	Total Quality Management
VM	Virtual Manufacturing
WAN	Wide Area Network
WCM	World Class Manufacturing
WIP	Work In Progress (Process)
W-LTPT	Wafer with Longest Total Processing Times
W-STPT	Wafer with Shortest Total Processing Times

Glossary of Notation

Notation	Definition
$\overline{C}$	Mean completion time
σ_e^2	Variance of predicted error
π_r	Value of selection criteria 'r'
BS	Buffer Size
BT	Tool Buffer
c	Number of factors
C_{max}	Make-Span
CT	Cycle Time
DOF	Degree of Freedom
F_{cal}	The calculated value of statistical 'F'
F_{tab}	Tabulated value of 'F'
$GTSS$	Grand total sum of squares
J	Job
K_x	Binary variable (Flag) of x
l_j	Number of levels for factors j
L_x	Orthogonal Array of x experiments
M	Number of machines
m	Overall mean
MSD	Mean Square Deviation
n	Number of jobs
NL	Number of layers
N_{QT}	number of qualified tools for particular layer
O_{jk}	Operation of job k
P_{jk}	Processing time of job k
PM	Product-mix
PS	Product Sequence
QC	Quality Charateristics
S/N	Signal to noise ratio
SA	Scheduling agent-based
S_m	Score of tool 'm'
SSB	Sum of the sums of squares due to various factors
SSE	Sum of squares due to error
SST	Total sum of squares

TPT	Throughput Time
w_{ij}	Waiting time of job 'j' on machine 'i'
WS	Wafer Starts
w_{xxO}	Weighting if the criterion xx is not applied
w_{xxR}	Weighting if the criterion xx is right (ready)
X_{ij}	idle time of machine 'i' before start job in position j in the sequence
α	Confidence interval coefficient
η	Objective function
$\eta_{obs.opt}$	Observed optimum value
$\eta_{pre.opt}$	Predicted optimum value

Chapter 1

Introduction

Global competition and rapidly changing customer requirements are demanding numerous changes in manufacturing environment, heralding a new industrial era with new planning and control techniques, recognition of decision-making tools, and new approaches and ideas about the arrangement of manufacturing systems. Confronted with ever-increasing competition and new products increases, the product life-cycle time has become shorter, necessitating reduced development cycle time [1]. The need to shorten the loss zone and to reach the maximum profit phase in shorter time is becoming a challenging task for research, Figure 1.1.

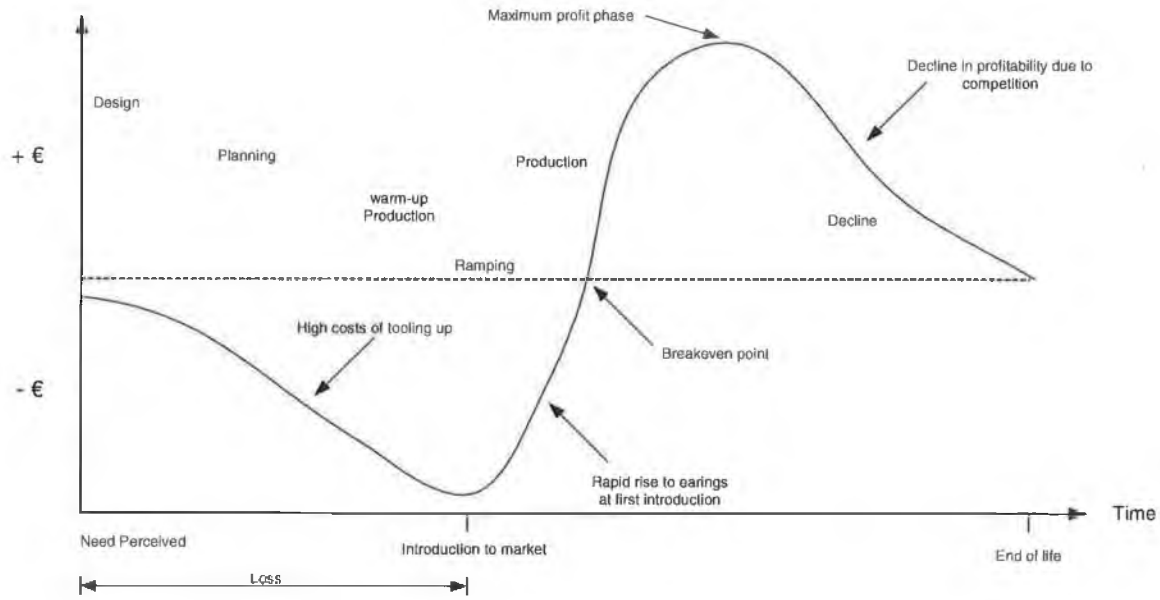


Figure 1.1: Product Life Cycle

With the advent of computer technology, much capital has been invested to explore new technologies such as computer integrated manufacturing and highly automated systems. Since errors within advanced manufacturing systems result in a greater cost, production planning and scheduling of these systems play a crucial role in driving the different components in an efficient manner. A large part of this role is associated with scheduling, one of the most critical industrial activities, which is concerned with balancing resource-demand problems.

Traditionally, many researchers have dealt with scheduling problems as an abstract mathematical model using relaxation algorithms and simplifications [2]. Interestingly, in a survey performed by Reisman *et al.* [3] based on 170 research articles regarding the scheduling domain between 1952 and 1994, it was found that 87.3 per cent of the non-survey papers dealt with static deterministic problems. Most of those models could not provide effective solutions to real-world applications. Confronted with the increasing automation level and the complexity of new technologies, scheduling is becoming evermore challenging task. Traditional solving techniques are no longer capable on delivering satisfying solutions in a reasonable time with respect to the short product life cycle. Heuristic and advanced solving approaches (e.g. simulation, AI) have contributed significantly in providing solutions that are more comprehensive. The effective use of these new approaches can increase the efficiency of the manufacturing systems.

1.1 Problem Definition

Semiconductor manufacturing is one of the most complex manufacturing processes in the world. Although the importance of the semiconductor industry is widely acknowledged, few researchers have addressed production planning and scheduling problems encountered in this environment. Nevertheless, scheduling of semiconductor manufacturing is still a problem area due to the dramatic increase in the number of devices on an IC (see Table 1.1), complex product flows, random yields and rework, time-critical operations, batching, simultaneous resource possession, and rapidly changing products and technologies. Typical wafers undergo hundreds of processing steps, reentering the same processing machines multiple times, as each layer is successively added. Often, some processes are skipped, repeated, or completed in a different order. This results in a complex, highly reentrant process flow in a flexible manufacturing environment that is impossible to schedule in an optimum manner manually.

Table 1.1: Levels of Integration in Microelectronic Industry [4]

Integration Level	Number of devices on a chip	Approximate year introduced
Small scale integration (SSI)	10 – 50	1959
Medium scale integration (MSI)	$50 - 10^3$	1960s
Large scale integration (LSI)	$10^3 - 10^4$	1970s
Very large scale integration (VLSI)	$10^5 - 10^6$	1980s
Ultra large scale integration (ULSI)	$10^7 - 10^8$	1990s

This thesis has focused on the most critical manufacturing area in wafer fabrication, photolithography, which is considered a high-risk process due to technology complexity and expensive equipment. As shown in Figure 1.2 , photolithography is a central process among the wafer fabrication processes. Appendix A provides more information about wafer fabrication processes and steps.

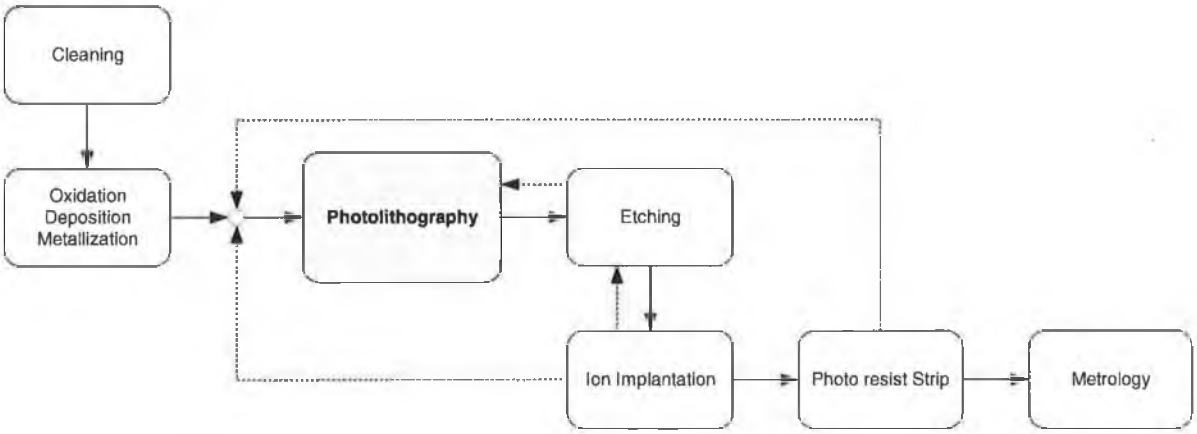


Figure 1.2: Advanced techniques for process improvement

The photolithography process is described as the most complex process in the semiconductor fabrication in terms of technology and procedure [196]. In the photolithography area, the structure of the circuits is mapped from the pattern on the mask to a wafer in a process similar to photographing. The system analysis is very complicated in the case of photolithography of wafer fabrication facilities due to the following:

- Complex technology

- Re-entrant process flows
- Random yields
- Product/Layer sensitivity (dependant set-up times)
- Expensive equipment
- Data availability
- High product-mix
- Rapid product turnover
- Maintenance problems
- Lots priority
- Non identical parallel machines
- Random scheduling
- Critical dimensions
- Complex metrology

These result in

- a low overall performance within photolithography tools
- less utilization/ higher cost
- more work in process inventory
- high risk planning area
- delay in products due dates/delivery
- an increase in throughput time
- an increase cycle time per wafer/lot
- defining photolithography as the factory bottleneck area

The previous problems together with the high capital cost of photolithography equipment have put more pressure on the planning staff to schedule the photolithography in an optimum manner. Added to that, the experimentation to reach a satisfactory solution is a non-option. Therefore, the need of a powerful decision support system to minimize production cost and increase productivity while improving both quality and due date delivery is urged.

The planning and control of semiconductor manufacturing systems is usually complex with a huge number of interrelationships between cells and units. The photolithography is even more complex due to those factors mentioned earlier. The approach which we used in chapter four and five that accommodate such complexity is that of breaking the problem into a hierarchy, which emphasizes the interrelationship of the manufacturing processes. Once the problems have been broken into some specific sub-problems such as wafer arrival, buffer capacity etc., there are several methods that can be used to model the integration of these sub-problems. In this thesis, two main approaches (IDEF, and SASM) were used to carry out the modeling task. Then, the simulation models were run iteratively under different scenarios for the sake of verification and validation.

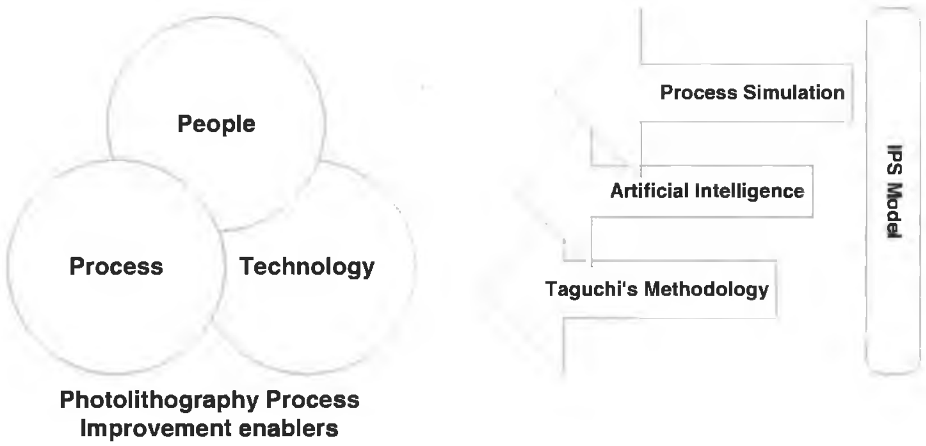


Figure 1.2: Advanced techniques for process improvement

The hypothesis of this thesis is to propose a decision support approach to solving scheduling problems in the photolithography manufacturing area in semiconductor facilities. A methodology to direct the proposed approach is presented in the context of the shop scheduling domain. A novel modeling approach, Schematic Approach for Simulation Modeling (SASM), to handle the complexity of photolithography area has been developed. Simulation has been used into a comprehensive manner to build robust models (e.g. FMC model, IPS model) that mimic the factory floor in a satisfactory level of detail. The intelligent Photolithography Scheduling model (IPS) provides promising assistance in decision making for the photolithography area. It has yielded important increases in both solution efficiency and schedule quality over a variety of competing existing techniques.

1.2 Aims of the thesis

This thesis investigates a new scheduling approach, which combines three effective solving techniques (i.e. simulation, artificial intelligence, and Taguchi's optimization methodology) in order to provide quality scheduling at a low cost. Additionally, these techniques have been used separately to deal with scheduling problems, but the key feature of the research that will be discussed lies in using their abilities to dynamically adapt while solving the hard scheduling problems in photolithography area. The proposed approach aims to provide an easy-use efficient technique to support decision makers. More specifically, the research objectives are successively presented as follows:

- To develop risk assessment model(s) to evaluate the photolithography tools performance.
- To characterize flexible manufacturing tools of photolithography manufacturing area in semiconductor fabrication using state-of-the-art simulation models.
- To provide an efficient scheduling approach that can examine the impact of various production plans on the performance of photolithography toolsets.
- To demonstrate the feasibility of integrating simulation based models and AI techniques within Taguchi's paradigm to provide effective scheduling model.
- To find an efficient schedule for lot sequence in every photolithography toolset.
- To enhance the toolset performance (e.g. increasing tool utilization, and decreasing throughput time and cycle time).
- To assist detecting bottlenecks (constraint equipment/tools) in order to avoid building work in process (WIP).
- To use the neural network ability to design a model that is capable of learning and dynamically generate near optimum schedules.

In addition, the thesis presents a comprehensive review on shop scheduling problems and their many solving techniques. A general flow shop scheduling model has been solved using traditional and heuristic techniques in order to analyze and compare features of these techniques.

1.3 Thesis Outline

The hierarchy of this thesis is shown in Figure 1.3. Chapters 2 – 6 form the main body of thesis, while eight appendices provide background information.

Chapter 2: This chapter introduces manufacturing systems under market pressures due to global competition and the applications of new technologies. It gives a brief classification of manufacturing systems based on production type, automation level, and production flow. It then goes on to discuss scheduling problems in manufacturing starting with different planning levels, followed by concise discussion about shop scheduling, i.e. objectives, characteristics, and different models. Solving techniques to shop scheduling problems are classified into two main groups; traditional and advanced techniques. Moreover, the gap between theory and practice is discussed. There are three appendices attached to this chapter. Appendix A provides a review on semiconductor manufacturing and main processes in wafer fabrication.

Chapter 3: This chapter reviews a broad range of literature, which has relevance to the areas of research in this thesis. Many tools have been used in scheduling of semiconductor manufacturing systems such as system modelling approaches and simulation. Simulation has received more attention in literature and therefore, Appendix C provides a comprehensive review of simulation applications in manufacturing systems and shows a new methodology for selection and evaluation of simulation software packages. Intelligent scheduling in semiconductor manufacturing has been introduced in this chapter in more detail with a review of past research. Finally, a methodology for intelligent scheduling using simulation, intelligent-agent based, and Taguchi is proposed.

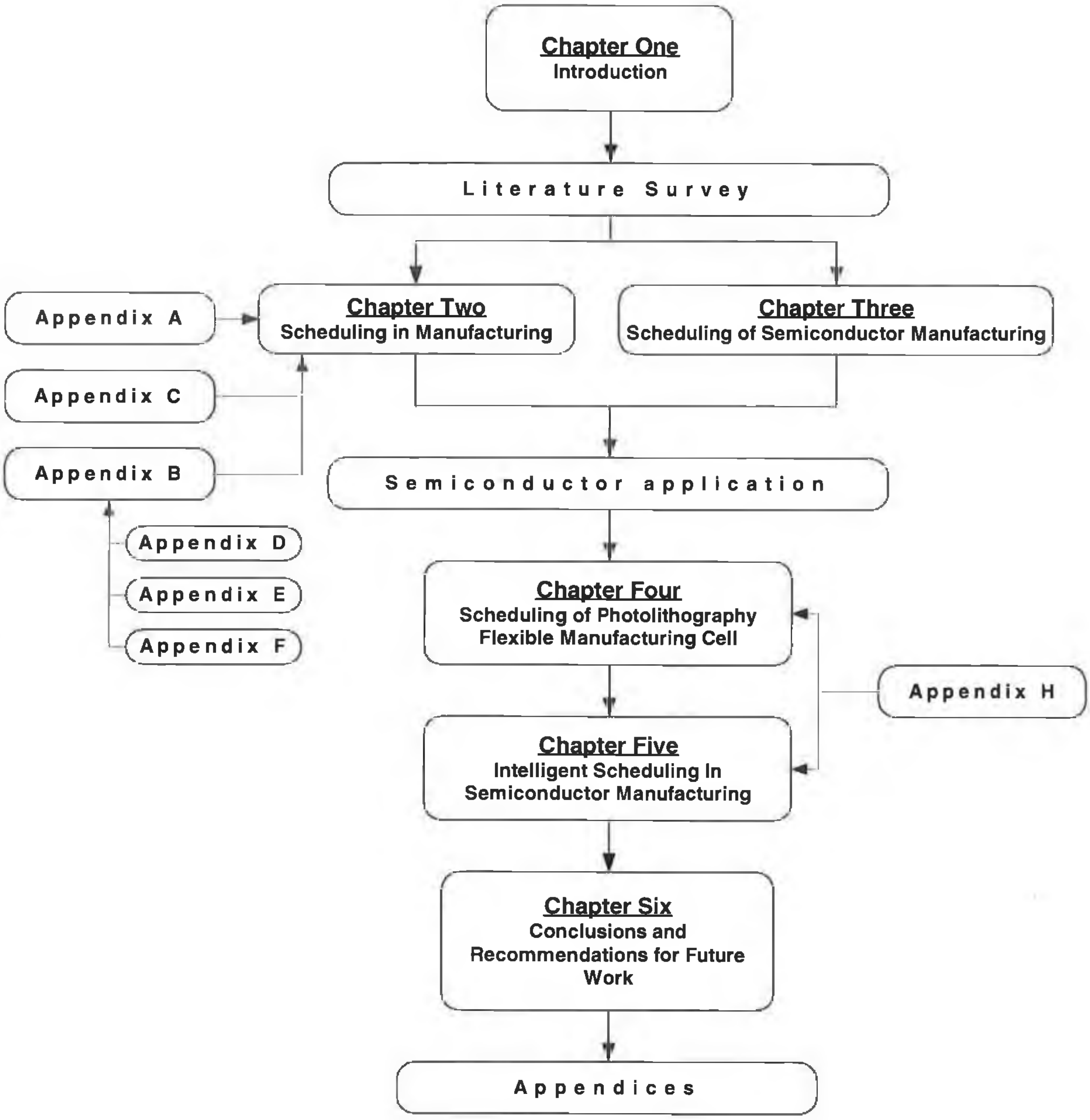


Figure 1.3: Thesis Outline

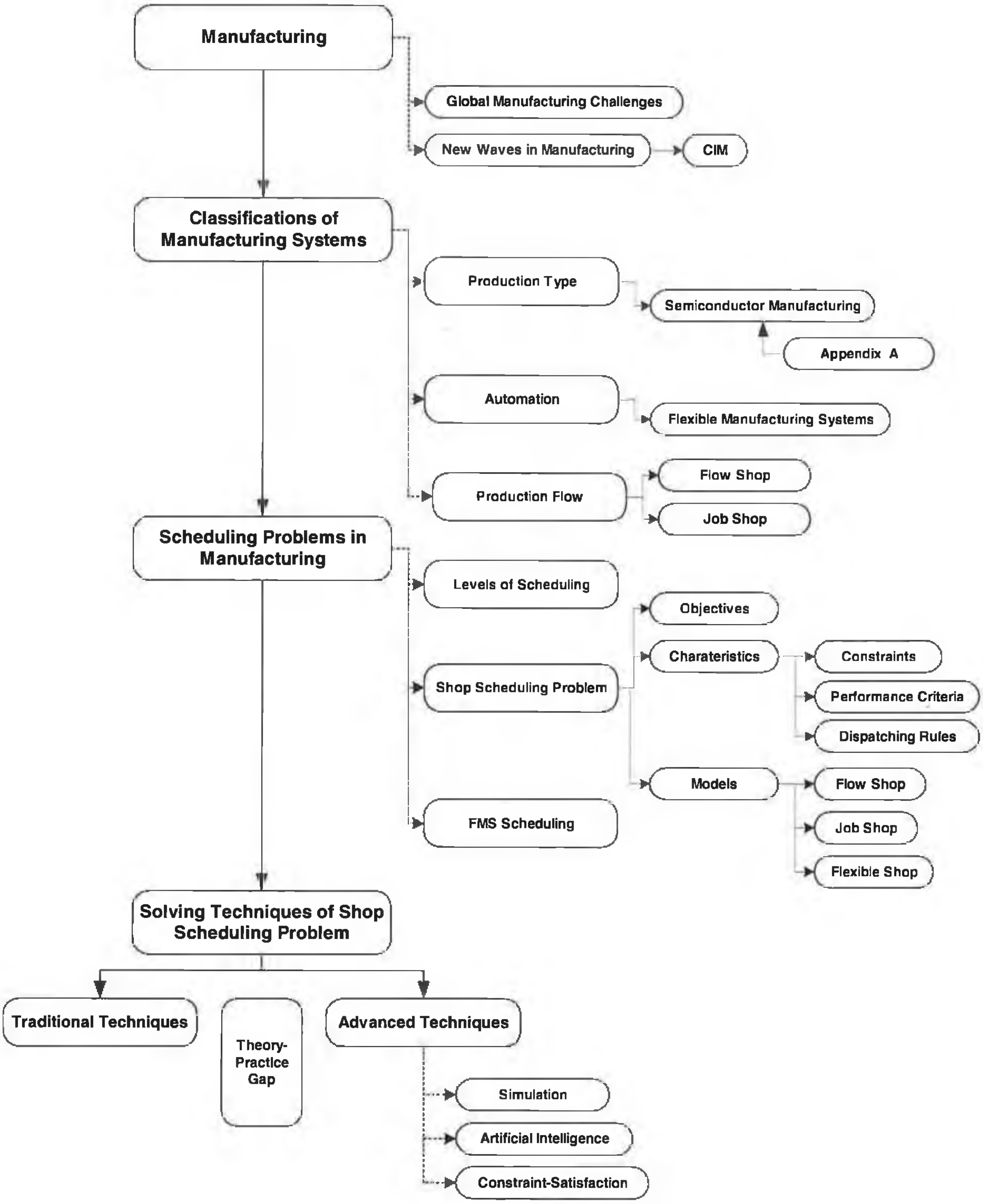
Chapter 4: The scheduling of a flexible manufacturing cell (FMC) in photolithography has been solved using a generic simulation model combined with Taguchi's methodology for experimental design and optimizing selected scheduling parameters. The model also uses the system approach IDEF0. Sensitivity analyses have been conducted in order to provide recommendations for the manufacturing team.

Chapter 5: This chapter describes the IPS Model for a parallel cluster of FMC's and is an expansion of the photolithography FMC model. A hybrid intelligent scheduling methodology that integrates simulation, the intelligent agent-based approach, neural networks, and Taguchi's methodology has been explored. Results obtained, using this methodology, have been used for scheduling actual production. This chapter presents the intelligent agent-based approach used to optimize the model.

Chapter 6: The final chapter contains conclusions and recommendations for future research work.

Chapter 2

Scheduling in Manufacturing



Chapter 2

Scheduling in Manufacturing Systems

2.1 Introduction

Increasing global competition has made many business leaders and policy makers turn their attention to new technologies in order to remain competitive [5]. The purpose of this chapter is to give a brief introduction to manufacturing systems and scheduling approaches in four sections. Firstly an overview of the pressures imposed on manufacturing systems and the new technologies applied in last four decades is presented. Section 2 discusses the classification of manufacturing systems based on production type, automation, and production flow. Section 3 defines scheduling problems in different planning levels in manufacturing systems and in particular reviews shop scheduling problems which are a function of the manufacturing system being considered. Finally, section 4 presents a comprehensive classification of shop scheduling solving techniques including traditional and advanced techniques.

2.2 Manufacturing

Manufacturing can be defined two ways, one technological and the other economical [4]. Technologically, manufacturing is carried out as a sequence of operations to be done to make parts or products. Each operation brings the material closer to the desired final state. Economically, manufacturing is the transformation of materials into items of greater value by means of one or more processing and/or assembly operations. The key point is that manufacturing adds value to the material by changing its shape or properties or by combining it with other materials that have been similarly altered.

Manufacturing consists of a set of processes and systems (and people, of course) designed to transform a certain limited range of materials into products of increased value (Figure 2.1).

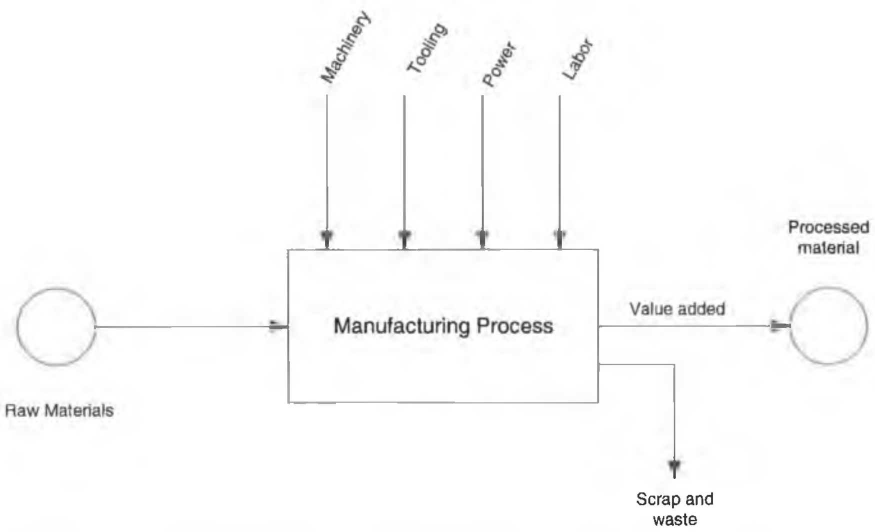


Figure 2.1: Manufacturing definition according to Groover [4]

Deuermeyer (1994) [4] suggested a comprehensive definition of manufacturing systems:

"*A manufacturing system is an objective-oriented network of processes through which entities flow.*"

The major objective of manufacturing system is to making profit [4]. The system contains processes, which may include the usual physical processes (machining, packaging, etc.) but can also include other steps that support the direct manufacturing processes (order entry, maintenance, etc.). Entities include not only the parts being manufactured, but also the information that is used to control the system. The flow of the entities through the system describes how materials and information are processed. Management of this flow is a major part of a manufacturing manager's job. Finally, it is important to recognize that a manufacturing system is a network of interacting parts. Managing the interactions as well as managing individual processes and entities is a crucial task. This definition of manufacturing system serves to highlight the roles of the different disciplines that deal with manufacturing. As Hopp *et al.* [6] explained: mechanical and electrical engineering staff deals principally with manufacturing processes and the design of the entities (products), while industrial engineering team focuses on the flows and the network. Management is concerned with ensuring compliance with the objective, keeping harmony between activities, and measuring progress towards the goals.

2.2.1 Global Manufacturing Challenges

Global competition and rapidly changing customer requirements are demanding increasing changes in the manufacturing environment. Today's business climate for manufacturers requires high quality, high flexibility, and quick response systems that turn out a wide variety of product configurations. Confronted with ever-increasing competition, companies need to produce parts of higher quality, at lower cost, with the shortest possible lead times. To complicate matters further, increasing rate of technological change has caused product life cycles to be dramatically reduced. A product reaching the maturity stage of its life cycle may become much less common. Consequently, industry has recently accommodated numerous changes, heralding a new industrial era with new advanced planning techniques, new ideas about the arrangement of manufacturing systems, and the recognition of the decision-making powerful techniques the manufacturing system.

With the advent of computer-technology, The need of effective techniques to deal with scheduling and planning of manufacturing systems, which involve very large capital costs and complex interactions, becomes an urgent request.

2.2.2 New Waves in Manufacturing

A significant difference between Western and Japanese companies, observed by Shigeo Shingo [7], is that Western companies generally implement improvements as a step change and by the application of technology, whereas the Japanese change incrementally and continually, and they generally involve people. The Japanese approach of continuous improvement will also typically involve low cost and high technology improvements. Manufacturing trends or innovations over last decades are shown in Figure 2.2.

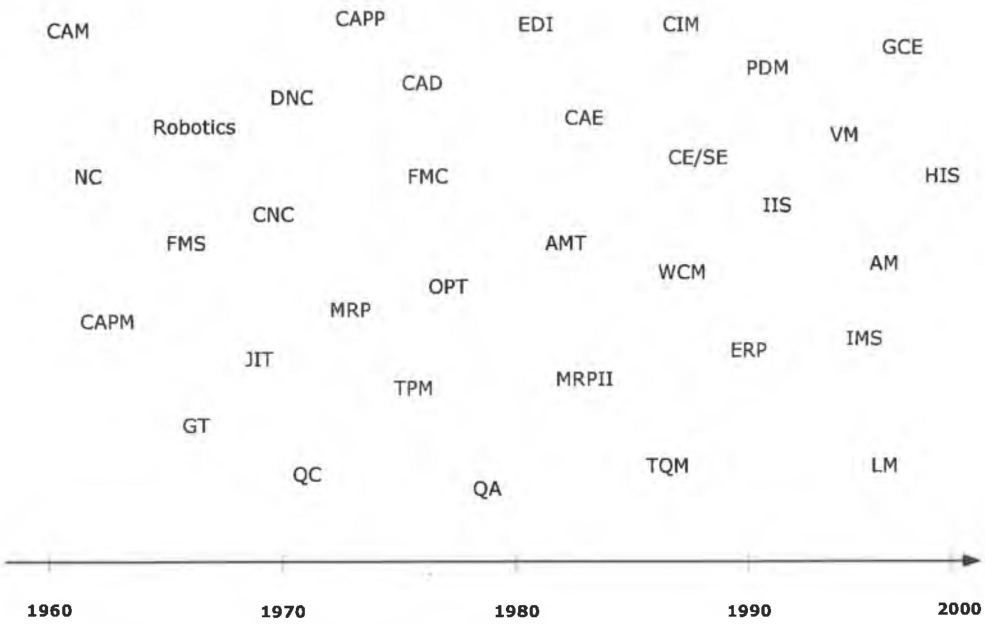


Figure 2.2: New trends in manufacturing systems

Most research and development efforts towards automation for modern manufacturing have been independently developed [8]. At the present time, highly automated areas in manufacturing (e.g. Computer-Aided Design (CAD), Computer-Aided Process Planning (CAPP), and Computer-Aided Manufacturing (CAM)) have been integrated to provide a Computer Integrated Manufacturing (CIM) environment, Figure 2.3.

Computer Integrated Manufacturing (CIM) is a management philosophy in which the functions of design and manufacturing are rationalized and coordinated using computer, communication, and information technologies (Bedworth *et al.*) [9]. CIM has the capability to largely, or entirely, automate manufacturing by coordinating work cells, robots, automatic storage and retrieval facilities (AS/AR) and material handling systems. To schedule operations successfully in the continuous state of flux which flexible manufacturing imposes, the controlling systems need some "look-ahead" capabilities such as predictive scheduling. A vast body of literature (e.g. [10][11][12][13]) has discussed CIM in detail.

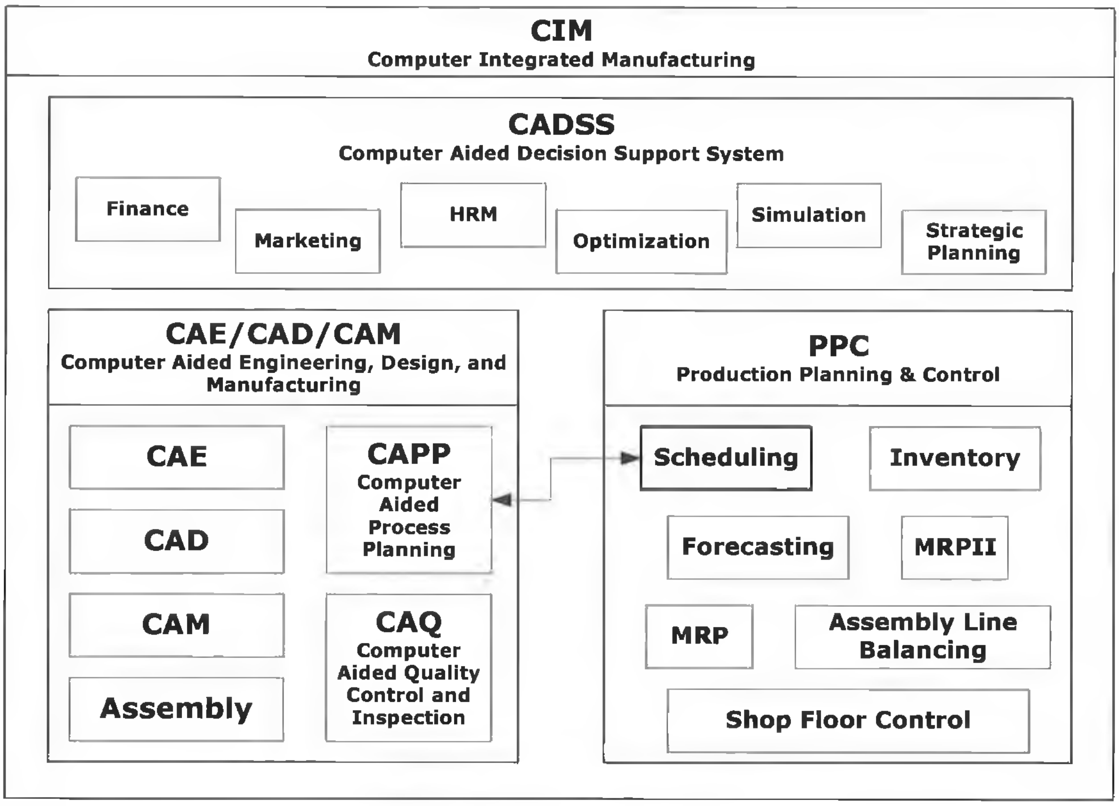


Figure 2.3: Computer Integrated Manufacturing Components [14]

2.3 Classifications of Manufacturing Systems

Manufacturing Systems (MS) can be classified based on many various schemes. MS can be grouped into classes based on the production type, production volume, production flow, production layout, or automation level. Table 2.1 summarizes main classifications of production systems based on literature review (e.g. Groover [4], Zarembra *et al.* [8], and ISIC [15]). The key types of production systems discussed in the research are emphasized in italics. The following sections briefly highlight these manufacturing systems (e.g. semiconductor manufacturing).

Table 2.1: A summary of production system classifications

Classification based on	Types	Notes
Production Volume	Small-quantity production	Production range (1 to 100 units/year), Job shop style
	Medium-quantity production	Production range (100 to 10000 units annually), complexity increases with product-mix increases
	High-volume production	Production range from (10000 to millions units annually), mass production style
Production Flow	***Job Shop production***	Different set of jobs on different machines, and order or sequence is prescribed.
	Batch production	Product variety is hard, batches for every type of product, and mostly repeated orders
	Flow/line/mass production	High volume of product, stable design and demand required, and transfer lines are typical.
	Mixed production	Mix between any two of previous, flexible system, and productivity is function of order.
Layout	Fixed position layout	Large and complex industries (e.g. aircrafts, space ships)
	Process layout	Grouping the equipment based on the function, (e.g. lathes in one department, welding in another), Material handling is the disadvantage.
	Cellular layout	Similar to flexible manufacturing cells and group technology style.
	Product layout	Layout arranged into the operations sequence of the product, normally one long production line (e.g. assembly, packaging)
Product Type	Automotive production	Complex automated manufacturing with high capital of investment.
	Semiconductor Manufacturing	
	Electronic industry	
Automation	Level 0 (manual) : : ***Level 6 (FMS)***	Production systems can be categorized based on the level of automation. There are many standards to determine the automation levels.

2.3.1 Semiconductor Manufacturing

The miniaturization of electronic components by means of Very Large Scale Integration (VLSI) technologies has been one of the most significant technological developments of the last fifty years. Improving technologies and decreasing prices have led to integrated circuits appearing in all walks of life, see Figure 2.4.

The computer revolution of the past two decades is a direct result of the ability to develop and fabricate these components economically.

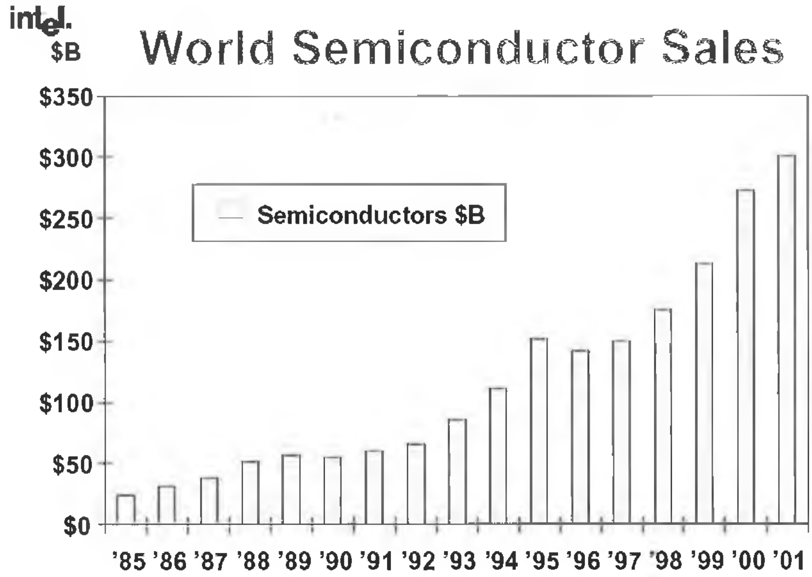


Figure 2.4: World Semiconductor Sales (www.intel.com)

The development of computer integrated manufacturing systems has a significant effect in the maintenance of competitiveness edge in today's highly competitive global markets.

Semiconductor manufacturing is one of the most complex manufacturing processes in the world [16] with random yields and rework, complex product flows, time-critical operations, batching, simultaneous resource possession, and rapidly changing products and technologies. Typically, wafers undergo hundreds of processing steps, reentering the same processing machines multiple times, as each successive layer is added. Often, some processes are skipped, repeated, or completed in different order. This results in a complex, highly reentrant process flow that is difficult to be scheduled manually in an optimum manner. A quick look at the manufacturing process might help to show the complexity of this industry.

The process by which integrated circuits are manufactured can be divided into four basic steps: wafer fabrication, wafer probe, assembly or packaging, and final testing as shown in Figure 2.5.

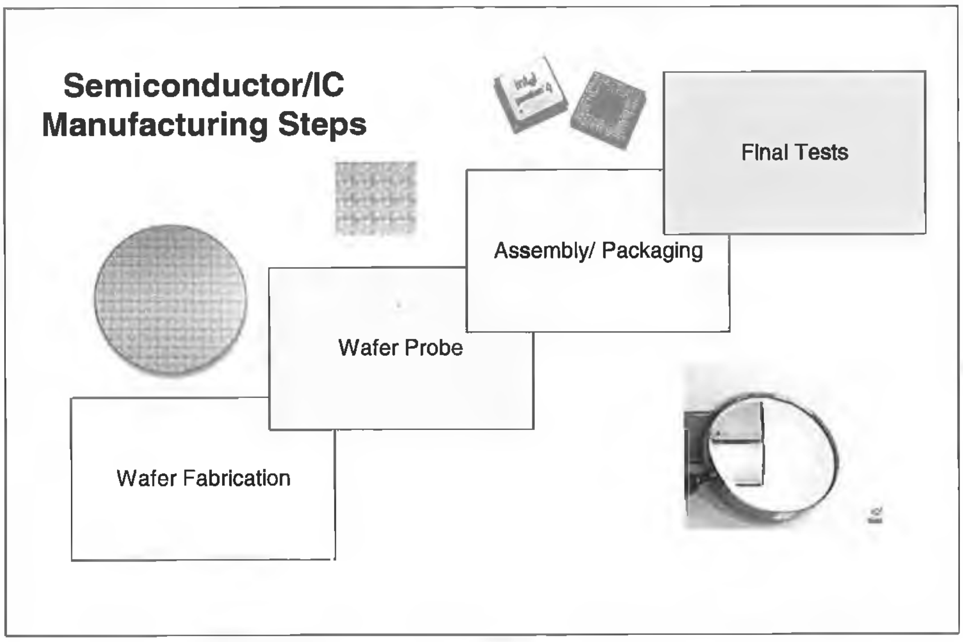


Figure 2.5: Basic Steps in very large-scale integrated circuits manufacturing

The front-end steps are manufactured in the system under study while the back-end processes are done overseas for economic reasons (i.e. cheaper labors, less tax). A description of the four basic steps is presented in Appendix A. A jigsaw diagram in Figure 2.6 shows the main manufacturing areas in wafer fabrication.

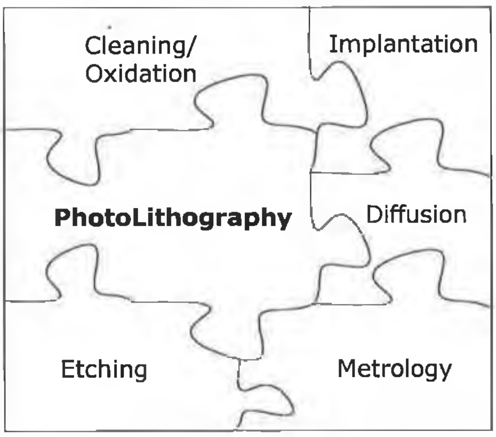


Figure 2.6: wafer fabrication jigsaw

Although the industry is intensely competitive, only recently has attempts been made to apply industrial engineering and operations research technologies to the operational aspects of semiconductor manufacturing [17]. The escalating cost of semiconductor manufacturing in the last decades, Figure 2.7, focused research on manufacturing strategies in semiconductor manufacturing to minimize the production cost and increase productivity while improving both quality and delivery on-time performance.

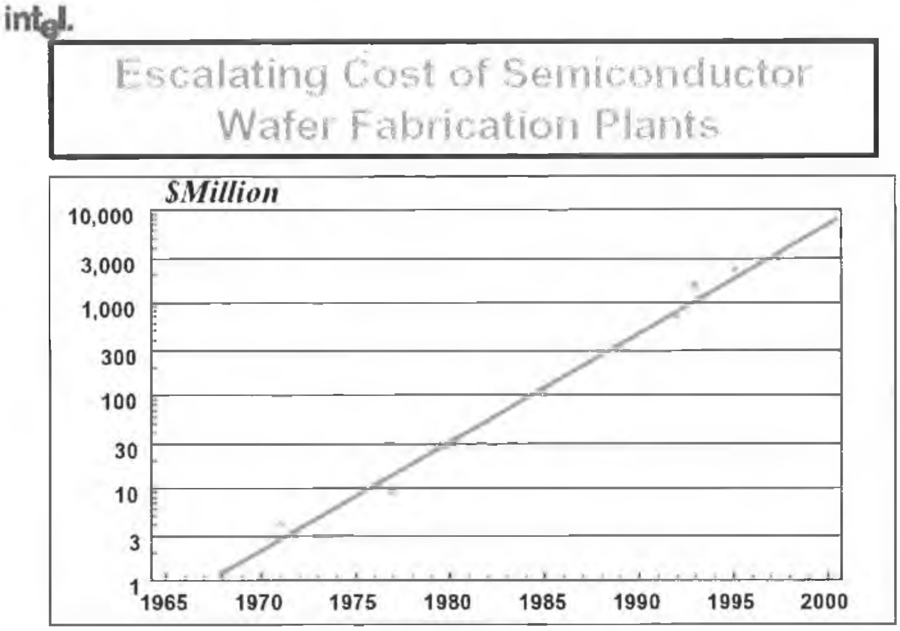


Figure 2.7: Escalating cost of semiconductor wafer facilities (www.intel.com)

Many researchers have contributed in the semiconductor development in areas such as product design (e.g. Avram *et al.* [17]), chip allocation (e.g. Heath *et al.* [18]), quality control (e.g. Phdake [19]), and implementation of JIT manufacturing (e.g. Martin-Vega *et al.* [20]). This research concentrates on production planning and scheduling applications towards process improvement (optimization) in the semiconductor industry focusing in particular on the photolithography process.

Automation plays a crucial role in semiconductor manufacturing [21]. The level of automation and the quality of scheduling have a significant impact on the general performance of semiconductor fabrication in terms of productivity and quality. Therefore, flexible manufacturing systems are considered the essential element of semiconductor manufacturing. The coming section discusses FMS in more detail.

2.3.2 Flexible Manufacturing System

Flexible Manufacturing Systems (FMS) are the heart of semiconductor manufacturing as they include at least 90% of the semiconductor equipment. Despite all the interest in flexible manufacturing systems, there is no uniformly agreed definition of the term FMS. The main feature which distinguishes an FMS from traditional manufacturing systems is "flexibility" [22] which does not have a precise definition. In fact, the scope and variety of flexible manufacturing are commonly disputed and are the focus of many research efforts.

Ranky [23] defines an FMS as a system dealing with high level distributed data processing and automated material handling flow using computer-controlled machines, assembly cells, industrial robots, inspection machines and so on, together with computer integrated material-handling and storage systems. This production technology has been designed to attain the efficiency of well-balanced machine paced transfer lines, while utilizing the flexibility that job shops have to simultaneously machine multiple part types.

To summarize the components and characteristics of an FMS, as described by different authors and researchers, are as follows [24]:

- Potentially independent NC machine tools.
- An automated material-handling system.
- An overall method of control that co-ordinates the functions of both the machine tools and material handling system so as to achieve flexibility.

Technical descriptions and definitions of FMSs can be found in Hopp's text (Hopp *et al.* [6]). Kaighobadi *et al.* [24] also defined FMSs, reasons for change from conventional systems to FMSs, application issues, and problems of FMSs. Rachamadugu *et al.* [26] and Liu *et al.* [25] have classified exisiting scheduling procedures based on key factors such as the FMS type, and scheduling environment, while Browne *et al.* [27] focused on the advantages arising from utilizing processing, routing, and other flexibilities.

The characteristics that distinguish FMS from conventional equipment are summarized in Table 2.2. The main features which bring flexibility are the integration, mechanization and re-programmable automation of operations (processing, material handling, and tool change), technical flexibility, complexity, regulation and expense.

Table 2.2: Characteristics of FMS Technology

Characteristics	*Definition*
Integration	The extent to which the system is able to perform different types of operations, to change tools, to transfer and load workpieces, and to integrate with other programs and production schedules through LAN or any other media.
Technical flexibility	The ability to quickly change mix, routing and sequence of operations within the part groups
Mechanization and re-programmable automation	The degree to which operations such as workpiece transfer, loading/unloading and fixturing, tool change, machine tool control, cutting tool control and inspection, are performed by the system, with minimum or without human intervention.
Complexity	The number of inter-related elements comprised in the system, such as operating units, material-handling system, control system, and other elements.
Regulation	The extent to which the system regulates the work of operators, process planners, production planners, maintenance engineers and other personnel.
Expense	The cost incurred in the investment in, and the operation, maintenance and operational management of the system. Moreover, the technology updating is effective factor in saving expenses.

One of the main characteristics that can significantly affect flexible systems which used in complex industries such as semiconductor manufacturing is the FMC/FMS life cycle (Figure 2.8). In the first stage, the flexible manufacturing unit is put into production with the low volume to examine the output in order to avoid the risk of the high investment cost of the flexible manufacturing unit. Ramping of the production volume is then based on the performance of the unit in the first phase of production. During these two stages, the process modifications and system configurations changes are less predictable, and hence these stages require a flexible rather than a dedicated process. Once these stages are successfully done, the maturity stage of the FMS takes place with high production potential of the production unit in order to maximize the capacity utilization. The product technology holds fixed during this stages however, each change in the product mix produced by the system causes a certain amount of delay and has an impact on process planning. Due to the high investment cost of the production unit, which urges companies to make the best use of available capacity, the cell might be used in less capacity. The product technology might keep fixed for the production unit although the process configurations might change. The life cycle time of the flexible manufacturing unit varies in range from two to five years based on many parameters such as manufacturing type, nature of product, technology.

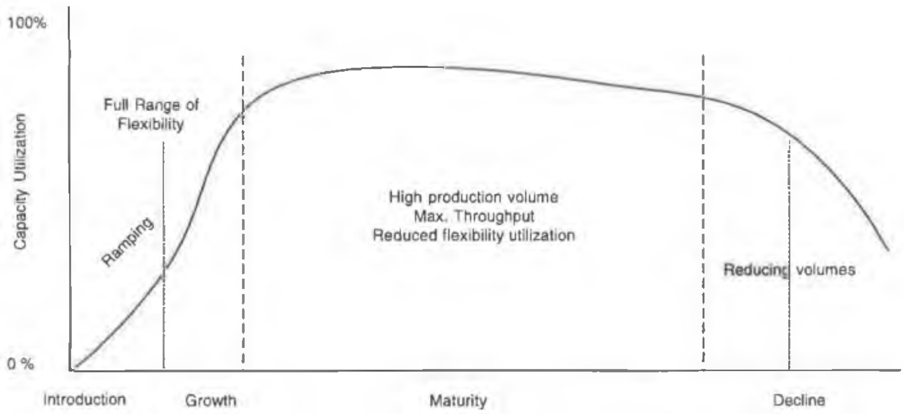


Figure 2.8: Flexible Manufacturing Systems life cycle

2.4 Scheduling Problems in Manufacturing

Scheduling is a decision making process that plays an important role in most manufacturing and service industries. The scheduling function optimizes limited resource allocation to the processing of jobs. Resources can be machines, materials to be processed, or staff at the work center. Jobs may be operations in workshop or tasks (maintenance or function) to be done. Each job may have a priority level, an earliest possible starting time, and a due date. The optimization objectives may also take many forms, such as minimizing the time to complete all jobs, minimizing the number of late jobs, and so on.

The role of scheduling in a generic manufacturing environment is crucial. Customer orders are translated into jobs with associated due dates in a manufacturing setting. These jobs often have to be processed on the machines in a work center in a given order or sequence. The processing of jobs may be delayed if certain machines are busy, and preemption may occur when high priority jobs arrive at machines and have to be processed at once. Unexpected events on the shop floor, such as machine breakdowns or longer-than-expected processing times, also have to be taken into account, since they may have a major impact on the schedules. The scheduling system makes decisions dynamically about matching activities and resources in order to finish jobs and projects needing these activities on time and with adequate quality fashion while simultaneously maximizing throughput and minimizing direct operating costs by increasing utilization. Developing a detailed schedule of the jobs to be performed helps maintain efficiency and control of operations. For example, classical scheduling has a set of basic decisions to be made include:

a. Sequencing,
b. Timing/release, and
c. Routing.

Added decisions for extended scheduling models include:

d. Resource reconfiguration and
e. Activity reconfiguration.

2.4.1 Levels of Scheduling

The theory of scheduling is characterized by a virtually unlimited number of problem types. Basic classification for the scheduling problem and comments in this section are based on vast body of literature (e.g. Blazewicz *et al.*[28], Conway *et al.* [29], French [30], Lenstra [31], Pinedo [32], Rinnoy Kan [33], Tanaev *et al.* [34], Tanaev *et al.* [35], Herrmann *et al.* [36]).

Scheduling, in the field of production, appears in many facets characterized by several criteria. Table 2.3 summarizes common scheduling types of manufacturing/production systems.

Table 2.3: Different Types of Scheduling Problem

Classification based on	*Scheduling Levels*
Production Volume	- High Volume Scheduling - Intermediate Volume Scheduling - Low Volume Scheduling
Nature of Production	- Activity Scheduling - Batch Scheduling - Network (Project) Scheduling
Production Capacity	- Infinite Capacity Scheduling - Finite Capacity Scheduling
Manufacturing System	- Flow shop Scheduling (Transfer lines). - Job-shop Scheduling. - Flexible Manufacturing System Scheduling.
State of Scheduling	- Static Scheduling - Dynamic Scheduling (Reactive)

The scheduling may be at different levels of detail and realism depending on the need. At a high level, a manufacturing facility may be considered as a single resource, while local scheduling must deal with the detail within the facility. This may lead to a classification of manufacturing planning at several levels. Most schemes have four or five levels. However, time horizon classification can be employed on these types of scheduling. Table 2.4 shows one classification with five levels of scheduling based on time horizon. All these levels can be described as scheduling, in that they have issues of time objectives and reconfiguration decisions.

Table 2.4: Classification of Scheduling Levels Based on Time Horizon [37]

Level		*Examples*	*Horizon*
I	Long-range planning	Plant expansion, plant layout, plant design	2 – 5 years
II	Medium-range planning	Production smoothing, logistics	1 – 2 years
III	Short-range planning	MRP, shop bidding, due date setting	3 – 6 months
IV	Scheduling	Job shop routing, assembly line balancing, process batch sizing, floor scheduling	2 – 6 weeks
V	Reactive scheduling / control	Hot jobs, down machines, late material	1 – 3 days

While this research focuses on shop scheduling level (level IV in Table 2.4), in practice, many scheduling levels can be merged together to characterize a single manufacturing environment.

2.4.2 Shop Scheduling

Optimizing product mix and resource allocation based on inventory levels, demand forecasts, and resource requirements are major aims of higher planning level in manufacturing systems and decisions made have affect the scheduling process directly.

Shop scheduling is one of the most critical levels of scheduling because it is the linkage between the tactical planning level and operational level. It is driven, as shown in Figure 2.9, by decisions taken in the medium planning levels as well as long term planning.

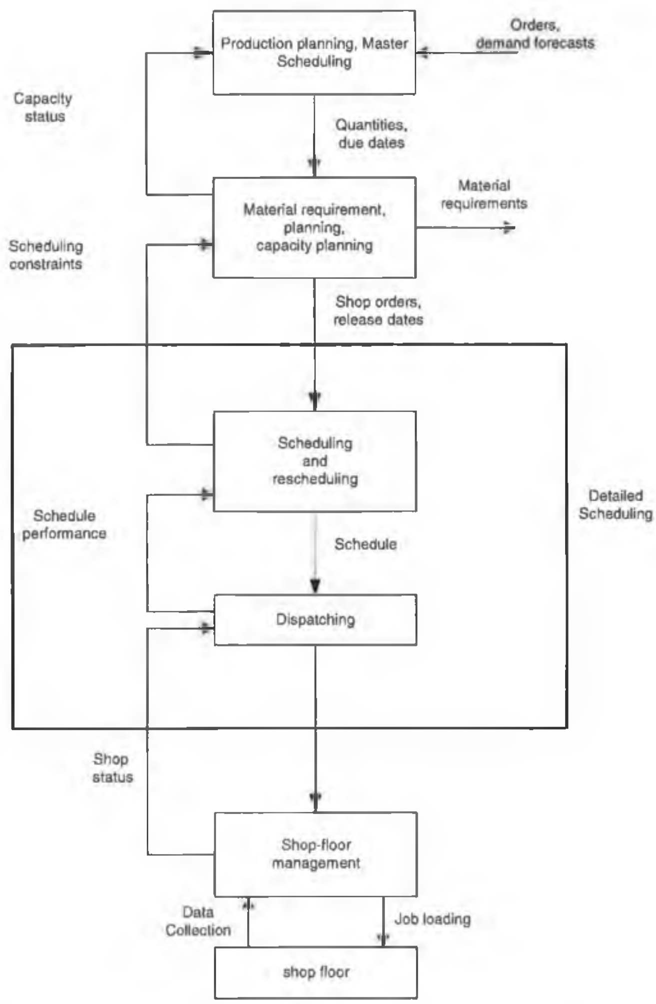


Figure 2.9: Information flow in a generic manufacturing system [38]

The main activities are; dispatching, loading, and resource allocation. Shop scheduling has to interact with other decision-making procedures used within the plant. One popular system that is widely used is the material requirement planning (MRP) system. After a schedule is set up, all raw materials and resources need to be available at the specified times to perform the operations. The due dates of all jobs have to be determined jointly by the production planning and scheduling system and MRP system. Figure 2.9 depicts the information flow in a generic manufacturing system [38].

2.4.2.1 Shop Scheduling Objectives

Virtually all manufacturing managers want on-time delivery, minimal work in process, short customer lead times, and maximum utilization of resources. Unfortunately, these goals can conflict, e.g. customer lead times can be made essentially zero if an enormous inventory is maintained, or maximum utilizations are obtained with lower resources, and so on. The goal of shop scheduling is to strike a profitable balance among these conflicting goals [6].

The basic objectives of shop scheduling for real-world systems can be summarized as;

1. Meeting Due-Date

   Goal: Basic goal of production schedule is to meet due-date.

   Measure: Lateness and Tardiness are measures of distance from the goal.

   Influence: Production type (make-to-order, make-to-stock) is considerable factor.

2. Maximizing Utilization

   Goal: Cost accounting encourages high machine utilization, as higher utilization means higher return on investment.

   Measure: Make-span is the closest product based measure of utilization.

   Influence: Factory physics promotes high utilization, provided cycle times, quality, and service are not degraded excessively.

3. Reducing Work in Process (WIP) and Cycle Times (CT)

   Goal: Minimize WIP and CT in order to speed up the production flow.

Measure: WIP is often easier to measure, e.g. waiting and idle times, while overall cycle time is hard to measure in complex manufacturing systems.

Influence: There are some factors can keep cycle time short including: better responsiveness to the customer, maintaining flexibility, improving quality, relying less on forecasts, and making better forecasts.

2.4.2.2 Shop Scheduling Problem Characteristics

A) <u>Problem Constraints</u>

Most of real-world scheduling problems fall into the NP-hard category and tend to have many constraints. Pinedo [38], Brucher [39], and French [30] have discussed many of the constraints imposed on shop scheduling process. Briefly, these constraints can be:

- Precedence
- Routing
- Storage-Space and Waiting Times
- Preemption
- Machine Eligibility
- Production Plans
- Setup
- Material Handling
- Personnel Scheduling
- Tooling
- Resource
- Capacity
- Industry type
- Automation
- Economic
- Demand Pattern

In addition, the scheduling function in any organization or system has to interface with many other functions. These interfaces are system dependent and may differ substantially from one situation to another. For example, the scheduling environment may be defined as static or dynamic (scheduling parameters status), preemptive or non-preemptive (production activities), regular or special (objectives), and deterministic or stochastic (information type), see Appendix B. That adds more complexity on finding the optimal schedule.

B) *Performance Criteria*

The terms objective criteria, performance criteria or measures of performance are often employed with similar meanings and their use depends on the application and the model type. For example, make-span is preferable in case of flow shop and job shop although it is not in single machine scheduling models [40].

Many different types of objectives are used in evaluating production scheduling. In practice, the overall objective is often a composite of several basic objectives. The most important objectives lay in a shop scheduling objectives framework as follows:

- Criteria based on minimizing due dates
- Criteria based on maximizing utilization
- Criteria based on minimizing throughput time
- Criteria based on customized objectives

Panwalker *et al.* [41] presented a survey on performance measures, and Demirkol *et al.* [42] presented extensive sets of randomly generated test problems for establishing benchmarks for shop scheduling problems. Pinedo [32] has addressed criteria related to flow shop and job shop problems in detail in his book. Rather than go through the full range here, Arisha [40] describes the performance criteria in more detail.

C) Dispatching Rules

Dispatching, in the production context, is a procedure that uses logical decision rules to select a job for processing on a machine that has just become available [43]. These rules determine the value of a priority attribute that is assigned to each job, calculated as a function of such parameters as processing time, due date, the length of the queue in which the job is waiting, and the length of the queue at the next machine on the job's route. The particular parameters are based on the particular application and attributes assigned to the operations and jobs.

Dispatching rules can be categorized in six different groups:

I. Simple Priority Rules.
II. Combination of simple Priority Rules.
III. Local and Global Rules
IV. Heuristic Scheduling Rules.
V. Static and Dynamic Rules.
VI. Other Rules.

Research to study dispatching rules has been active for more than three decades; a comprehensive survey done by Panwalker *et al.* [41] has included over 100 dispatching rules. He listed the most common simple priority rules and their combinations. From different point of views, French [30] has classified the rules as static priority rules and dynamic priority rules based on the time function and the effect of passing time over the rules. Conway *et al.* [29] classified them as local priority rules (only information about the jobs to be processed on a particular machine), while global rule gives more information about jobs, machines, queue lines, and cost. It is impossible to identify any single rule as the best in all circumstances [44]. Outstanding surveys on shop scheduling have investigated the dispatching rules along with the shop scheduling problem from a variety of perspectives such as Jain *et al.* [45], Herrmann [36], and Blazewicz *et al.* [46]. Table 2.5 shows a sample of dispatching rules.

Table 2.5: A sample of dispatching rules and their definitions

Category	*Rule Symbols*	*Definition of rules*
Simple Priority Rules	SPT	Shortest Processing Time
	LPT	Longest Processing Time
	FCFS	First Come First Serve.
	LCFS	Last Come First Serve.
Combination of Simple Priority Rules	FCFS/SPT	Select jobs based on SPT, but for jobs whose waiting time is greater than a specific value, use FCFS rule.
	SEQ	Consider work-in-process value of the job, elapsed waiting time and the number of operation.
Local and Global Rules	SRPT	Shortest Remaining Processing Time (local)
	LRPT	Longest Remaining Processing Time (local)
	FOPR	Fewest Operation Remaining (global)
	MOPR	Most Operation Remaining (global)
Heuristic Rules	RANDO	Select in RANDOM order
	Look Ahead	Study the effect of scheduling a job on the other jobs that might arrive the queue before the scheduled job is completed
Static and Dynamic Rules	EDD	Earliest Due Date (static)
	StS	Static Slack: due date- arrival time
	DyS	Dynamic Slack: due date- the remaining expected flow time- the current date

2.4.2.3 *Shop Scheduling Models*

There is a wide range of shop scheduling models that can be used to address manufacturing systems. Shop scheduling models are often characterized by the machine configuration, processing constraints, production flow pattern, and the scheduling objective function. For example,

$$n/m/G/F_{max} \quad \text{or} \quad n/m/F/F_{max}$$

where,

F_{max} = Maximum flow time,

G and F refer to general job shop and flow shop, respectively.

The general scheduling models and in particular shop scheduling models are extensively classified and presented in Appendix B. This thesis is interested in the most common shop models; Job shop, flow shop, and flexible shop.

A) <u>Job Shop Model</u>

The job shop is the most general model of the shop floor. Job shops typically produce a large number of different products. For example, consider a shop floor where jobs are processed by machines, each job consisting of a certain number of operations. Each operation has to be performed by a dedicated machine and requires a predefined processing time. The operation sequence is prescribed for each job in a production recipe, imposing static constraints on scheduling. Thus, each job has its own machine order and no relation exists between the machine orders of any two jobs, Figure 2.10.

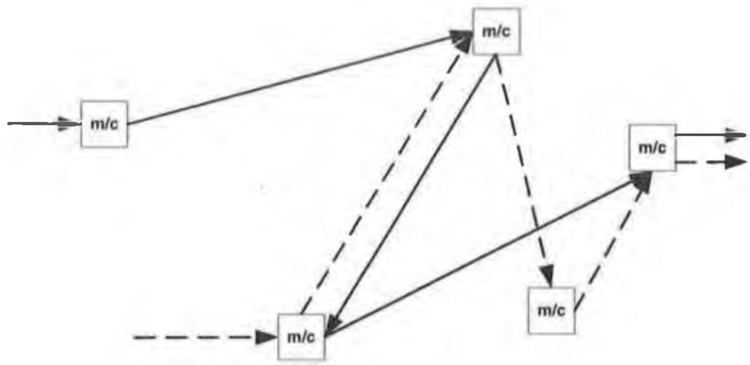


Figure 2.10: Typical job shop model

Theoretically, JSS problem is a set 'J' of 'n' jobs J_1, J_2, J_3, ... J_n has to be processed on a set 'M' of 'm' different machines M_1, M_2, M_3, ... M_m. Job 'J_j' consists of a sequence of 'm_j' operations O_{j1}, O_{j2}, O_{j3},....., O_{jmj}, which have to be scheduled in this order. Moreover, each operation can be processed only by one machine among the 'm' available ones. Operation 'O_{jk}' has a processing time 'P_{jk}'. The objective is to find an operating sequence for each machine such as to minimize a particular function of the job completion times, and in such a way that two operations are never processed on the same machine simultaneously.

An exhaustive survey on solving job shop scheduling problems using different techniques was presented by Arisha *et al.* [47], and Appendix B shows many types of job shop models with brief description.

B) Flow Shop Model

Flow shop model is a special case of job shop model where all the jobs have to take the same sequence of operations. The flow of work is still unidirectional, and can be represented as a pure flow shop (Figure 2.11) in which some of the operation times are zero. However, this does not allow a job to bypass a machine. In some flow shops, if a job does not need processing at a particular machine, it may bypass that machine (Figure 2.12) and go ahead of the jobs being processed or waiting for processing there.

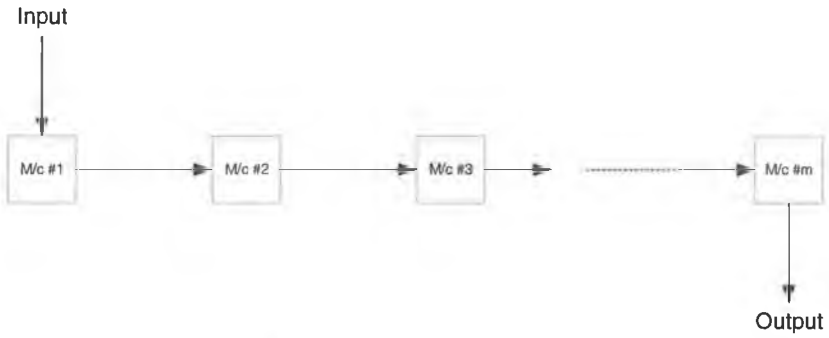


Figure 2.11: Work flow in Pure Flow Shop Scheduling Model

The focus of this problem is to sequence or order the '*n*' jobs through the '*m*' machine(s) so that some measure of production cost is minimized. Indeed, flow shop scheduling problem has been shown to be NP-complete for non-preemptive schedules. Chapter 4 discusses flow shop scheduling problem in more detail.

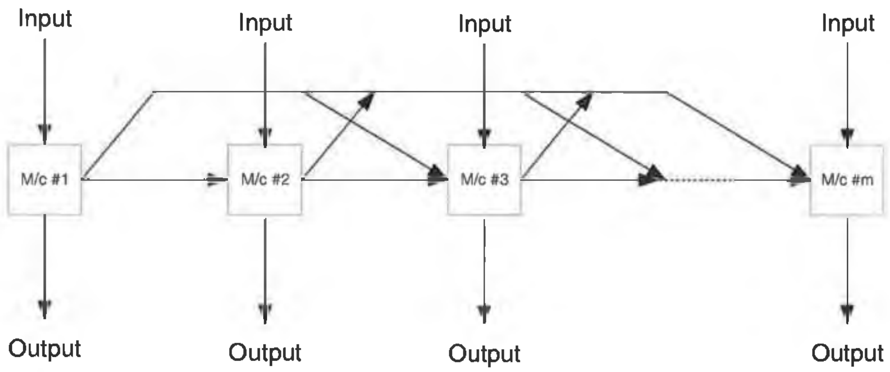


Figure 2.12: Work flow in General Flow Shop Scheduling Model

C) <u>Flexible Shop Models</u>

These models describe the flexible manufacturing cells, where the jobs path can be either flow shop or job shop. Flexible shop models are more difficult to schedule due to physical flexibility, and reentrant flow of products. The next section has an extensive literature survey on solving scheduling problems in flexible manufacturing environments.

2.4.3 FMS Scheduling Problems

Heavey [1] classified decisions within FMS into three levels; long term decisions, medium term decisions, and short term decisions. Figure 2.13 shows most of activities and problems encountered in FMS.

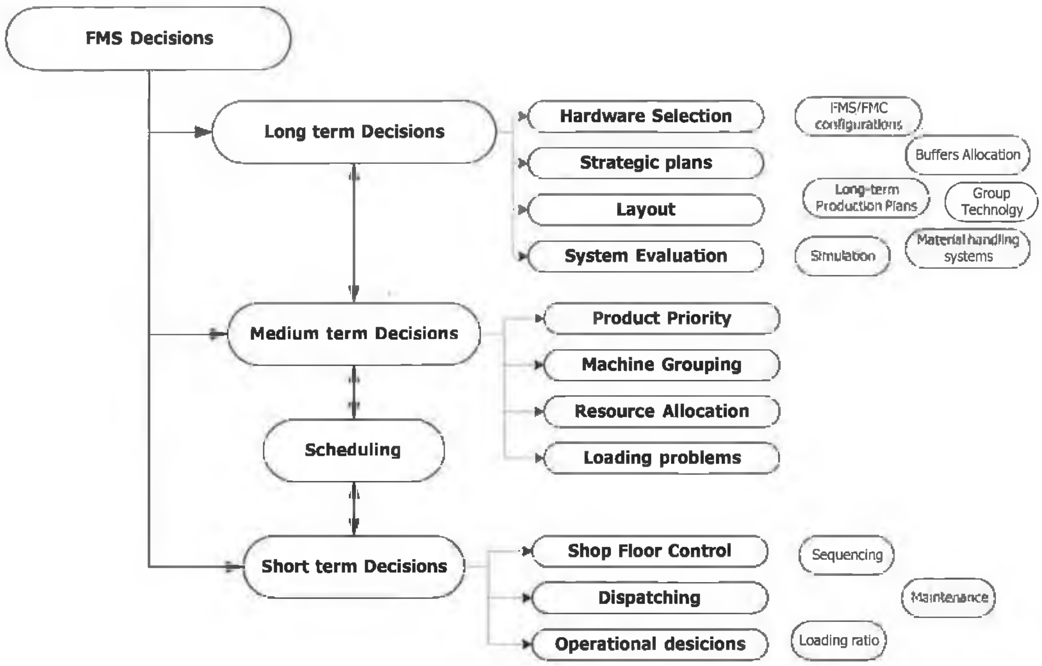


Figure 2.13: Flexible Manufacturing Systems Decisions

Comparison of different methods or models is always difficult. In order to analyze and compare scheduling strategies in FMS and develop more effective and efficient approaches, it is necessary to have a sound framework within which to define precisely the problem under investigation.

Liu and MacCarthy [25] presented a comprehensive classification of FMS scheduling problems based on attributes such as production system, capacity constraints, and performance measures. Rachamadugu *et al.* [26] reviewed the scheduling procedure of FMS, while Basnet *et al.* [87] focused on the control factor and scheduling approaches. Many more researchers discussed the FMS scheduling problem (e.g. Gupta *et al.* [88], Rodammer *et al.* [89]). Table 2.6 summarizes the FMS scheduling problems addressed in the literature into five groups.

Scheduling of FMC

FMCs have been developed over the last decades to help manufacturing industry move towards the goal of flexibility. An FMC comprises three principal elements: CNC tools, MHS, and a control system.

Table 2.6: Classification of FMS scheduling problems

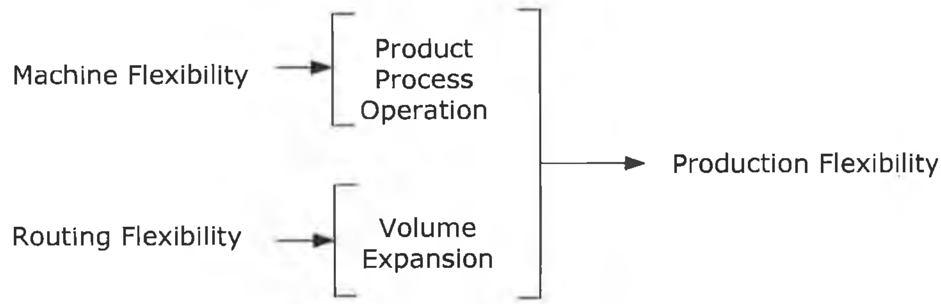
Factor	*Main types*	*Notes*
System Type	Flexible Manufacturing System (FMS)	FMS is a production system capable of producing a variety of part types, which consists of CNC or NC machine tools connected by an automated material handling system. The operation of the whole system is under computer control.
	Single Flexible Machine (SFM)	SFM is a computer controlled production unit that consists of single CNC or NC machine with tool changing capability, a material handling device and a part storage buffer.
	Flexible Manufacturing Cell (FMC)	FMC is a type of FMS consisting of a group of SFMs sharing one common material handling device.
	Multi-Machine Flexible Manufacturing System (MMFMS)	MMFMS is a type of FMS which consists of a number of SFMs connected by an automated MHS which includes two or more material handling devices or is otherwise capable of visiting and serving two or more machines at a time.
	Multi-Cell Flexible Manufacturing System (MCFMS)	MCFMS is a type of FMS that consists of a number of FMCs, and possibly a number of SFMs if necessary, all connected by an AMHS.
Capacity Constraints	Zero Capacity	It means that there is no buffer at that place in the system
	Limited Capacity	It indicates that the buffer can only accommodate a restricted number of parts at any one time.
	Infinite Capacity	When the buffer is large enough such that no delay will be caused by the waiting of a part for a place in the buffer.
Job Characteristics	One operation for each job	This factor describes the level of job complexity and the level of job routing. Complexity may be defined with the number of operations for the job and the capability of machines to perform more than one operation.
	Two or more operations for some or all jobs	
	One machine for each operation	
	Two or more machines for some or all operations	
Production Management Environment	Due Date requests	This represents the policies of the higher production management functions, which affect the scheduling activity. It is an important issue and needs careful consideration. The categories listed are not all the possible issues.
	Periodic orders	
	Continuous orders	
	Part per type	
	Lot per type	
	Ratio request	
	Batch Size request	
Scheduling Criteria	Make-Span	This factor can vary based on the planning objectives of the system. There are hundreds of objectives listed in publications such as, Baker [59] , Shannon [114], Iskander [41], and Banks [111]. The selection of the scheduling criteria is very critical issue for the system analysts to set with the planning and production stuff.
	Product/Batch Cycle Time	
	Machine Utilization	
	Maximum Tardiness	
	Machine Idle Times	
	Productivity measures	

Although there has been a vast body of work on production scheduling in the technical literature and industrial practice, the problem of assessing the quality of a given production schedule seems to be extremely expensive to implement in real-world manufacturing systems. The need of systematic framework for evaluating the performance of schedules generated is a crucial issue in research on scheduling.

FMC scheduling in semiconductor manufacturing is faced by profusion in product variety, decreasing lead times to delivery, exacting standards of quality, and competitive costs. Simultaneously, the need for quick and efficient approaches to deal with such multifaceted problems has increased. In chapter five, a proposed model to study the scheduling of FMCs has been used.

Relevant Literature

FMS comprises two main components that provide flexibility as shown in the diagram below.



Stecke and Solberg [90] carried out a study for a dedicated type of FMS examining five loading strategies and 16 dispatching rules. The study drew several significant conclusions about how the system should be controlled and indicated that the choice of applicable loading and dispatching strategies depends on many variables particular to the system. An FMS with multi-tool automated machines to produce a given product mix was studied by Arbib *et al.* [91]. They concluded that an optimal machine workload can be found by routing parts on a limited number of paths. They found the best routing by minimizing the global amount of part transfers among all machines.

For an FMS, the choice of an appropriate scheduling strategy is an important operational issue. However, only a few successful implementations of scheduling techniques were found in practice [92]. The benefit of flexible routing to address loading and parts dispatching problems have been studied [93]. Alternative operations can also significantly impact the performance of FMSs. As suggested by Ro and Kim [94], alternative operations could be implemented dynamically or be planned to offload bottlenecks, with the objective of improving machine utilization as well as part flow time by using a linear programming model before implementation. Chen and Chung [95] identified the potential benefits of multiple loading and alternative routing when the routing decision was planned or made before an order was released to the shop floor. Sabuncuoglu [86] examines the effect of scheduling rules on the performance of FMSs. He tested several machines and AGV scheduling rules against the mean flow time criterion.

An FMS ideally combines high levels of flexibility with high productivity and low levels of work-in-process inventory. It may also allow unsupervised production. In order to achieve these benefits, effective and efficient scheduling strategies are required. Many researchers have been studying routing policies and dispatching rules on different performance measures [96]. However, few studies cover machine selection and part dispatching simultaneously with various manufacturing parameters. A comprehensive survey on previous research on FMS scheduling problems have been presented by Chan [97]. The paper includes a summary of the publications on scheduling problems in last two decades. The majority of research dealt with part dispatching (70%) while few of these considered routing and operation selection problems.

2.5 Solution Techniques

A vast body of literature has focused on the most general forms of shop scheduling which are job shop scheduling and flow shop scheduling. This section provides an extensive survey on the solving techniques over the last decades.

The scheduling problem was first studied in the mid-fifties in the form of a paper presented by Johnson [48]. In the following years, several studies discussed the solution to SS in its standard form (e.g. Jackson [49], Conway *et al.* [29], Lawrence *et al.* [50], and Brucher [39]). An analysis of scheduling problem complexity is shown in Figure 2.14.

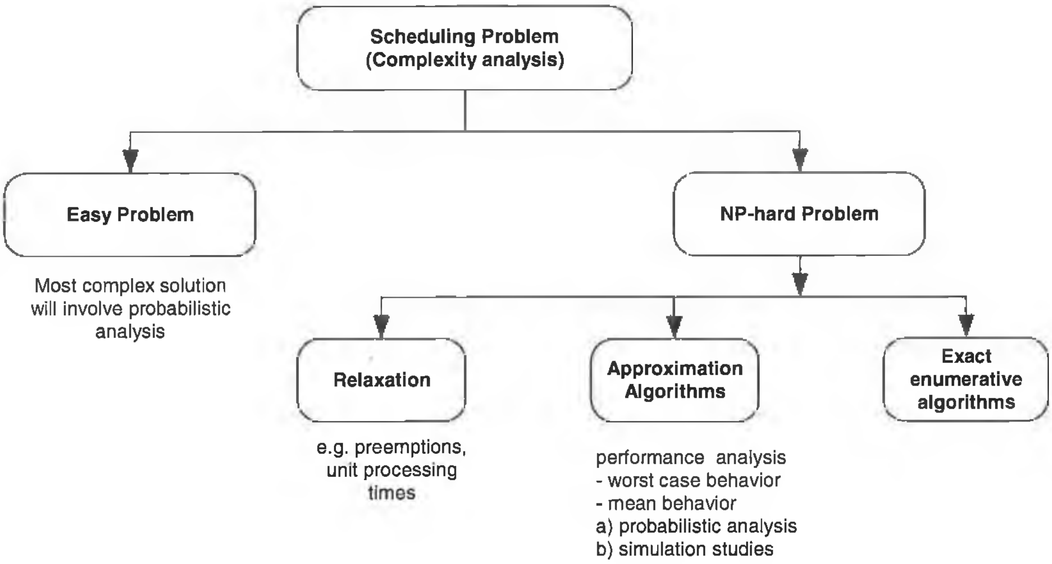


Figure 2.14: An Analysis of a scheduling problem – schematic view

The general SS problem is NP-hard in the strong sense (Lenstra *et al.* [31], Gonzalez *et al.* [51], Garey *et al.* [52], Rinnoy [33]) and is probably one of the most computationally intractable combinatorial problems existing in manufacturing systems. A practical proof of this intractability comes from the fact that a small example with 10 jobs and 10 machines posed by Thompson *et al.* [53] was an open problem for over 15 years. It was solved by Pinson *et al.* [54] as the culmination of a considerable amount of research.

Feasible schedules are obtained by permuting the processing order of operations on the machines (operations sequence) without violating the technological constraints, resulting in a maximum of $(n!)^m$ different solutions for a given problem [55]. The explosive exponential growth in the number of alternative schedules with the size of problem is central to the difficulty of identifying one of these as the optimum schedule.

The difficulty is twofold:

- First, there is the problem of deciding what characteristics should define for the best schedule.

- Second, how can such a schedule be efficiently determined?

Techniques have been developed by researchers to deal with the scheduling problem. These techniques can be grouped as traditional and advanced techniques, Figure 2.15.

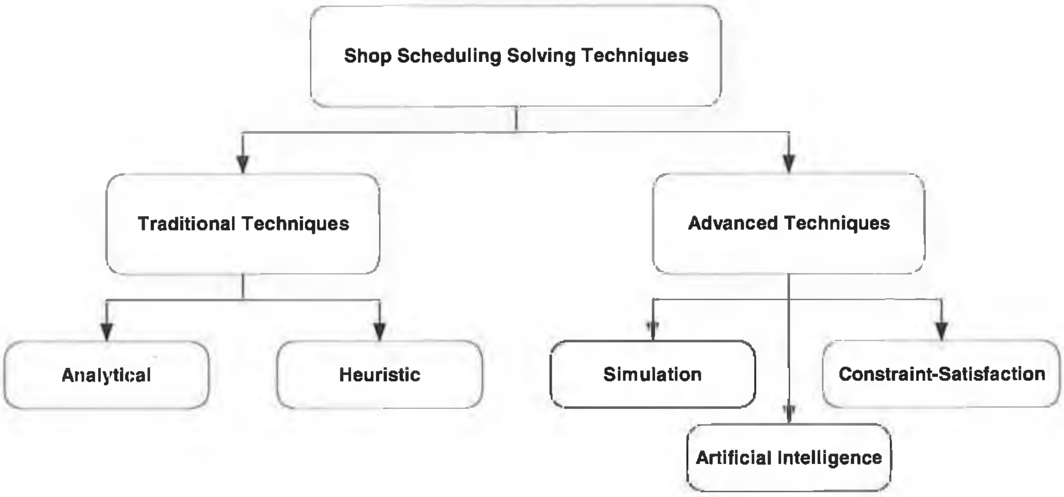


Figure 2.15: Shop scheduling solving techniques

Often these solution techniques have been combined to provide more comprehensive solutions, Figure 2.16.

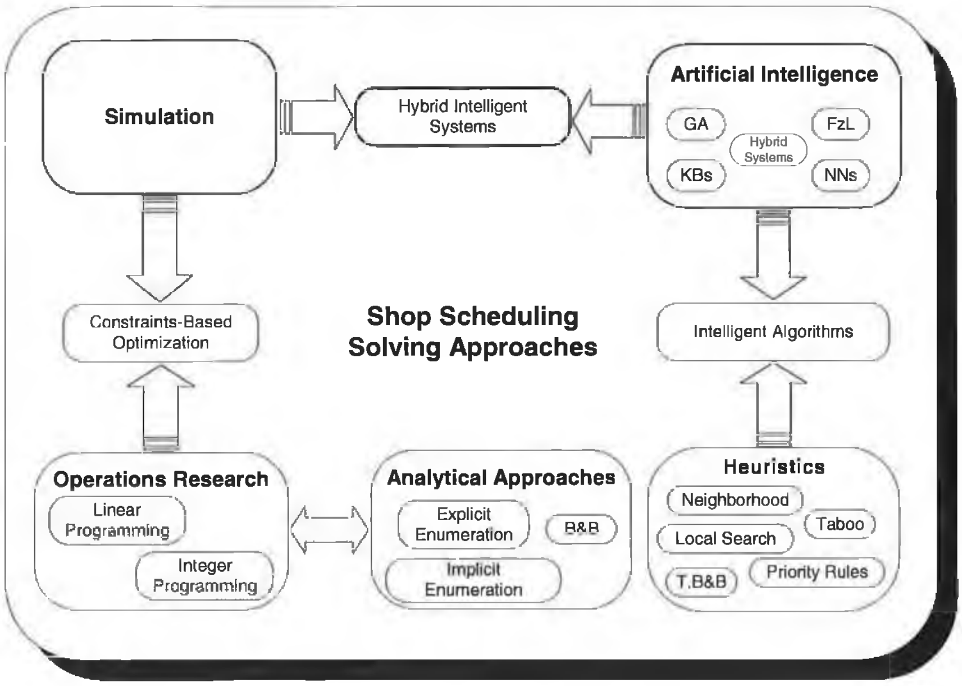


Figure 2.16: Relationship between shop scheduling solving approaches

2.5.1 Traditional Techniques

Traditional techniques can be classified under two main categories, i.e. Analytical Techniques and Heuristic Techniques, Figure 2.17.

The general approach of the analytical methods is to consider the problem in its total system form of scheduling '*n*' jobs on '*m*' machines. The relative lack of success of this approach in providing a general optimization method of wide applicability has led to a switch in the focus of attention from the total system to a more simple decomposed subsystem view of the problem, in which the job shop is considered to be a series of interrelated single machine scheduling problems [2].

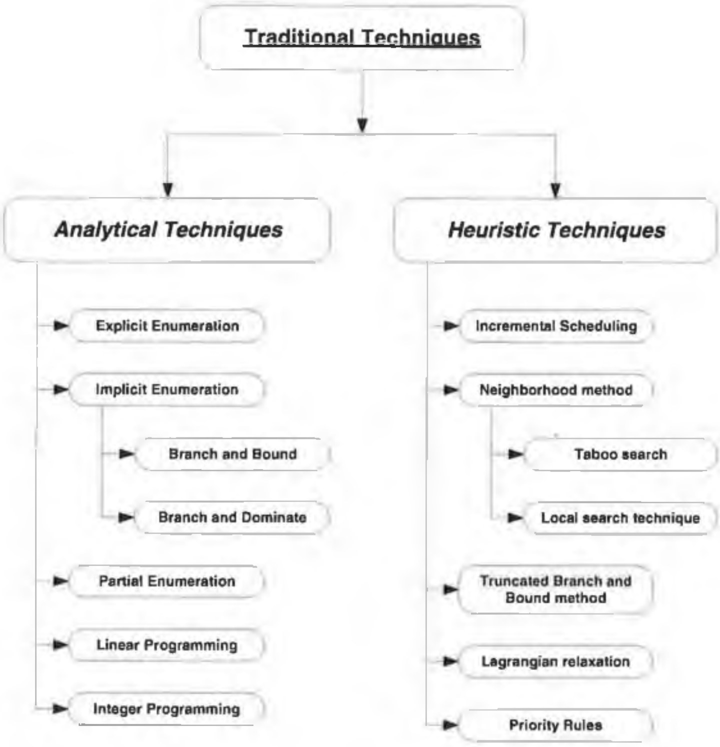


Figure 2.17: Traditional Techniques

Fisher [56] obtained a more efficient enumeration method for JSS problems. More recently, a technique to obtain near-optimal solution for parallel identical machines has been used by Hoitomt *et al.*[55]. A shifting bottleneck heuristic as one successful approach for decomposing the job shop into sub-problems is presented by Pinedo [58].

Table 2.7: Analytical Techniques

Main Features	***Limitations***
1. Explicit Enumeration - In this method, they generate a complete enumeration tree. The leaves of the tree represent all feasible solutions. The path from the root to a leaf of minimal make-span represents an optimal solution. - [59] and [53] offered an algorithm which created an active schedule with respect to disjunctive arcs. - A computer program algorithm, which deals with optimal job sequence (250x250), it was economic for small and medium-sized problems only [60].	- The remaining difficulty is the size of search tree generated. Since we have a maximum of $(n!)^m$ solutions to consider. - The limitation of algorithm [59], [53] is that it presents procedure relations that cannot be determined before a schedule is constructed. Also, it is not adequate to capture sequence dependent set-up and tear down time in every case (White *et. al* [61]).
2. Implicit Enumeration - The strategy of implicit enumeration attempts to minimize an objective function without considering every possible solution. Implicit enumeration schemes examine increasingly smaller subsets of feasible solutions until these subsets definitely do not contain improved solutions.	- All implicit enumeration approaches for the determination of an optimal schedule appear to be susceptible to the combinatorial nature of these problems, when they are tested with multiple-resources (more than 50 activities) [59].
2.1. Branch and Bound - Branch and Bound Algorithms cut branches from the enumeration tree and therefore reduce the number of generated nodes substantially. An optimal solution can be found by systematically examining the subsets of a feasible solution. - Several different algorithms exist for SS [29] [59] and have been applied to flow scheduling [62]. Survey on Branch and Bound methods in [39].	- The limitation of this algorithm is that the make-span is the only criterion, which can be evaluated. - The efficiency of the technique depends very much on the efficiency of the lower bound. The more efficient the bound, the smaller the amount of enumeration of the solution tree that needs to be carried out. - It is impractical to enumerate even this reduced set of alternatives because it is too large.
2.2. Branch and Dominate - Similar to Branch and bound but differs in the pruning approach. If there is a set of conditions at a node which mean that the schedules will be inferior to the best schedule at some other node, then the first node may be eliminated from further consideration. In this way the second node dominates the first.	- Using dominance conditions may shorten the search sufficiently such that a reduction in overall computational requirements is obtained [59] [33]. It is still impractical to enumerate this reduced set.
3. Partial Enumeration - The optimal schedule has been shown to always be in a subset of feasible schedules, termed 'active'. This identification of such active feasible schedules has been used in [63]. Recently, Shifting Bottleneck algorithm is considered as a good step in partial enumeration by [58].	- This method to define active and semi-active schedule's helps to reduce a computational work somewhat. However, there is still a need to generate a high number of schedules to get the optimal one. The problem complexity increases with more machines and jobs.
4. Linear Programming - The particular attraction, from the model building point of view, of linear programming is that highly efficient program codes are available which can deal with very large problems involving many variables and constraints. - It is fair to say that, in some cases the specific nature of the problem allows certain simplifying approximations that permit a solution by Linear Programming.	- Linear Programming can often be used as a practical technique, but only if the problem conforms entirely to the requirements of the approach. - The main shortcoming in that most of the real manufacturing planning and scheduling do not behave linearly in most cases, even after the simplifications. - Moreover, some or most of constraints in practice cannot be represented as linear. For example, the specification that either machine A or B may be used to process job j.
5. Integer Programming - To overcome some of the limitations of linear programming integer variables may be used. This complication the solution requiring the use of less efficient algorithms. Integer Programming formulations of the JSS problem have been reported in [64].	- The present integer programming codes available are over-stretched even by very small JSS problem formulations. Even allowing for possible developments and improvements in integer programming computer codes, it would appear that as a general method for solution of JSS problems it is a non-starter.

Table 2.8: Heuristic Techniques

Main Features	***Limitations***
1. Incremental Scheduling - Incremental schedule building starts with an empty timeline and a set of tasks to be scheduled. The basic idea behind incremental scheduling is to choose the next task to be scheduled and to place that task on the timeline so that no constraints will be violated. The placement algorithm may be very simple or very involved, attempting some degree of optimization. - This process repeats until either all tasks have been scheduled, or there are still tasks that remain to be scheduled but no times. In this latter case, the scheduler has effectively reached a dead-end.	- Some systems halt at this point of dead-end, presenting an incomplete solution to the user. Others attempt to free the scheduler from the dead-end condition by undoing some previously made decision. Incremental scheduling can degenerate into brute-force trial-and-error searches. Since this is a computationally intractable alternative, incremental scheduling systems tend to be either slow or poorly optimizing, or both.
2. Neighborhood method - Neighborhood search techniques begin with any feasible schedule, adjust this somewhat, check whether the adjustment has made any improvement. Continuing in this cycle of adjusting and testing until an improvement measure is achieved. Two related concepts, which are the basis of this method, are the neighborhood sequence and the neighborhood generating mechanisms for these sequences [30].	- The search procedure of this family of algorithms terminates with a sequence that is a local optimum. Unfortunately, there is in general no way to guarantee or even know if the terminal sequence is also a global optimum. However, few experiments indicated that, fundamental neighborhood search algorithm described above, is fairly reliable as a general-purpose heuristic procedure [59].
2.1. Taboo search - Taboo search approaches produce good results in reasonable runtime. Taillard [47] applied this global optimization technique to the SS and showed that it is typically more efficient than the shifting bottleneck procedure and simulated annealing implemented by Lenstra [31]. Taillard provides optimal solution for some identified problem with shorter computational time for more complex problems.	- Requires large memory, as subsets of the solution path are kept in memory. - Another crucial aspect is the maintenance of the taboo list using variable taboo list length and cycle detection mechanisms which prevent cycling around a number of neighboring solutions.
2.2. Local search technique - Simulated annealing [65] and taboo search techniques are the main local search techniques that have been tested on the SS problem. In both cases, the neighborhood structure is based on scheduling arrangement.	- In comparison with other heuristic methods both techniques yield quite consistently good solutions. - Simulated annealing is comparatively much more time consuming that taboo search on difficult instances.
3. Truncated Branch and Bound method - One of the most efficient approximate methods proposed so far is probably the shifting bottleneck procedure developed by Adams *et al* [57]. Its main idea is following: Starting with the initial SS problem, they optimally sequence one by one the machines, using Carlier (1982) algorithm for the one machine problem. The order in which the machines are sequenced depends on a bottleneck measure associated with them.	- This procedure is embedded in a second heuristic of an enumerative type, for which each node of the search tree corresponds to a subset of sequenced machines. In comparison to other algorithms, it is less efficient as each time a new machine is sequenced, they attempt to improve all previous scheduled machines in long re-optimization steps.
4. Lagrangian relaxation - Scheduling methodologies based on Lagrangian relaxation have proved to be computationally efficient and have provided near optimal solutions to identical parallel machine scheduling problems. It has been applied to schedule job shops, which include multiple machine types, generic precedence constraints and simple routing considerations [56].	- It can be applied to some cases in machine scheduling and under certain conditions. - The results are not guaranteed in complex JSS problems. - It provides near optimal in case of identical machine scheduling.

5. Dispatching Rules
- Dispatching Rules indicate how to assign a specific job to a specific machine at a given time, when a machine becomes available for process.. A lot of studies were done over these rules,[29][30][54][59]. Pinedo & Bhaskaran [43] presents classification of basic dispatching rules.
- Panwalker *et al.* [41] presented over 100 priority rules. Dispatching Rules can be classified into groupes:
  - Simple Priority Rules.
  - Combination of simple Priority Rules.
  - Weighted Priority Index.
  - Heuristic Scheduling Rules.
  - Other Rules.
	- Researchers have analyzed sequencing decisions jointly with other dynamic decisions (see Ref. [45]).
	- Unfortunately, none of the rules seems to outperform any other for practical problem setting. Recently simulated annealing was also applied to deal with JSS problems as a remedy.

Attempts to bridge the gap between heuristic approaches and analytical approaches have also been undertaken (e.g. Hoitomt *et al.* [55], Fisher [56], and Adams [57]). In Adams, the solution is provided by 'local optimization'. However, schedule evaluation could only be achieved through "selective enumeration". Recently, the Lagrangian relaxation technique has been used by A concise survey on main analytical and heuristic techniques that have been used to deal with JSS problem is provided in Tables 2.7 and 2.8.

2.5.2 Theory Practice Gap

Unfortunately, no simple scheduling algorithm exists for the general 'n' jobs, 'm' machines case of JSS. There is a gap between scheduling theory – as represented by analytical methods – and practice, Figure 2.18. This stems from the inability of theory, as so far developed, to cope adequately with the complexities of many of the real-world JSS problems.

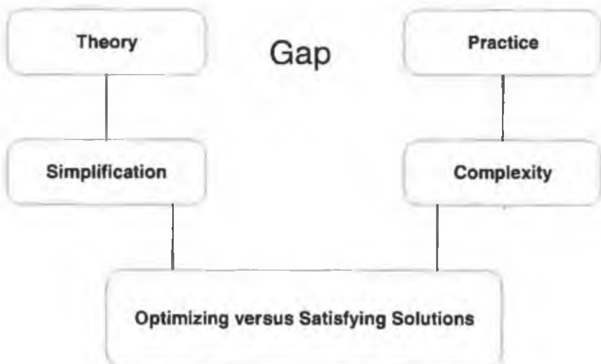


Figure 2.18:The Theory-Practice Gap [2]

Many researchers in the field, faced with these difficulties in solving scheduling problems, make simplifying assumptions and approximations to reduce the problem to a form that they can hope to solve. In case of SS the fundamental difficulties of real practical problems have led to simplifications on a scale, which in some cases has reduced the problem to a shadow of reality [2].

These have resulted in:

1. Emphasis on small-scale problems

Although most of research is concerned with the general n-job, m-machine scheduling problem there has been a great deal of concentration of effort on the small-scale problem involving at most four machines.

2. Simplified Problem constraints

These constraints are covered in detail by French [30], the main ones are:

- All jobs and their processing times are known prior to scheduling being carried out, effectively transforming the dynamic problem to a static one.
- Machines are assumed to be able to operate on only one job at a time.
- Job splitting and job lapping are only permitted in very exceptional cases.

3. Simple objective functions

Most of these are single parameter objective functions, which are to be optimized. The most commonly used objective functions are: Mean flow time, Total lateness, and Number of late jobs.

It is true to say that some insight into the solution of the practical SS problem has been derived from this research activity on simplified problems. In practice, the complexity of production planning and control in general and scheduling in particular varies from one situation to another. The degree of complexity is governed by such factors as average number of operations per product, product variety and scale and type of production.

4. Optimization & satisfying solutions

Analytical methods are generally concerned with optimizing solutions, where the optimization is carried out with respect to some particular success criteria. It is well known that the measures of success are multi-dimensional. For example, make-span time may be important but so is work in progress, machine utilization, labor utilization, delivery performance ... etc.

The relative importance of each will vary from company to company, and indeed may change over time within a company. Probably, the most rational objective function would be one comprising a weighted function of the various performance criteria deemed relevant, with management determining subjectively the relative weight of each. In practical scheduling work, they are more concerned with establishing feasible schedules, which will provide satisfactory performance against measures of success such as flow time, delivery dates, utilization, ..etc.

It is worth mentioning that the computational time to find the solution is uneconomic in most uses of traditional techniques. Heuristic methods of scheduling, by definition, give rise to satisfying solutions in shorter computational times and the advanced techniques build on this to provide a more comprehensive evaluation in the research for such solutions.

2.5.3 Advanced Techniques

2.5.3.1 Simulation

Simulation has proven to be an excellent strategic tool for high level planning; however, it can also be used as a day-to-day tactical tool on the shop floor. While simulation can be applied to many aspects of manufacturing systems, two areas stand out in particular:

1. In Job shop, the simulation of dispatching rules and the assessment of the effect of different rules on the shop's ability to meet delivery dates and utilize the machines.
2. In flow lines to try to minimize the loss of output.

However, it has also been applied to more advanced systems in manufacturing such as FMS, Automation,...etc. The first application of simulation was studies of different priority rules carried out by Elmaghraby & Cole [66], applied their control of the production at Western Electric.

Other investigations such as [49][67][68][69] have experimented with computer simulation models of hypothetical shops in which assumptions are made about the mechanism for generating job arrivals and processing times, while Jones [70] establishes an economic evaluation of job shop dispatching rules. The priority dispatching rules in job shops with assembly operations and random delays has been studied [71], followed by more comprehensive study of Sculli *et al.* [72] in a fabrication/assembly shop.

Ezat & El Baradie [73] showed how to use computer aided-simulation as a tool for the optimization of pure flow shop scheduling under different priority rules, followed by further study on the effect of various priority rules on minimizing multiple criteria. The objective of all these simulation experiments has been to evaluate and determine efficient and effective scheduling rules that may be generally applied in practice.

Arisha *et al.* [171] developed a simulation model for general flow shop scheduling. This study aimed to:

1) To provide a simulation model able to find the optimum / near optimum sequence for general flow shop scheduling problem with make-span minimization as main criteria;
2) To compare computational time to obtain feasible solution in two different solving approaches.
3) To examine different dispatching rules on minimizing multiple criteria.

This Simulation model can use for deterministic and stochastic flow shop scheduling. It reads and manipulates data for 500 jobs on 500 machines. The model presents heuristic technique (Dispatching rules) with different factorial experiments in a comparative study on the performance of different dispatching rules, such as FCFS, SPT, LPT, SRPT and LRPT with respect to the objectives of minimizing make-span, mean flow time, waiting time of jobs, and idle time of machines. The proposed model is evaluated and found to be relatively more effective in finding optimal/ near optimal solutions in many cases. The influence of the problem size in computational time for this model and the results obtained are discussed in Appendix B.

The use of simulation software has thus become widely accepted as a tool for the improvement and enhancement of the performance of a manufacturing system in general. Simulation is also accepted as the tool for the evaluation of the manufacturing system in operations using "What-If" scenarios prior to doing any harm in real life. Current tools make it relatively simple to build a simulation model for planning and scheduling. Using this model, through the definition and application of the rules used to assign work to the available resources, the scheduler can be sure that all of the combinations and exceptions are considered and the production objectives satisfied. More recently the tracking and reporting of this process has been integrated within the software, and hence simulation-based scheduling has become the start point in solving the scheduling problems. Improvements in simulation software can help to find efficient way to shorten the time needed to get the optimal scheduling. More about simulation applications in scheduling is described in chapter three.

2.5.3.2 Artificial Intelligence

The need for rapid solutions prompted researchers to use AI techniques such as Knowledge-Based Systems (KBS), Expert Systems (ES), Neural Networks (NNs), Case-Based Reasoning (CBR), Genetic Algorithms (GA), Fuzzy Logic and any combination of these techniques [5]. Figure 2.19 illustrates some applications of AI in scheduling to deal with SS problem.

2.5.3.3 Constraint Satisfaction Methods

One of the most promising general approaches for solving combinatorial problems is repair heuristics. Scheduling appears to be an excellent application area for repair-based methods. Supporting evidence comes from previous work on other real-world scheduling applications by Zweben [74], Biefeld and Cooper [75], and Lee *et al.* [135]. Each of these projects uses iterative improvement methods that can be characterized as repair-based. Repair-Based approach is that it is extraordinarily well suited to rescheduling. Rescheduling typically takes less time than the initial schedule generation, as it requires fewer repairs. It has been pointed out, there are real-world scheduling problems where humans find repair-based methods very natural [84].

Repair-based idea was extended in a natural manner to solve constraint satisfaction problems (CSPs) [76]. A CSP consists of a set of variables and a set of constraints. The constraints indicate the allowable combinations of values that can be assigned to the variables. A solution is an assignment specifying a value for each variable, such that all the constraints are satisfied. A repair-based constraint-satisfaction method takes the variables and the constraints and begins by generating an initial assignment for the variables. The initial assignment is then repeatedly “repaired” until a solution is found. Heuristics such as Min-Conflicts attempt to minimize the number of variables that will need to be repaired in order to reduce the search space.

The *n-queens* problem is still the standard benchmark for testing CSP algorithms [77][78]. Whereas the n-queens problem is only of theoretical interest, scheduling algorithms have many practical applications. A scheduling problem involves placing set of tasks on a time line, subject to temporal constraints, resource constraints, preferences, etc. CSP has considered scheduling as a constraint optimization problem [80]. The fact that the min-conflicts approach well on *n-queens*, a well studied "standard" constraint-satisfaction problem, suggests that AI and repair-based methods might be more useful than previously thought. Minton *et al.* [79] developed the initial scheduling system 'SPIKE' using the min-conflict method to solve scheduling problems. There are still many possible extensions to the CSP methods which would improve the performance of the solution technique. Minton *et al.* [76] suggested more heuristics are still needed to combine with the existed heuristic to provide robust solutions. For example, min-conflicts may combine with hill climbing heuristic search or backtracking in order to minimize the search space and the solution time. In addition, more sophisticated techniques such as best-first search is investigated [78] to be used with min-conflict to provide quicker solutions.

Srivastave *et al.* [81] have developed a 'REALPLAN' in which resource allocation is de-coupled from planning and is handled in a separate scheduling phase. This research can be viewed as an important step towards merging planning with real-world problem solving where plan failure during execution can be resolved by undertaking only necessary resource re-allocation and not complete re-planning. Jönsson *et al.* [82] has integrated ANN methods for general CSP to solve problems which include boolean variables. In contrast to conventional ANN methods, it employs a particular type of non-polynomial cost function, based on the information balance between variables and constraints in a mean-field setting. The performance is comparable to that of dedicated heuristics, and clearly superior to that of conventional mean-field annealing. Shang *et al.* [83]developed a new heuristic taking into account the complexity of continuous constrained problems. In their research, they studied complexity phase transition phenomena of continuous CSPs, then analyzed three continuous constraint satisfaction formulations based on (discrete) 3-SAT problems, which have a strong relation between structure and search cost. They proposed a generic benchmarking model for comparing continuous CSPs and algorithms, and presented two example problems based on sine functions. Regarding local versus global search techniques for constraint solving, the obtained results show that local search methods are more efficient for weakly constrained problems, whereas global search methods work better on highly constrained problems.

However, there is no guarantee that a solution will be found quickly or even that a satisfactory solution will be found in some complex cases using CSP methods.

2.6 Conclusions

Scheduling is a ubiquitous task that involves time-sequenced allocation of the available resources and jobs loading. With manufacturing responding more and more to the preferences of customers, the scheduling process is getting more complex. The scheduling complexity is a function of manufacturing systems complexity. A range of classifications of manufacturing systems exist which can help in defining scheduling problems. Solultion techniques have been classified into two main groups; traditional and advanced techniques. These different techniques have characteristics which dictate what problems they can be most effectively applied to. The low acceptance of traditional techniques and the consequent overreliance on advanced techniques is due to a number of reasons (e.g. time needed to find the solution, and the quality of the solutions obtained). This thesis utilized many solution techniques (e.g. analytical, heuristic, simulation, AI, and hybrid techniques) for scheduling problems (e.g. pure/general flow shop, flexible job shop in semiconductor manufacturing environment). There are number of papers that are relvant to the development of scheduling models in flexible manufacturing systems in semiconductor fabrication. These and other aspects used in the development of the proposed model are described in detail in the next chapter.

HybridSystems:

- Szelke and Markus (1997) [81]combined machine learning techniques and CBR to solve shop floor scheduling problems.
- Kim (1998) [82] used an integrated approach of inductive learning and competitive NNs for developing multi-objective FMS schedulers. The FMS scheduler can assist to make a decision in real time, while satisfying multiple objectives desired by the operator.
- Lee (1998) [85]combined GAs and machine learning to develop a JSS system.
- Soon and Souza (1997) [90]have explored the possibility of using a hybrid approach employing neural networks and simulation technology to solve the detailed scheduling problem (a known class of NP-complete problems) in a manufacturing cell.
- Hybrid GA is proposed for job shop scheduling where selected dispatching rules are integrated into the process of genetic evolution [96]. In addition, neighborhood search is adopted as an auxiliary procedure.

Fuzzy Logic :

- Yu (1999) [88] proposed approach to FMS with Multi-criteria based on Fuzzy inference.
- The proposed system performed very robustly with respect to shop workload for all used performance measures..
- Fuzzy logic tool intended to rank flow shop dispatching was presented by Petroni et al. [95].

Neural Networks (NNs):

- Neural Networks have been successfully applied to the solution of constrained optimization problems.
- The scheduling can be solved using NNs to reduce the computational complexity (Grabot 1998) [84]. NNs have been tested in order to emulate the expertise involved in the definition of such compromise (Zhang et al. 1995) [89].

Intelligent Scheduling Tools

Knowledge-Based Systems (KBS), Expert Systems (ES):

- A reformulate approach to scheduling is proposed to find a satisfactory solution. Although the algorithmic approach concentrates heavily on the solution methodology and treats the problem formulation as given. The reformulate approach acts to enrich the model (problem formulation) until a satisfactory solution [76].
- Orcich and Frost (1984) [91] has described the first working system ISA (Intelligent Scheduling System).
- Then ISIS (Intelligent Scheduling and Information System) is described its first level by Fox and Smith (1984) [92] as a knowledge based decision support system designed to provide intelligent support in the domain of JSS.
- scheduling is an optimization problem. Thus, the design of ISIS focus on: 1) Constructing a knowledge representation that captures the requisite knowledge of the job shop environment and their constraints. 2) Developing search architecture
- One of the most important advantages in the ISIS family that they are designed for both predictive and reactive scheduling (Fox&Smith 1984) [92].
- ISIS-1 uses pure constraint guided, but was not very successful in solving practical scheduling problems.
- ISIS-2 uses more sophisticated search techniques. Search divided into the four phases: Job selection, Time analysis, Resource analysis and Resource assignment.
- ISIS-3 tries to schedule each job using more information from the shop floor especially about bottleneck resources.
- The architecture of ISIS is inflexible as far as modifications of given schedules are concerned, a new scheduling system called OPIS developed (Blazewicz et al. 1994) [93].
- Where as the ISIS systems are primarily job-based and OPIS switches between job-based and resource-based considerations, CORTES takes a task-oriented point of view, which provides more flexibility of greater search effort.
- SONIA is the knowledge based scheduling system that relies on constraint satisfaction and It has a very flexible architecture, generates predictive and reactive schedules and integrates both solution approaches.
- CAMPS, EMPRESS, and EMPRESSII were an important class of scheduling systems that assist the human scheduler [94].

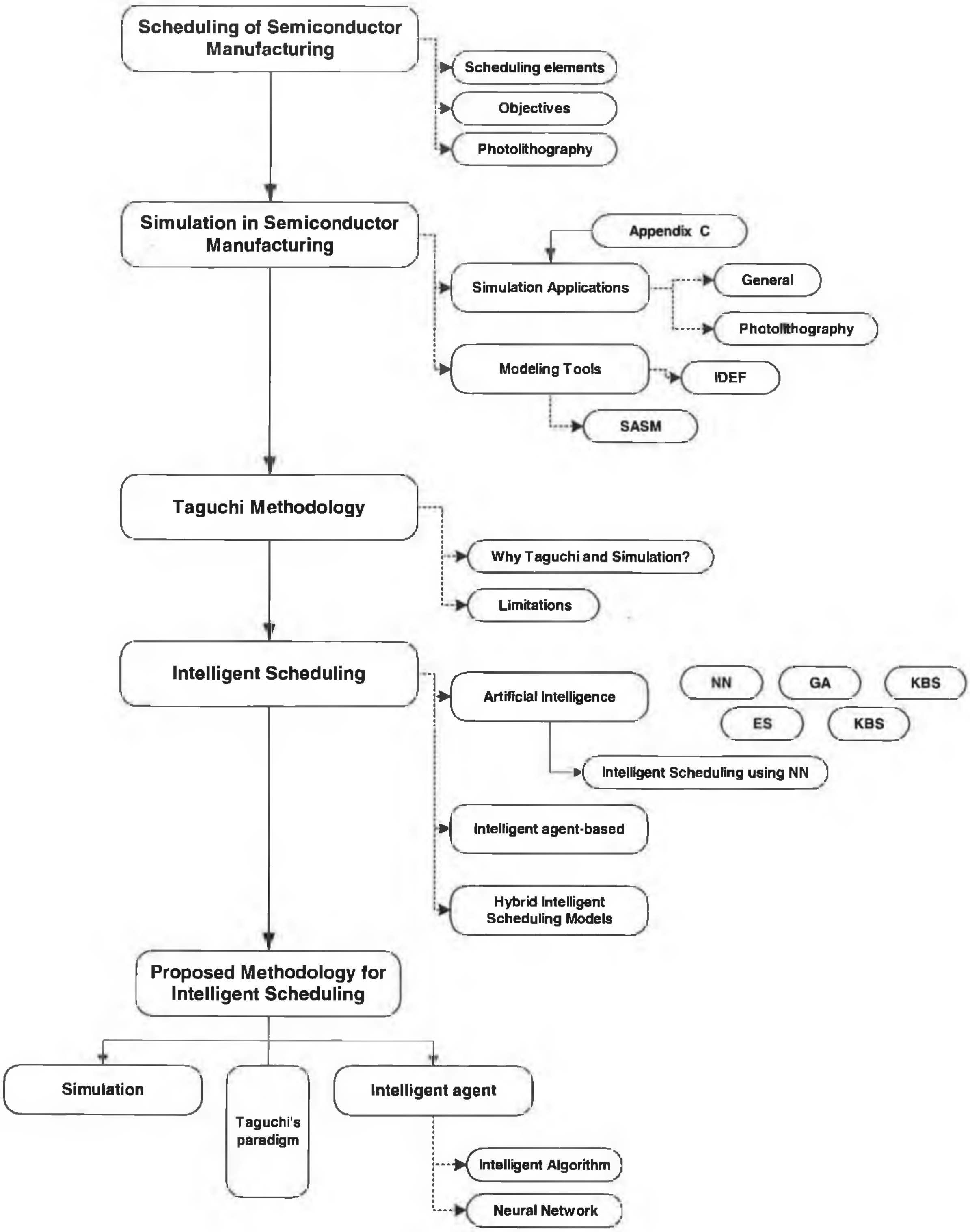
Genetic Algorithm (GA):

- Candido et al (1998) [83] used a GA to solve job scheduling problems with many constraints such as jobs with several subassembly levels, alternative processing plans for parts ..etc.
- Hybrid GA that incorporates a local hill climbing procedure is applied to the set of local optimum schedules.
- Webster (1998) [87] used a GA for scheduling jobs about unrestricted common due date on a single machine.
- Min and Cheng (1999) [86] have used GAs for the identical parallel machine-scheduling problem for minimizing the make span.

Figure 2.19: Intelligent scheduling tools

Chapter 3

Literature Review: Scheduling in Semiconductor Manufacturing



Chapter 3

Literature Survey: Scheduling of Semiconductor Manufacturing

3.1 Introduction

Semiconductor manufacturing is among the most complicated and capital-intensive manufacturing processes in the world. In this chapter, an overview of scheduling problems of wafer fabrication, which is the complicated portion of planning, is discussed. Semiconductor manufacturing is a highly competitive business [121]. In the past, competition has been primarily in the product design arena, but in the last years, the cost to manufacture has become an important competitive factor, especially when the cost of the wafer fabrication facility is expected to exceed \$ 3 billion (US). The magnitude of investment required makes it imperative to use equipment in an optimum manner. In addition, reducing the time to manufacture a product is becoming increasingly essential.

Scheduling addresses these needs and also extends to affect the decisions about the impact of changes in the products, enlarges in type and number of equipment necessary, and capacity planning. This chapter discusses the scheduling problems in semiconductor manufacturing. It also presents solution techniques for these problems. The need for a highly productive precise alternative to batch manufacturing, in particular, has always been the major driving force behind the flexible manufacturing system (FMS) development [1]. Semiconductor manufacturing needs efficient automatic tools to turn thousands of operations into a final complex product [86]. FMS represents more than 90% of semiconductor equipment in use. Therefore, scheduling problems in flexible manufacturing systems and reviews related research to solve associated scheduling problems are described.

Scheduling tools, such as using simulation and artificial intelligence to solve scheduling problems along with other modeling and optimization techniques are also discussed. Finally, a new methodology for scheduling in photolithography manufacturing area is proposed.

3.2 Scheduling of Semiconductor Manufacturing

A detailed description of major semiconductor processes can be found in Appendix A. The main aspects in wafer fabrication scheduling problems are classified into six main groups. Examples of the elements of each group is shown in Figure 3.1. Scheduling has to take these elements into account.

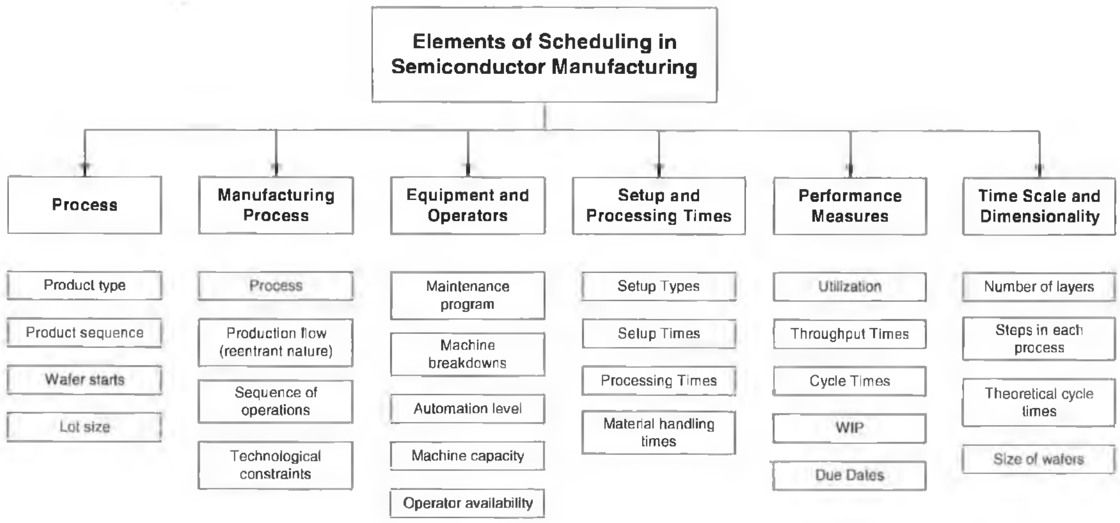


Figure 3.1: Scheduling elements in Semiconductor Manufacturing

Scheduling is concerned with the problems ranging in scope from the next hour to the next several months. The issue addressed by scheduling at the shop floor level include decisions on the following:

1. How much of each product should be produced on a given period of time.
2. What priorities should be assigned to the different lots competing for the same resource.
3. When to perform preventive maintenance (PM).
4. Decisions to reduce cycle time per resource and per lot.
5. How to reroute product flows when tools are down.
6. Which tool to used for a given product and layer.

Photolithography is perhaps the most complex process in wafer fabrication. Most of the literature that addresses the planning and scheduling problems in wafer fabrication in last years has not dealt with the photolithography area with a high level of detail. This is no doubt that the model complexity increases due to the reentrant nature of the process flow and the high level of variability in the manufacturing areas.

Research on planning and scheduling that employ simulation often model the entire factory and treat the photolithography as a black box [120]. As a consequence, the quality of results obtained do not show the impact of proposed schedules on this critical manufacturing area.

Scheduling in the photolithography area is a difficult task. This comes from the fact that there are so many decision points along the process flow. Developing a model incorporating all the process details, manufacturing procedure details, and associated operation details becomes extremely complex for the photolithography process. Well-thought out simulation models along with a robust systems approach can be used effectively to characterize this manufacturing area. Hence, an optimization process can provide the manufacturing team with optimum/near optimum solutions. Chapter four and five have more detail about the photolithography scheduling models.

3.3 Solution Tools for Semiconductor Scheduling

Due to the complexity of wafer fabrication manufacturing systems, simple analytical approaches to solve scheduling problems connot provide satisfying solutions. Many deterministic scheduling algorithms can be found in Uzsoy *et al.* [117]. However, most of these algorithms/techniques run into trouble with the uncertainty of many elements in semiconductor manufacturing such as job arrival, equipment breakdowns. A key factor in linking production planning and shop floor decisions is the development of accurate methods of modeling complex manufacturing areas such as photolithography. These methods have to be fast, flexible and adaptable in order to be valid for the dynamic environment. A major issue is how much time one needs to make a scheduling decision. Thus, while the models formulated and solved by deterministic

scheduling techniques are often removed from reality. Simulation modeling comes with a solution to many of these problems, with a suitable attention to the structure of the particular industrial problem at hand.
In general, there are basically three approaches in dealing with scheduling problems in semiconductor manufacturing: analytical techniques (e.g. Stecke [98], Raman *et al.* [99], Ulusoy *et al.* [100]), simulation (e.g. Denzler *et al.* [101], Sabuncuoglu *et al.* [102][103]) and artificial intelligence/expert systems (e.g. Kusiak *et al.* [104], Udo [105]). While each approach helps to provide a better understanding and solution to the scheduling problems, this research uses a hybrid model combining simulation and artificial intelligence to provide a system to address scheduling in the photolithography manufacturing area.
The next sections discuss three main approaches used in solving scheduling problem; simulation, AI, and hybrid models. In addition, Taguchi methodology is highlighted for design of experiments and as an optimization tool.

3.4 Simulation

Simulation is one of the most extensively used techniques in manufacturing in general and in complex industries (e.g. semiconductor manufacturing) in particular, see Appendix C. Simulation is an indispensable problem-solving approach for the solution of many manufacturing problems (e.g. Banks *et al.* [111], Hollier [112], Kochhar [113], Shannon [114], and Arisha *et al.* [115]).
Simulation has become a powerful approach in addressing FMS problems in last decade due to the rapid advancement in technology and the complexity in design of FMS's [116]. Therefore, simulation is extensively used especially for FMS's in semiconductor manufacturing. The reasons for this are the intractability of detailed analytical models of the semiconductor manufacturing process, the uncertainties inherent in the manufacturing process itself, and the steady improvement in computer technology which makes building simulation models easier and reduces the risk and the computational expenses.
Simulation models can also be developed at different levels of detail: a highly detailed model of a particular process step or workcenter, or more aggregate model of an entire facility or sub-system. The focus in this survey is on

scheduling and planning aspects in semiconductor manufacturing. Considerable effort has gone into the development of simulation models for wafer fabrication and their use in analyzing the effects of different control strategies and equipment configurations. Many of these efforts (up to 30 publications) have been discussed in Uzsoy *et al.* (1994) [117]. In this section, a number of applications (after 1994) concerning scheduling and planning are presented briefly.

Moench *et al.* [118] presented a simulation study for the solution of load-balancing problems in a semiconductor facility. However, they recommended more research to improve the performance of their local search algorithm. Hunter *et al.* [119] simulated a full-scale semiconductor manufacturing plant. The model was able to provide an overview picture of the interactions between different manufacturing areas as well as detecting the bottlenecks.

Ignizio [120] used simulation to build a template model that permits the user to configure and analyze a given toolset, without need to have any experience in simulation. Mackulak *et al.* [122] used simulation for comparing automatic materials handling system (AMHS) performance in semiconductor fabrication facilities. As the semiconductor industry moves towards 300-mm manufacturing, the design of AMHS's becomes a significant issue as well as testing the performance of the new system. Lin *et al.* [123] analyzed the performance of a double-loop interbay AMHS in wafer fab by considering the effect of the dispatching rules. Sivakumar *et al.* [124] presented a preliminary analysis of the relationship between selected input and output variables in semiconductor backend manufacturing systems, using a data-driven discrete-event simulation model. Vergas-Villamil *et al.* [125] proposed the application of a two-layer production control method to a discrete event simulation of a semiconductor reentrant line. It provides for real-time production control; however, they recommended additional optimization and scheduling studies for semiconductor reentrant lines in particular.

Dabbas *et al.* [126] proposed a modified dispatching approach that combines multiple dispatching criteria into a single rule with the objective of simultaneously optimizing multiple objectives. The combined dispatching has been implemented into a scheduler at Motorola's factories. They suggested a dynamic weight assignment algorithm in order to adapt to changing factory conditions. Dabbas *et al.* [126] have extended his research in order to combines more dispatching criteria into single rule. The research showed good results although further sensitivity analysis is needed to study more critical production parameters such as product-mix, start ratios, ..etc.

Dayhoff *et al.* [127] described a heuristic methodology (signature analysis) to characterize overall lot dispatches using three dispatch schemes i.e. earlier-steps-first, later-steps-first, and round-robin scheme. The simulation model provided the basis for comparison of the dispatch schemes, but the real dispatch schemes involved more interactions than those simulated. Atherton *et al.* [128] expanded the simulation work to include the inventory, cycle time and throughput in the form of trade-offs. Uzsoy has made a significant contribution to wafer fabrication research [129][130] and has set some optimization methodologies for test operations in semiconductor manufacturing as well as using decomposition methods for sequence dependent setup times in reentrant flow [131][132]. For scheduling, Uzsoy *et al.* [133] have described a number of considerations to evaluate and assess the quality of schedules. He [134] also examines benefits of turn around time or cycle time reduction using a stochastic simulation model.

Shen *et al.* [197] have proposed a stochastic dynamic programming model linked with simulation model for scheduling new releases and bottleneck processing by stage. MATLAB software is used to compute the optimal Fab scheduling. However, their paper focuses on layer-level release decisions, much work remains in translating the proposed linear release into on-line real-time dispatching decisions. In particular, it needs to evaluate the performance of the scheduling policies using a discrete-event Fab simulator.

Chern *et al.* [198] have proved general family-based scheduling rules to perform better than the individual job scheduling rule in terms of machine utilization. Five special family-based scheduling rules are constructed, of which FCFS-F, SRPT-F, EDD-F, and LS-F are modified from previous well-known scheduling rules. The simulation model is built to evaluate the performance of these five family-based rules. However, the research has not managed the complexity of the real manufacturing floor and simplifications have led the problem being reduced to a shadow of reality especially for manufacturing areas such photolithography.

Kim *et al.* [199] focus on production scheduling in Fabs that have different due date and different process flows. They considered three rules for releasing lots but not simultaneously. The shortcomings in this research are that they used limited number of rules and one set of data to verify. In addition, the model ignores some effective factors such as maintenance and variable yield rates. Kim *et al.* [202] also presented a simulation-based real-time scheduling (SBRTS) methodology to speed up scheduling decisions for unexpected events. In SBRTS method, lot scheduling rules and batch scheduling rules that give the best performance are selected from sets of candidate rules using results of simulation tests. Although the time required to respond to system disturbances is reduced without deteriorating the quality of schedules, further research is still needed to adapt the model for multistep reentrant flows. Also, more scheduling rules need to be considered to develop more effective lot scheduling.

Photolithography

Comparatively little research effort has been dedicated to photolithography, a bottleneck manufacturing area in semiconductor fabrication. Lee *et al.* [135] addressed production planning and scheduling problem in particular the shift-scheduling problem for photolithography. Their study provides empirical evidence that the sequential method yields satisfactory results.

Williams *et al.* [136] presented dynamic deployment modelling for photolithography WIP management in order to reduce capacity loss. Lachman-Shalem *et al.* [137] described an approach for the control of material handling within photolithography cells using a combination of genetic programming and nonlinear predictive control methods. Yoon *et al.* [138] address a deadlock-free scheduling method for material handling systems in photolithography cells. They employed the resource ordering approach to identify the potential deadlock in the track system and their proposed deadlock-free scheduling approach can be applied to track systems in the production of semiconductors, flat panel displays, disc-drive heads, etc. Karafyllidis [139] designed a dedicated parallel processor for the simulation of the photolithography process using a genetic algorithm. Starting from a cellular automaton with continuous state space that simulates the photolithography process, the GA is used to find a cellular automaton with discrete state space, having the smallest possible lattice size and the smallest possible number of discrete states. The results have not been validated and the detail of this complex process has not been included.

3.4.1 Modeling Approaches

Efficient modeling is the key to ensuring the success of any simulation project, as it is one of the critical tasks in simulation model building. The goal here is to develop structures for applying computer technology (e.g. simulation) to manufacturing systems and to use computer-based methods to better understand how best to improve manufacturing productivity and follow the production flow (items and information) in the system. As problems become larger, the preparatory phase of analyzing the problem becomes larger as well. For example, semiconductor manufacturing needs special requirements in modelling due to:

- complexity
- large amount of data
- reentrant nature
- rapid changes in product configurations due to demand
- manual intervention in the system

These factors make the systems analysis phase more difficult but they reinforce the need for a tool to enable systems modeling. GIGO – garbage in, garbage out – not only applies to data, it applies to the logic of the software and the implementation of a system [108].

A number of modeling methods have been developed to model manufacturing systems. Some tools such as flow charts and block diagrams concern physical process flow, while others such as IDEF, and data flow diagrams relate to process information flow.

3.4.1.1 Flow Diagram

Flow diagrams are one of the most common graphical methods to represent various activities and relationships within manufacturing systems. Although the use of flow diagrams is a simple method used in modelling, it is still widely used in most applications. Flow diagrams are limited in representing complex manufacturing systems because they do not provide useful information. However, they have been used along with other methodologies such as the decomposition approach to represent the major steps in the manufacturing process. Chapter five and six use flow diagrams to provide overview of the main operations within the manufacturing prcesses.

3.4.1.2 IDEF

One of the most effective tools in modeling complex industrial systems is the Integrated Computer Aided manufacturing (ICAM) DEFinition (or IDEF). It has been presented relatively quickly through the United States Air Force's (USAF) ICAM program, which started in 1977 [109]. The USAF was using contractors in the United States and Europe and required a common means of specifying systems and communicating results across the program. This led to the development of IDEF [110]. The ICAM program developed three well-documented modeling methodologies around the IDEF approach to system study:

1. A functional model of a manufacturing system and environment called IDEF0;
2. An information model of the system and environment called IDEF1;
3. A dynamics model to describe time-varying system behavior called IDEF2.

In this thesis, IDEF0 has been used in modeling the manufacturing process, see chapter five and six. A brief description of IDEF0 follows.

IDEF0 provides a formalized modeling notation for representing the type of functional analysis. The building blocks of the notation are shown in Figure 3.2.

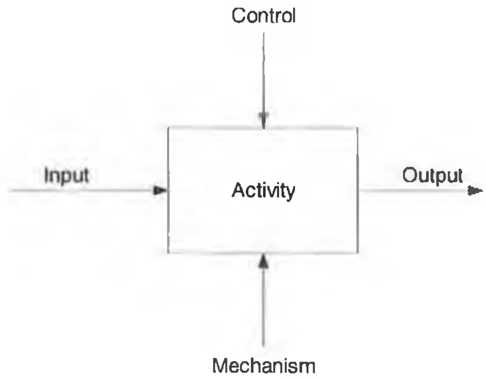


Figure 3.2: The IDEF0 building block

The inputs, outputs, controls, and mechanisms signified by the arrows can be resources, such as machines or equipment, material, data, information, people, products, etc. In other words, almost any aspect of an enterprise's operation. IDEF0 can be used for the purpose of identifying the data flows among a set of activities [108]. It also tackles the complexity of activity details by adopting a hierarchy of levels, with the lower levels giving progressively greater detail. The top level of an IDEF0 hierarchy comprises a single activity specified in a rectangle designated A0, as shown in Figure 3.3. The next level down, shown in Figure 3.4, has expanded this into three rectangles, A1, A2, and A3. The next level will similarly explode each of these three activities into A1.1, A1.2, A1.3, etc.

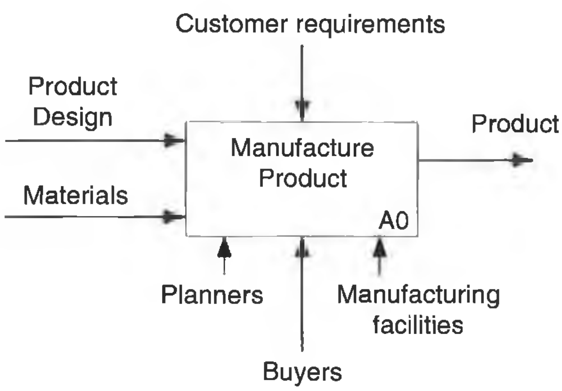


Figure 3.3: Example of A0 level

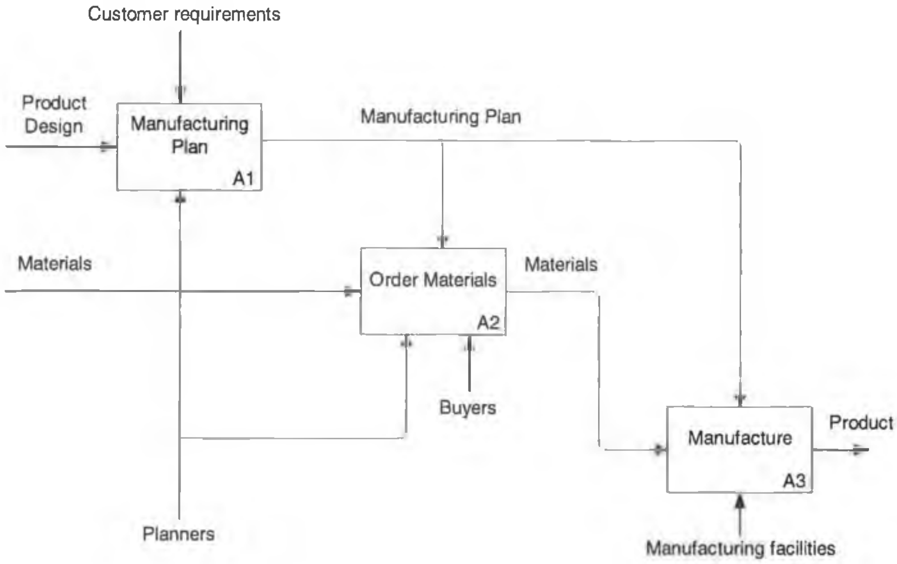


Figure 3.4: Example of second level of IDEF0

Discussion

Although those modeling methods (i.e. flow diagrams, IDEF) are adequate in many applications, they still have their pitfalls. Flow diagrams are too simple to contain all the required information needed for simulation model building especially for complex manufacturing systems. IDEF is an effective standard system approach, but are not easy to use by the non-specialist. There is therefore a need for a simple effective method to model complex manufacturing in a manner that allows any non-specialist to analyze the system and build a simulation model using an appropriate simulation software package. The approach which follows has been used in complex real-world applications (i.e. semiconductor manufacturing) and found to be efficient.

3.4.1.3 Schematic Approach for Simulation Modeling (SASM)

In order to make effective use of simulation in manufacturing systems, it is often helpful to develop a simple, intuitive model that describes the subsystem elements and the relationships among the elements in the simulation model. This thesis discusses the use of a new schematic approach for simulation modeling (SASM) aims not only to visualize problems or to gain an understanding of complex systems on a heuristic basis, but also to allow the non-specialist to translate the model into coded simulation model.

The SASM allows the user to understand the information flow with a preliminary perception of the nature of the system that can be used as a starting point for simulation. In other words, a simulation expert who does not know about the manufacturing system will be able to code the simulation model and explore all the interactivities of the system to provide further information and insights.

SASM provides a modeling notation for representing the behavior of systems in terms of entities that pass through a series of blocks (see Appendix G). The diagram indicates the information resources required for the activity and the way to model that in simulation. SASM is designed to specify the relationships between data. As with defining the relationship, this description of the system will make it easy for the simulation team to convert the SASM model into a simulation model using any software package available and capable to do the job.

In addition, any simulation user can understand SASM notations simply and hence this approach saves plenty time by letting the simulation team understand the process before building the simulation model. SASM has been used in chapter five and six as modeling approach in the simulation phase and it seems to be one of the promising modeling approaches for simulation. Figure 3.5 shows a simple example of information flow in one machine unit.

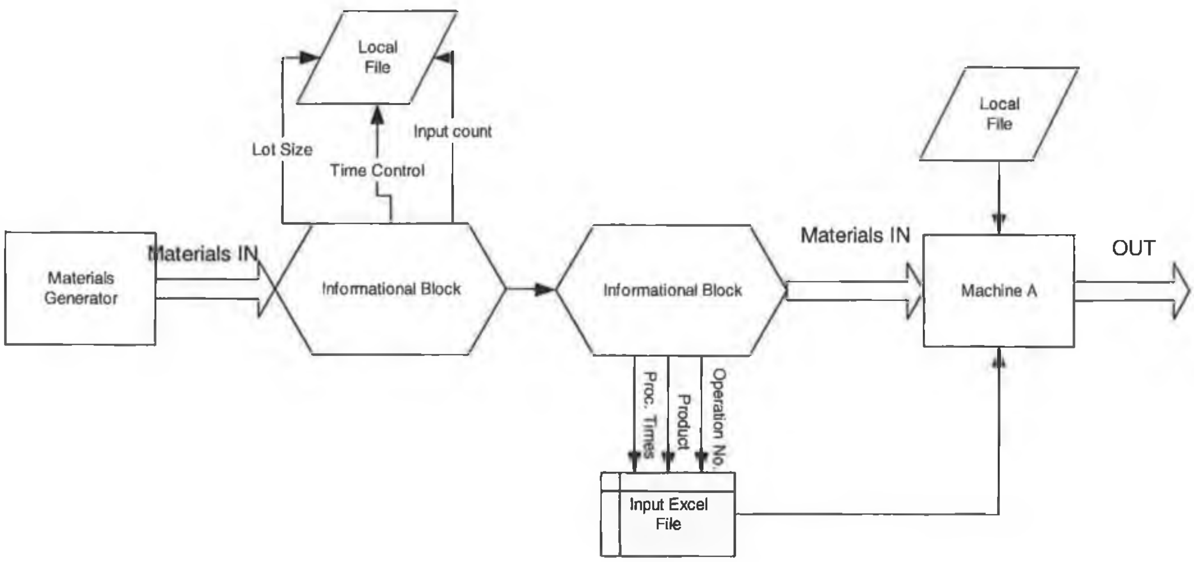


Figure 3.5: SASM Example

3.5 Taguchi methodology

The Taguchi approaches were initially used in quality engineering especially in the parameter design phase of product quality by design [203]. Taguchi developed orthogonal array (OA), interaction tables and linear graphs based on the design of experiments theory to study a large number of decision variables with a small number of experiments [179][204]. Moreover, he suggested more techniques which involve graphing the effects and visually identifying the parameters which appear to be significant. The schematic diagram of Taguchi's parameter design is depicted in Figure 3.6.

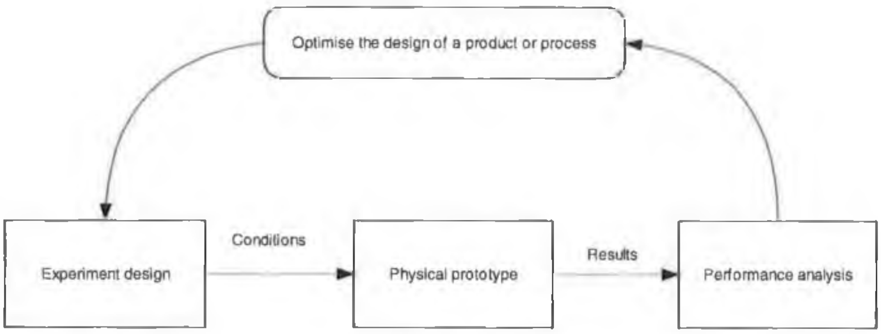


Figure 3.6: Taguchi's parameter design [203]

Taguchi methods were applied in many applications in industry in order to analyze the effect of process parameter and their interaction on the quality characteristic. Krishna *et al.* [205] used Taguchi method in cold solid state joining process.

The reduction of chemical usage in semiconductor manufacturing has been a topic of wide discussion over the past several years. Taguchi method was used by Namose [206] to optimize the process parameters in plasma reactions.

For job shop manufacturing system under constraint, Taguchi method was applied to the design phase and allowed the manufacturer to reduce the loss incurred by all part types according to their relative importance in the design of the system (Chen *et al.* [207]). Moreover, Taguchi methods were applied to reconfigure the design of robust manufacturing system through experiments for

the estimation of AGV speed in real-time circumstances (Mezgar *et al.* [208]). The use of Taguchi in flexible manufacturing systems were discussed in some papers such as D'Angelo *et al.* [210], and Arisha *et al.* [213].
The applications of Taguchi methods extends to use in optimization of neural networks, Khaw *et al.* [209]. More applications of Taguchi methods can be found in Antony *et al.* [204], Peace [211], and Clausing [212].
There are many applications of using Taguchi methods in solving industrial problems. This thesis has presented Taguchi method in a new application which is scheduling of photolithography area in semiconductor manufactuting. The coming sections show briefly the reasons to use Taguchi method integrated with simulation and explain the limitations on Taguchi methods.

3.5.1 Why Taguchi and simulation?

The inference between Taguchi's parameter design and the activities of simulation can be expressed briefly as follows:
In Taguchi methods, we seek to improve or establish the design of a product or production processes using physical prototype. In simulation approaches, we attempt to establish or improve the design of a system (such as manufacturing system, facility performance, etc.) using simulation models.
In Taguchi methods, the measures of performance are the desired functional characteristics of the product. In simulation approaches, the performance measures are the model performance measures for the system being designed or operated.
In Taguchi methods, the number of experiments required is very economic. In simulation, the time is one of the quality measures of the simulation model performance.
In Taguchi methods, stochastic variables cannot be optimized used the standard procedure. In simulation, most of simulation models can sort out variables with stochastic nature.
In Taguchi methods, the use of orthogonal arrays can systematically lead to optimize the parameters of the selected process. In simulation, trial and error

method is the most common approach used because of the ease with which parameters are changed and new solutions calculated.

In the Taguchi method, the significance and the relative significance of the selected process parameters can be obtained directly from graphs or tables within an economic number of experiments. Using simulation models, the significance is not guaranteed as an output. However, simulation can provide a sensitivity analysis quicker than Taguchi methods.

The previous discussion emphasize the importance of the integration of Taguchi's methods and simulation to provide quality solution in an economic time. There are more advantages of applying Taguchi methods together with simulation models can be found in Antony *et al.* [204] and Law *et al.* [214]. Simply, we can conclude "Taguchi is the manual for using simulation tool".

3.5.2 Limitations on Taguchi Methods

Taguchi methods have their limitations similar to any other optimization technique. Taguchi methods have limited capabilities to directly handle variables in stochastic optimization problems (Parks [215]). That's one of the advantages in integrating simulation models with Taguchi methods. Continuous variables have to be converted into a discrete form that Taguchi methods can optimize, however, the approximation leads sometimes to a considerable reduction in the solution quality. Furthermore, the selection of the orthogonal array and the levels of the factors can have a significant impact on the outcome of Taguchi analysis. Added to that Taguchi matrix say nothing exactly about the intermediate values between levels. Therefore, after the taguchi analysis we might need to use simulation models to examine one parameter or more in isolation.

3.6 Intelligent Scheduling

Since the decision-making process in the advanced manufacturing system environment in general and in scheduling problems in particular is getting more difficult and indeed, overwhelming for humans, Artificial Intelligent (AI) is widely used to assist human efforts. AI and intelligence mechanisms have provided several techniques to help solving scheduling problems. Intelligent scheduling as an optimization problem with an objective to allocate a limited amount of resources to a set of tasks has been subjected to many AI approaches as mentioned in chapter two. The following sections highlight the approaches which are used in this thesis.

3.6.1 Intelligent Scheduling using Artificial Neural Network

In the last decade, neural networks have gained more and more acceptance and been used more in the successful implementation of projects in manufacturing [105] [106][107]. Artificial Neural Networks were defined by Kohonen (1988) in ref. [140] as 'massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with objects of the real world in the same way as biological nervous systems do'. Figure 3.7 shows a simple architecture of a typical artificial neural network. The basic components of any neural network (NN) are neurons and weighted connections.

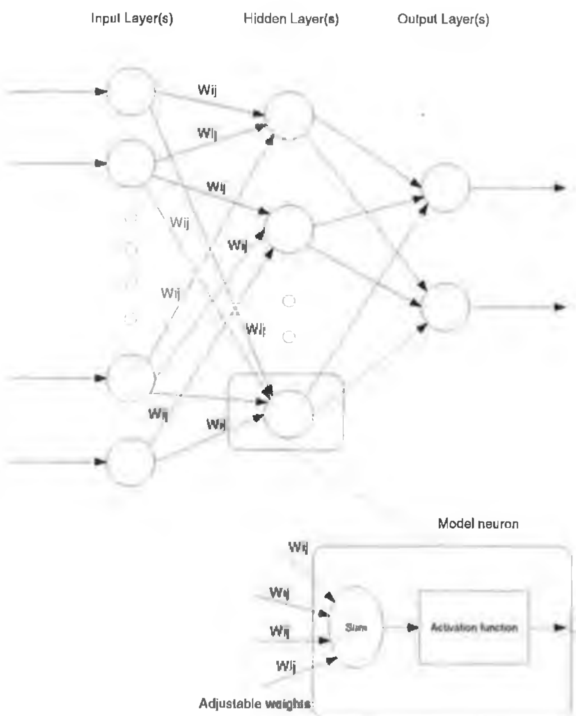


Figure 3.7: Topology of a typical ANN [141]

There are different types of neurons that can be used in neural networks. NN may be distinguished based on the directions in which signals flow. There are two basic types of NNs, namely, feedforward networks and feedback networks. In feedforward network, signals propagate in only one direction from an input stage through intermediate neurons to an output stage. While in a feedback network, signals may propagate backward from the output of any neuron. The neural network shown in Figure 3.7 is a feedforward network.

NNs (Gurney [140]) are based on ideas about how the brain may work. They look for patterns in training sets of data, learn these patterns, and develop the ability to correctly classify new patterns or to make forecasts and predictions. Therefore, they excel at problem diagnosis, decision-making, prediction, and other problems where pattern recognition is important and precise computational answers are not required. Input stimuli (e.g. the parameter values encountered in a problem situation) are connected through a network of nodes to output nodes (i.e. solution). This technique has been widely used in many classification and optimization situations. Historic data are used to "train" the network, automatically determining the most appropriate configuration of the hidden network. Perhaps the most attractive property of neural network structures is their ability to adapt patterns (weights) within data input array. The learning procedure commonly consists of repeatedly presenting inputs which the output is known and comparing the output of the system to the desired output. The network begins by finding relationships between the inputs and the output. Weight values are assigned to the links between the input and output neurons, Figure 3.7, so that the outputs match the training data. After those relationships are found, neurons are added to the hidden layer so that nonlinear relationships can be found. Input values in the first layer are multiplied by the weights and passed to the second (hidden) layer. Neurons in the hidden layer "fire" or produce outputs that are based upon the sum of weighting values passed to them. The hidden layer passes values to the output layer in the same fashion, and the output layer produces the desired results (predictions) [147]. The network "learns" by adjusting the interconnection weights between layers. The answers the network is producing are repeatedly compared with the correct answers, and each time the connecting weights are adjusted slightly in the direction of the correct answers. Additional hidden neurons are added as necessary to capture features in the data set. Eventually, if the problem can be learned, a stable set of weights evolves and produces good answers for all of the sample decisions or predictions. The real power of neural networks is evident when the trained network is able to produce good results for data that the network has never "seen" before.

The advantages of NNs are that they can achieve high computation rates by employing a massive number of simple processing elements with a high degree of connectivity to provide a new approach for optimization problems. More specifically, feedback networks provide a computing model capable of exploiting fine-grained parallelism to solve a rich class of optimization problems (Zhang *et al.* [141]). NNs have been successfully applied to the solution of constrained optimization problems. The scheduling problem can be solved using NNs to reduce the computational complexity [142]. The use of NNs combined with an interface-driven mechanism may provide a robust model capable of data interpretation as well as decision support. Chapter five provides a real-world application using NNs as an optimization tool in intelligent scheduling.

Many scheduling problems can be formulated as linear or non-linear programming problems, then solved using optimization approaches. However, solving the combinatorial optimization problem often attains a local optimum solution depending on the initial state of the network and the training data. Recently, some stochastic neural network models have been proposed for avoiding convergence to local minima. There are many literature contribution towards solving scheduling and manufacturing problems using NN (e.g. Hopfield [143], Maa *et al.* [144], Vaithyanathan & Ignizio [145], and Arizono *et al.* [146]).

3.6.2 Intelligent-agent based scheduling approach

Unlike traditional manufacturing scheduling systems using a centralized scheduler, an agent-based manufacturing system supports distributed scheduling such that each agent can locally handle the schedule of its machine/machine centre, operator, robot, or station. However, the participating agents can collectively perform global scheduling through some mechanisms and/or protocols.

3.6.2.1 Advantages of Intelligent-agent based approaches

Agent-based scheduling approaches have several potential advantages for manufacturing scheduling [148][149]:

- The agent paradigm makes integrating process planning and manufacturing scheduling easy to realize the simultaneous optimization of manufacturing process planning and scheduling.
- It is possibile to carry a connection between resource agents directly to their represented physical devices, to realize real-time dynamic rescheduling. This can provide high fault tolerance.
- Agents can develop schedules using the same mechanism that businesses use (negotiation rather than simple search) in the manufacturing supply chain. Thus, direct connection between manufacturing capabilities of different manufacturing enterprises can ease the scheduling process and make optimization possible at the supply chain (enterprise) level as well as the shop floor level.
- The combination of agent-based approaches with other techniques at certain levels for learning and decision-making might promote better solutions.

Intelligent agents are seen as the solution for integration between manufacturing activities [150]. Intelligent agents are used in distributed AI. They allow the co-ordination of local AI systems distributed throughout the manufacturing process (i.e. production, scheduling, inventory and maintenance etc.) and throughout the business as a whole (e.g. marketing , product design, operations, finance and personnel etc.). Negotiation between the separate AI systems, each with their own set of local optima or preferences, enables the selection of policies more closely aligned to the objectives of the manufacturing business (e.g. [151]).

3.6.2.2 Major Issues of Intelligent-agent based approaches

Anyone developing an agent-based manufacturing scheduling system might deal with some or all of these four main issues among others: encapsulation, coordination and negotiation protocols, system architectures, and decision schemes for individual agents.

Encapsulation

Among the different approaches for agent encapsulation in manufacturing scheduling systems, two are important: functional decomposition and physical decomposition. Functional decomposition uses agents to encapsulate modules assigned to functions such as order acquisition, planning, scheduling, material handling, transportation management, and product distribution. No explicit relationship exists between agents and physical entities. Physical decomposition uses agents to represent entities in the physical world, such as operators, machines, tools, products, parts, and operations. An explicit relationship exists between an agent and a physical entity. Both approaches have distributed implementations.

Functional decomposition tends to share many state variables across different functional agents. This can lead to inconsistency and unintended interactions. Physical decomposition naturally defines distinct sets of state variables that individual agents with limited interactions can manage efficiently. However, it needs a large number of resource-related agents, which can lead to other problems such as complex management. However, functional decomposition is useful in integrating existing systems (for example, CAD tools, materials requirements planning systems, and databases), so as to resolve legacy problems.

Corresponding to these two agent encapsulation approaches, there are two types of distributed manufacturing scheduling systems.

In the first, scheduling is an incremental search process that can involve backtracking (e.g. Sycara *et al.* [152], and Burke *et al.* [153]). Agents, responsible for scheduling orders, perform local incremental searches for their orders and might consider multiple resources. The system merges the local schedules to produce a global schedule, similar to centralized scheduling.
In the second system, an agent represents a single real-world resource (for example, a work cell, a machine, or even an operator) and maintains this resource's schedule. This agent might negotiate with other agents to carry out overall scheduling. Most agent-based manufacturing scheduling systems use this approach and it is described in detail in chapter five.

Coordination and negotiation protocols

Systems that use functional decomposition are similar to traditional integrated systems; they usually use a predefined coordination mechanism. The coordination has been discussed in detail in Sycara [152] and Smith [154]. The negotiation protocols are somewhat beyond the scope of the thesis.

Architectures

Agent system architectures provide the frameworks within which agents are designed and constructed. Architectures for agent-based manufacturing systems come into three main categories: hierarchical, federated, and autonomous [153]. The focus in this thesis is hierarchical structures. The hierarchical structure is typical of most manufacturing enterprises with many levels of control over resources and with different information requirements. The use of hierarchical structures to simulate the functional areas of manufacturing systems has many advantages when using a decomposition approach.

Decision Scheme

In most scheduling situations, the system needs to compare, negotiate, and compromise the alternatives. Each step in scheduling should have information about the capabilities, availability, and the entire system configurations. Using the intelligent-agent based approach to support decision-making depends on many factors e.g. knowledge of previous cases, system updates, and production policies.

3.6.3 Hybrid Intelligent Scheduling Models

The current trend of hybrid intelligent scheduling models is towards a combination of the three common solving approaches; Operations Research-based, simulation-based and AI-based. It is suspected that a synthesis of paradigms will be required (e.g. [89] and [155]). Chen et al. [201] have an embedded search strategy over a colored timed Petri net (CTPN) for wafer fabrication. Through the CTPN model, all possible behaviors of the wafer manufacturing systems such as WIP status and machine status can be completely tracked down by the reachability graph of the net. The simulation tool, UltraSim, provides a friendly user interface. Although the model outputs are more superior than the conventional dispatching rules, the model could not reach the level of detail that scheduling problem needs in wafer fabrication. Sample of efforts to use a mixture of several of the above paradigms are shown in the Table 3.1.

Table 3.1: Sample of research using Hybrid techniques

Author(s)	*Hybrids Techniques*	*Notes*
Sereco *et al.* [156]	KBS	Optimization techniques, hierarchical planning, and heuristic search
Dagli *et al.* [157]	Lawler's Algorithm & NN	Algorithm generates schedules to train NN
Rabelo *et al.* [158]	ES & NN	IFMSS: intelligent FMS scheduling, expert system and a back propagation NN
Rabelo *et al.* [159]	IFMSS	Enhancing the model with adding simulation and GA to his control architecture
Yih *et al.* [160]	AI& Simulation	Hybrid model of AI and simulation for a small set of candidate scheduling heuristics
Yih *et al.* [161]	Semi-Markov & ANN	Semi-Markov optimization and ANN for robot scheduling in a circuit board production
MacCarthy *et al.* [162]	LP & Simulation	Rule-based framework; mathematical optimization procedure and simulation.
Sim *et al.* [163]	ES & NN	Expert system to train NN to reduce the time required for training.
Szelke *et al.* [164]	CBR & Machine Learning	Reactive learning of machine for shop floor scheduling
Kim *et al.* [165]	Inductive Learning & NN	Multi-objective FMS schedulers
Lee *et al.* [166]	GA & Machine Learning	To generate empirical results using machine learning for releasing jobs into the shop floor and GA to dispatch jobs.

3.7 Proposed Methodology for Intelligent Scheduling

In fact, no single modeling paradigm currently appears to offer the basis for a unified theory of shop scheduling or to provide an appropriate calculus for generating schedules, or even to support a complete representation of the attributes of a complex shop scheduling environment.

Operational research, simulation and AI representation schemes appear to be capable of capturing a wide range of problem attributes but fail as yet to provide good insights on workable solution strategies. Machine sequencing, resource-constrained scheduling, and AI search techniques offer insight into possible solution approaches but do not address the richness of the complex scheduling environment.

Most manufacturing systems are too complex to allow realistic models to be evaluated analytically (Rodammer *et al.* [89]). As an alternative simulation-based scheduling can provide an effective tool for shop floor scheduling while requiring few assumptions. The schedules generated must be based on an accurate, realistic model of the production facility. It can allow various dispatch rules, or decisions regarding the system to be tried out and selected based on simulated performance results. Simulation-based scheduling has gained acceptance from both researchers and practitioners (see Ref. [167], [168][169][170]).

This research has found that Taguchi methods are completely suitable for the activities of simulation based on the literature review and the problem definition. It can be seen in the proposed methodology could comprise simulation approach with Taguchi method in a new application of scheduling in semiconductor manufacturing (see Chapter four). In addition, modeling approaches and AI have been used to deal with the complexity of semiconductor manufacturing. However, this thesis provides a new methodology for scheduling the most difficult area of the semiconductor manufacturing, the photolithography area.

The first steps are used for system modeling; a simulation model is built and executed using an appropriate verification and validation phase. Taguchi methods are used for planning and conducting simulation experiments and then optimizing the selected parameters. The experimental results are analyzed with the Taguchi method and a confirmation experiment is carried out to validate the result. The proposed methodology is a generic approach and the activities of simulation do not necessarily contain all these steps in the same order stated. Furthermore, it is worth noting that such a simulation study is not a strictly sequential process.

Artificial neural networks have shown good promise for solving combinatorial optimization and constraint satisfaction problems like shop floor scheduling. Extensive research has shown the capabilities of neural networks for automatic learning, association, generalizing and pattern recognition through their ability to provide non-linear transformations to model highly complex functions. In addition, the approach does not require the strong underlying assumptions on the structure of the data which are required by many traditional techniques. The complex interactions of the dynamic problems on the shop floor are therefore candidates for using neural network to learn and perform dynamic scheduling for a shop floor. The only disadvantage of neural networks is that "knowledge" in the network is not easily available to the user.

Therefore, the proposed methodology uses these effective tools; simulation, neural networks, intelligent algorithm, Taguchi's paradigm, and modeling approaches (e.g. IDEF0) for shop scheduling has been set up and applied to solve a real-world application (chapter four and five). This intelligent methodology is shown in Figure 3.8, while the main phases in the methodology are given in Table 3.2.

Table 3.2: The main phases in the proposed metholodgy

<table>
<tr><th>Phase</th><th colspan="2">Activities</th><th>Functions</th></tr>
<tr><td>I</td><td colspan="2">Problem Definition</td><td>• Identify Scheduling problem
• Find system constraints
• Set assumptions/approximations</td></tr>
<tr><td>II</td><td colspan="2">Objectives</td><td>• Set scheduling objectives (with management)
• Objectives agreement (production/manufacturing staffs)
• Select Performance measures</td></tr>
<tr><td>III</td><td>Model Building</td><td rowspan="2">Data Collection</td><td>• Data Collection phase starts
• Model Building using modeling tools (e.g. IDEF, SASM)
• Intelligent model planning phase (AI tool selection)
• Planning for experiments</td></tr>
<tr><td>IV</td><td>Model Coding</td><td>• Building simulation model
• Software assumptions and constraints are considered
• Set coordination with the intelligent-agent
• Data collection phase ends</td></tr>
<tr><td>V</td><td colspan="2">Verification/Validation</td><td>• Verifying the simulation model
• Validate simulation outputs if possible
• Verification/Validation for the intelligent-agent</td></tr>
<tr><td>VI</td><td colspan="2">Experimentation</td><td>• Experimenting within Taguchi's framework
• Number of experiments
• Repetitions of experiments</td></tr>
<tr><td>VII</td><td colspan="2">Results Analysis</td><td>• Results analysis
• Determine significance of selected parameters
• Review results with production staff</td></tr>
<tr><td>VIII</td><td colspan="2">Optimization</td><td>• Optimizing selected parameters by finding best combinations
• Apply AI techniques for optimization</td></tr>
<tr><td>IX</td><td colspan="2">Sensitivity Analysis</td><td>• Further experiments for sensitivity analysis
• Sensitivity analysis of intelligent model</td></tr>
<tr><td>X</td><td colspan="2">Enhancement</td><td>• Improvement of model performance (simulation time, model size, coordination)
• Modifications for enhancement (user-friendly)
• Building knowledge base is recommended</td></tr>
</table>

3.8 Conclusions

For this review of past research relevant to scheduling in semiconductor manufacturing systems, a number of conclusions can be drawn:

- Within semiconductor manufacturing systems, scheduling remains among the most important and challenging tasks that must be performed routinely. Furthermore, within this scheduling of photolithography is the most complicated task due to the complex process flow and parameters.

- Developing a schedule involves designating the resources needed to execute each operation of the process routing plan and assigning the times at which each operation in the routing will start and finish execution and should be performed intelligently to increase system efficiency.
- Many scheduling tools have been developed to support decision-making process. Simulation is a powerful technique to model complex manufacturing systems. The benefits of using simulation models have allowed the planner to easily experiment with new scenairos.
- Modeling tools such as IDEF0 and SASM are the key elements to successful of simulation development models.
- Intelligent scheduling models have received attention from many researchers in last decade. Results from these techniques provide better solutions, in terms of satisfying solutions than traditional techniques. However, hybrid intelligent scheduling models show more comprehensive solutions for complex scheduling problems.
- From the literature, it can be seen that there is a need for a generic methodology for an intelligent tool that has the purpose of optimizing the scheduling activity in the areas of complex manufacturing systems (e.g. semiconductor fabrication processes).
- A hybrid intelligent scheduling (simulation, intelligent-agent based approach, and Taguchi's paradigm) has been chosen as the framework on which to develop a shop scheduling model of the the photolithography process in semiconductor.
- Simulation itself cannot provide optimal solutions, it must be driven in an intelligent manner. The Taguchi experimental methodology can provide a good framework for applying simulation to better understanding and optimize a manufacturing process.

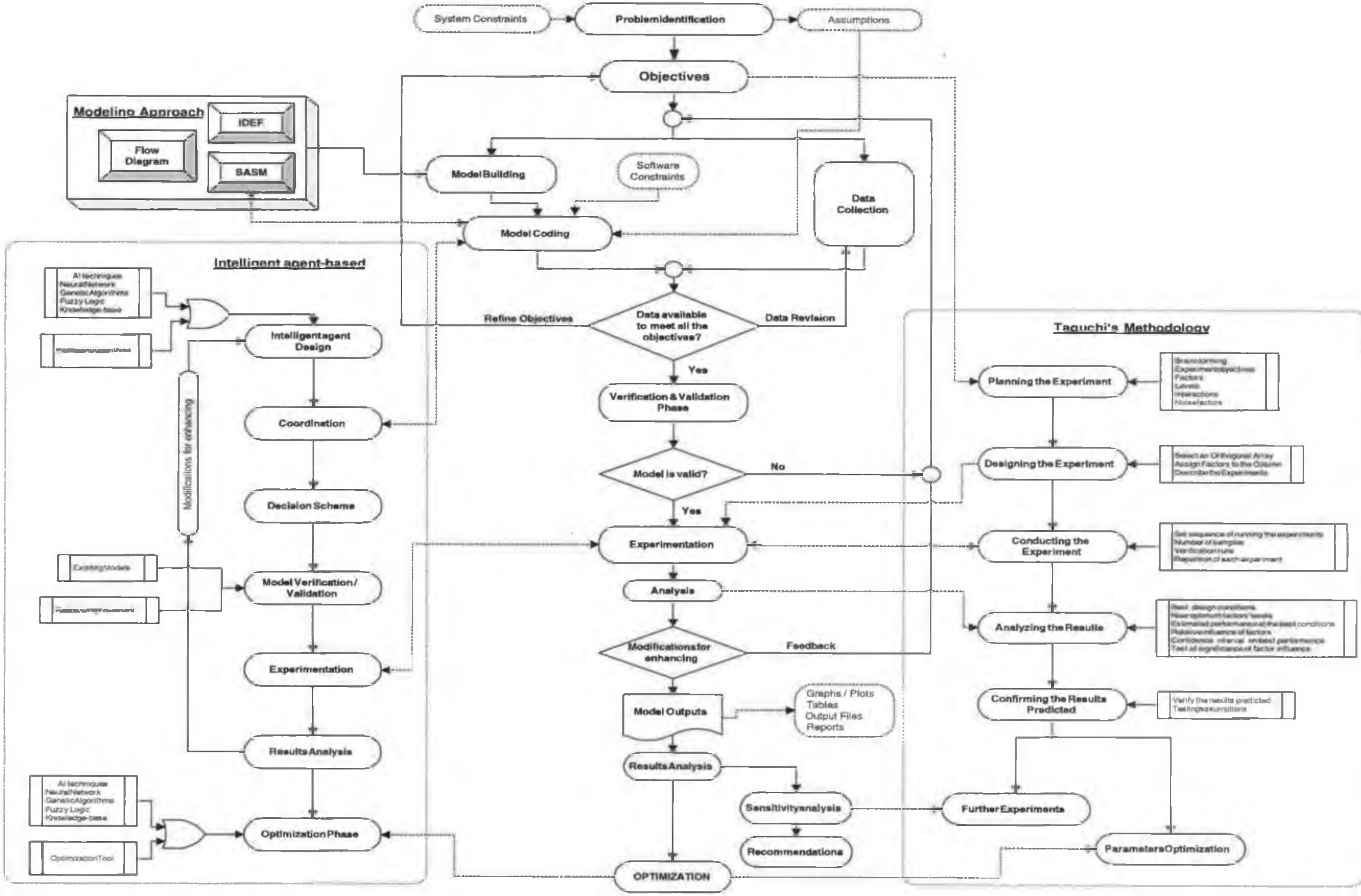
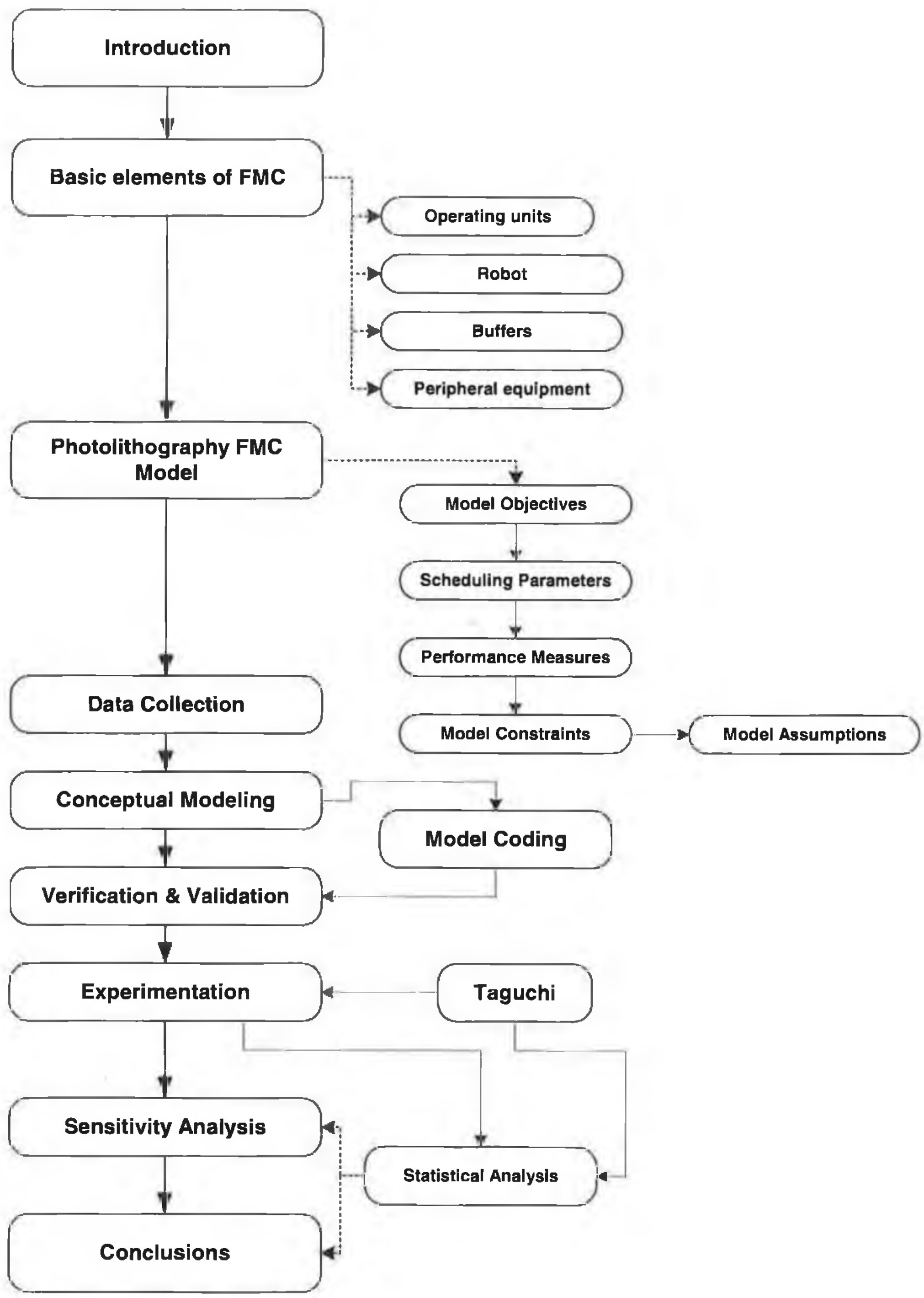


Figure 3.8: New methodology for intelligent shop scheduling

Chapter 4

Scheduling of Photolithography FMC



Chapter 4

Scheduling of Photolithography FMC

4.1 Introduction

The complexity of manufacturing high capacity semiconductor devices means that it is impossible to analyze the process control parameters and the production configurations using traditional analytical models. There is, therefore, an increasing need for effective models of each manufacturing process, characterizing and analyzing the process in detail, allowing the effect of changes in the production environment on the process to be predicted. Production scheduling of FMC's in such a dynamic environment is a complicated task due to the complex nature of wafer processing, constraint operations, product diversity, and resource costs. It is essential to characterize the performance of these tools in detail to examine the production scheduling plans before they are finalized. Experimenting in the production plans on the real floor is non-option due to the high cost of equipment and the sensitivity of the process [196]. Using state-of-the-art computer simulation, a structured modelling methodology and Taguchi's methodology, a generic model of photolithography flexible manufacturing cells has been developed and used to mimic the actual performance of the tools.

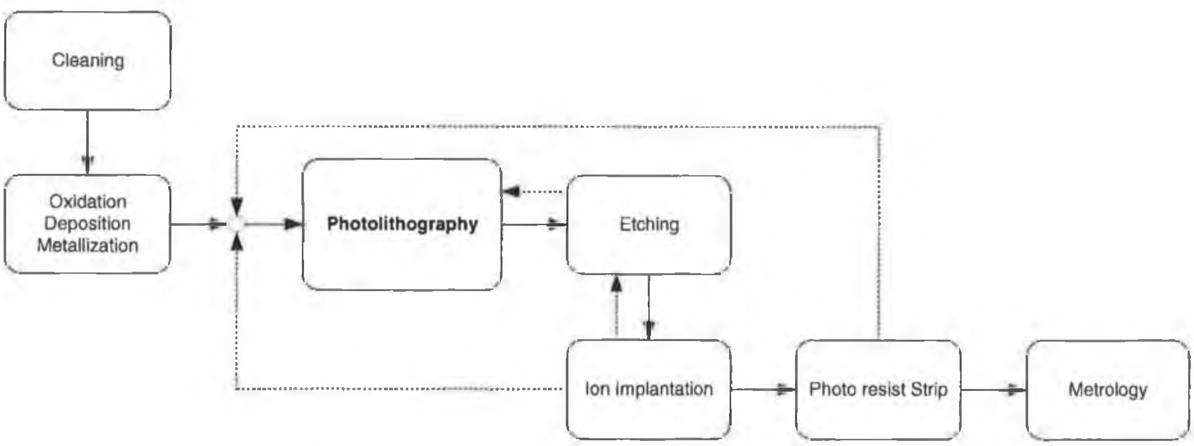


Figure 4.1: Photolithography process within the wafer fabrication processes

The Photolithography process is considered the constraint process within semiconductor manufacturing due to its complex technology, critical dimensions, re-entrant flow, and product/layer sensitivity [117][172]. The photolithography process, a central process in wafer fabrication processes (Figure 4.1), involves the processing of wafers in order to build up the layers and patterns of metal and wafer material to produce the required circuitry. During the photolithography process the circuit pattern is transferred from a mask onto a photosensitive polymer and finally replicates the pattern in the underlying layer. The object of this process is the accurate and precise definition of a three-dimensional pattern on a semiconductor substrate. The basic photolithographic sequence is shown in Figure 4.2. Typically, the lot to be processed goes through a coating operation, where the wafers are coated with a photo-resistant substance. The lot is then moved to the exposure operation where the patterns are projected on the wafers. The exposed wafers are moved to the developing operations. Once these steps are completed, the lot typically is moved to post-photolithography analytical operations. The amount of metrology is dependent on the product and the layer being processed. Details of the three basic fabrication steps in photolithography are described later in *section 4.5.2*.

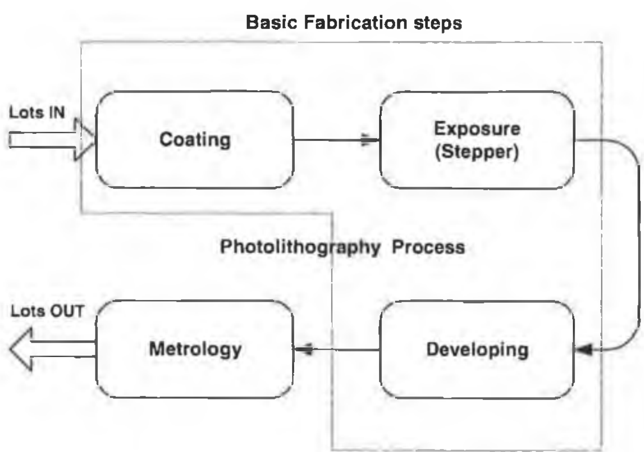


Figure 4.2: Simplified diagram of typical photolithography process flow

This chapter deals with the FMC by using a decomposition approach [132] and [173], and then integrates the output with the whole manufacturing system. The photolithography area in the factory under study includes three different types of cells. These can perform the same jobs with variation in capability, speed, and efficiency. The deficiencies of using traditional modeling techniques for predicting the performance of FMC's were clear when compared with actual production performance [174]. Mathematical models and deterministic approaches were somewhat acceptable in small and simple processes where the simplifications and assumptions fit well.

The primary objective of this chapter is to establish a framework for simulation-based approach combined with Taguchi's procedure for the experimental design to provide management with an effective decision support system (DSS). The impact of changing many scheduling parameters on the performance of the FMC needs to be evaluated. First, a detailed simulation model was built to characterize the flexible manufacturing cell in detail, then the effect of scheduling, planning and control problems on the cell performance was examined.

4.2 Basic elements of FMC

4.2.1 Operating units

The operating units are the workstations that are responsible for processing, inspecting, assembly, and control. Most of these units are computer numerically control (CNC) machines. The function of the operating units varies depending on its use.

4.2.2 Material Handling

The material handling equipment is considered one of the most critical elements in the flexible manufacturing systems in general and FMC in particular. The main characteristics of material handling systems include speed, number of degrees of freedom, payload, external sensors, intelligence of control system, response time, accuracy, reliability, and precision

Robots are used in many cells and their function might vary in every cell; they can perform basic assembly, material handling, processing, or inspection.

Some FMCs may have more than one robot where the cooperation between two or more robots is necessary to perform the jobs harmonically in the same working space. However, this complicates the control system [175]. In semiconductor manufacturing, and especially with photolithography tools, there is always one robot that is responsible for delivering the wafers to the operating units. The intelligence of the robot control to handle the wafers and the information can have a significant effect on saving throughput time.

4.2.3 Buffers

On-site storage devices such as buffers, indexed rotating tables, or part feeders before or after the operating units are key elements in FMC. The capacity of these buffers has a significant effect on product cycle time especially with different product mixes in the same cell. The main objective of the buffers is to maximize machine utilization.

4.2.4 Peripheral equipment

The many different types of complementary equipment required for manufacturing processes include:

- Accessories required for the operations themselves.
- Tools that execute special functions (e.g. alignment)
- Gripper / Fixtures to hold the wafers in the place precisely during operating.
- Sinlge wafer buffer.
- Batching / Un-batching devices at the beginning and end of some operations.

4.3 Photolithography FMC Model

The photolithography FMC model has followed the proposed methodology discussed in chapter three. The model presents a comprehensive integration between three analytical techniques such as IDEF and SASM (for modeling), simulation, and Taguchi's methodology in order to accomplish the objectives below.

4.3.1 Model Objectives

The model objectives are to:

- Simulate in detail the manufacturing cells in the photolithography area.
- Evaluate the actual performance of FMC in the area using a multi-criteria approach.
- Determine bottleneck operations/steps in the cell.
- Examine the impact under different loading conditions of product-mix/volume on the FMC performance.
- Study the effect of various scheduling strategies on the system.
- Develop a DSS to predict the impact of policy decisions on key manufacturing system parameters, like product cycle time, tool throughput, WIP inventory, and utilization of each device.
- Determine the significance of the production and planning parameters on performance measures.
- Determine the sensitivity of the performance to the assigned parameters (in terms of their main factor effects).
- Determine appropriate or near optimum combinations of the parameters for better shop performance.

4.3.2 Process Parameters

In most of the cases, the photolithography process can run uninterrupted after wafers are loaded into the manufacturing cell. However, configuration changes sometimes interrupt the processing. In this study, the production staff selected the following key process scheduling parameters;

1) Wafer starts(WS), which is the number of new wafers starting production in a specified time period.
2) Product-mix (PM), which is the number of different products that will be in production at the same time.
3) Product Sequence (PS), which is the sequence of the products in the production schedule.
4) Stepper Buffer Size (BS); which is the size of the buffer in front of the stepper (exposure step).

4.3.3 Performance Measures

The selection of performance measures depends on many parameters such as manufacturing application, nature of the production system, administration requirements. The performance measures included in this Photolithography FMC model are:

1. Throughput Time (*TPT*)
2. Product/Layer Cycle Time
3. Machine Utilization
4. Mean Completion Time
5. Waiting Times
6. Average Delays
7. Machine Idle Time
8. Production Rate

4.3.4 Model Constraints

Due to the complexity of photolithography operations, the constraints imposed on the model fall into two main groups; constraints due to the technology complexity, and constraints due to production. The main constraints imposed on the model are listed below:

- The sequence of operations

  The photolithography area is divided into three main operations: spin/coat, Align/Expose, and Develop. The sequence of these operations is fixed for the manufacturing of the products.
- Re-entrant nature

  Photolithography is performed at various times throughout the manufacturing process. That means a lot can visit photolithography many times in the production cycle but only return after it has been processed in other tools.
- Lot Integrity

  One of the most important factors in semiconductor manufacturing is lot integrity. Every lot should keep the same wafers throughout the production process.

- Operating Times

  The processing times differ from one cell to the next due to the variability in products, layers, and tool capacity.
- Photolithography manufacturing cells

  Each photolithography cell is considered to be a flow shop environment as every single product has to be processed on each operation inside the cell following the same sequence of operations.
- FMC Loading

  The loading of the cells can be either automated or manual. The cells receive the lots in boxes (max. 25 wafers per lot), although the operations can only be performed on a wafer-by-wafer basis. As a result, un-batching and batching functions are required in the cell to interface within the rest of the factory.
- Maintenance

  Preventive maintenance (scheduled maintenance) is specified on weekly basis. Unscheduled maintenance is estimated based on the recorded history of the cell.
- Preemption

  Lot order may only be changed by changing the order of their insertion into the cell. Preemption is not allowed in the cells.
- Setup Times

  Setup time applies every time the product/layer is different from previous lot to be processed for a lot.
- Storage Areas

  The storage areas (buffers) inside the cell employ FIFO (First In First Out) rule regardless of the lot priority.
- Metrology

  Photolithography metrology is an essential function to ensure the products are within specifications before leaving the cell.

4.3.5 Model Assumptions

The model assumptions have been classified into three main groups based on their source;

a) Simulation constraints,

b) Manufacturing constraints, or

c) Recommendations of manufacturing team.

Table 4.1, shows the main groups of assumptions and brief description of each one. It is worthwhile saying that all the assumptions were reviewed by the manufacturing team to verify the model logic.

Table 4.1: Classification of model assumptions

<table>
<tr><th>Group</th><th>Sub-group</th><th>Description</th></tr>
<tr><td rowspan="4">Simulation</td><td rowspan="2">Simulation software capabilities</td><td>- Lots priorities can be set in the start of the run.</td></tr>
<tr><td>- The global time unit in the model is seconds.</td></tr>
<tr><td rowspan="2">Simplification</td><td>- The initial setup is negligible.</td></tr>
<tr><td>- All parts (wafers) are available for processing at the start of simulation with the sequence required, although wafers entry into the cell is dependent on request from the cell sensor.</td></tr>
<tr><td rowspan="7">Manufacturing Constraints</td><td rowspan="2">Technological constraints</td><td>- Preemption is not allowed.</td></tr>
<tr><td>- The default lot size is 25 wafers.</td></tr>
<tr><td rowspan="2">Equipment Setting</td><td>- The setup time of changing reticle considers 300 seconds.</td></tr>
<tr><td>- Each machine or operating unit can process only one operation at a time.</td></tr>
<tr><td rowspan="3">Capacity</td><td>- Tools are available 100% at the start of the run.</td></tr>
<tr><td>- Limited size buffers are located before specific operating units.</td></tr>
<tr><td>- Buffer size can be variable.</td></tr>
<tr><td rowspan="5">Production Planning and Manufacturing Staff</td><td rowspan="5">Simplifications</td><td>- The models consider every cell is an independent unit that has its own input source.</td></tr>
<tr><td>- Processing times of every product/layer have included the material handling times.</td></tr>
<tr><td>- Machines are never unable to perform a required operation for lack of operator, tool, or raw material.</td></tr>
<tr><td>- Rework/Scrap products are considered as a percent overall of the production output.</td></tr>
<tr><td>- Preventive maintenance considered to be weekly based and the unscheduled maintenance has been considered as random distribution based on historical data from the floor.</td></tr>
</table>

4.4 Data Collection

The data collection phase is very critical in order to guarantee good output, and is also an essential step in modeling as a full description of each of the system entities must be obtained. Simulation output efficiency must always be judged based on the quality of the inputs. The input data for the model includes the processing times, maintenance schedules, material handling times, wafer arrival rates, product ..etc.

Operational time variability is one of the key parameters determining the actual utilization as well as average cycle time of wafers/lots [176]. Although the high level of automation involved in semiconductor manufacturing would suggest very little variations in processing times, many different sources of variability can be identified within the photolithography cells, such as wafer arrival rates, travel times between steps, breakdowns, changes in setup times due to product/layer changes and operator availability. The model consideres this variability in the system and uses the historical data to define the required times.

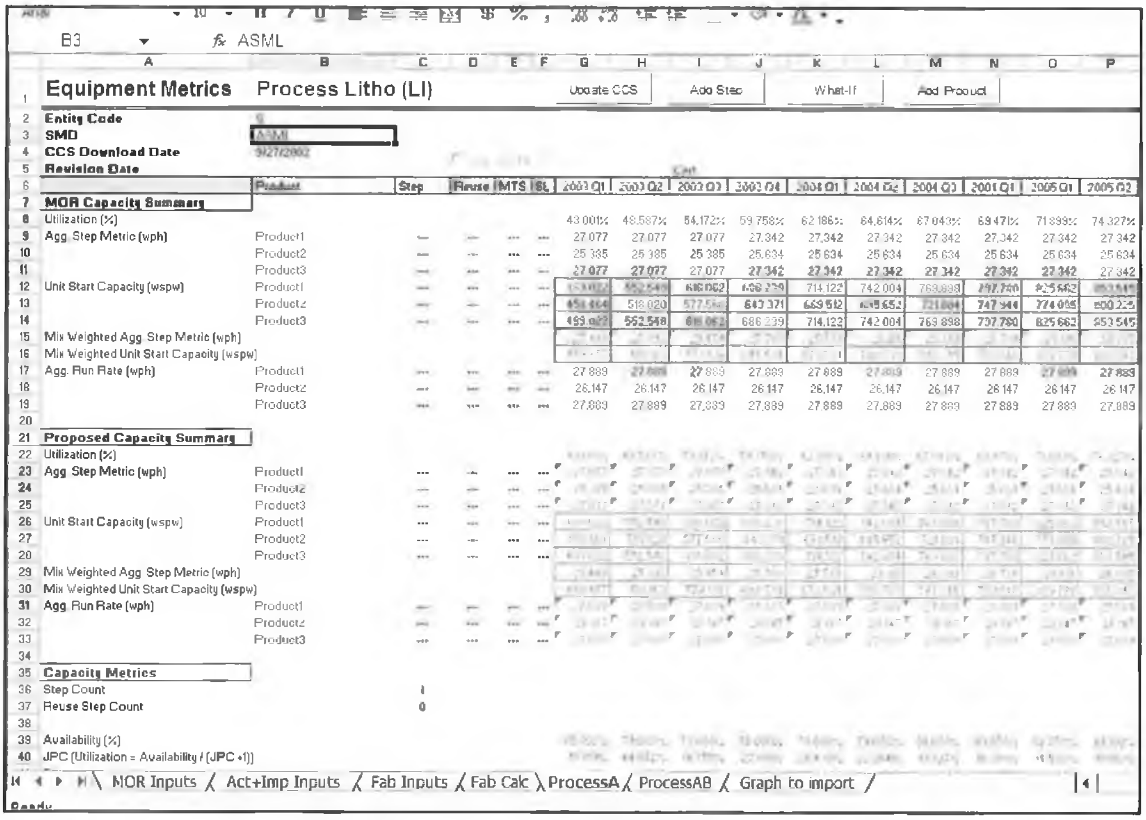


Figure 4.3: Snapshot of the input file (shopfloor data)

Most of data needed were saved in spreadsheet format are submitted to the model in same format after sorting them out, Figure 4.3 and 4.4. The probability distributions for the important system parameters have to be developed.

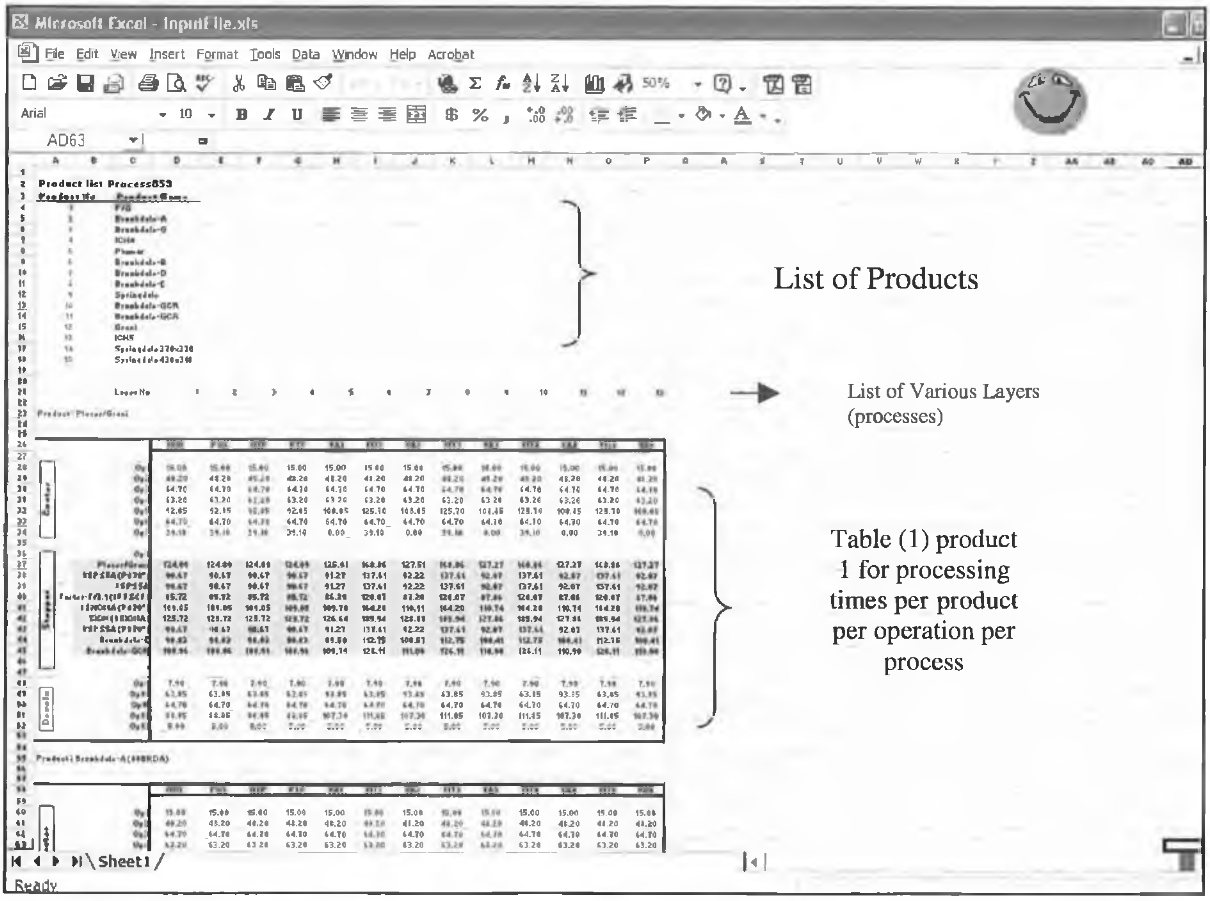


Figure 4.4: Snapshot of the input file (model input format)

The input data for the model have been classified into two main type: deterministic data and stochastic data. The available data from the factory floor have been used to provide the model with the information that needs to be processed to calculate the objective functions.

Deterministic data includes:

- Wafer starts
- Storage areas capacity
- Product sequence/Order
- Product-mix
- Required layers
- Operation sequence
- Nominal number of wafers per lot

Stochastic data such as:

- Processing times

- Maintenance schedule
    - o Preventive maintenance (PM)
    - o Unscheduled maintenance
- Wafers arrival times
- Material handling times

The stochastic items have been manipulated in order to use them in the model. For the sake of simplifying the form of the stochastic data, statistical distributions were assumed based on historical data collected from the shop floor. As indicated in the assumptions (Table 4.3), the material handling times have been included within the processing times. The processing times are assumed to have Normal distributions and are represented as mean and standard deviation times for each layer/product/tool combination calculated from the historical shop floor database. This must be completed before the model verification can take place. For illustration sake, a sample of the data collected for combined processing and material handling times for one single layer/product is shown in Figure 4.5 below.

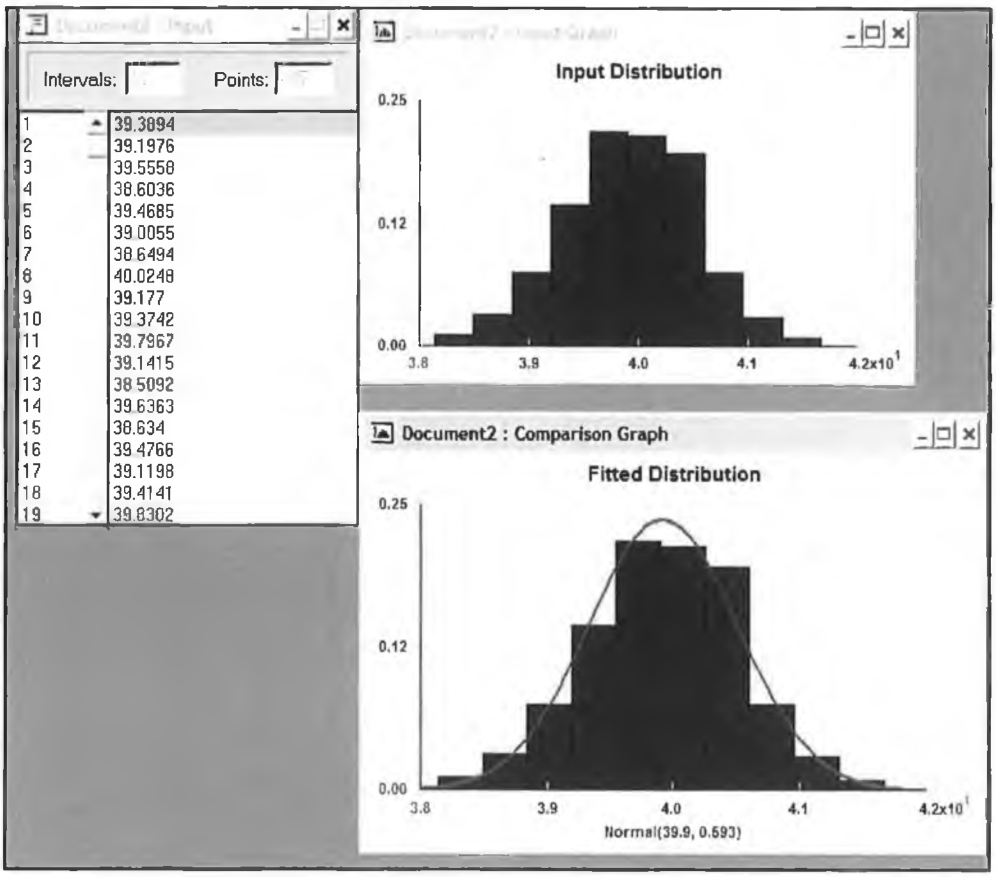


Figure 4.5: Snapshot of the Excel input file

A total of 195 such definitions are required to charaterise one photolithography tool. Each particular tool will have its own set of data, indicating the complexity required in developing such models !

4.5 Photolithography FMC Conceptual Modeling

The diversity and interdisciplinary nature of modern complex industrial systems such as semiconductor manufacturing almost require some forms of modeling that can provide a reliable mechanism to describe the real system. The model building simply can be defined as the process of setting the logical interrelations of the system in terms of its elements and their attributes, sets, events, activities, and delays.

4.5.1 Block Diagram

The block diagram is one of the most common approaches to represent the flow of the production in a particular process. Many studies use block diagram to provide a simple overview of the whole process under study. Figure 4.6 shows the overview of photolithography FMC.

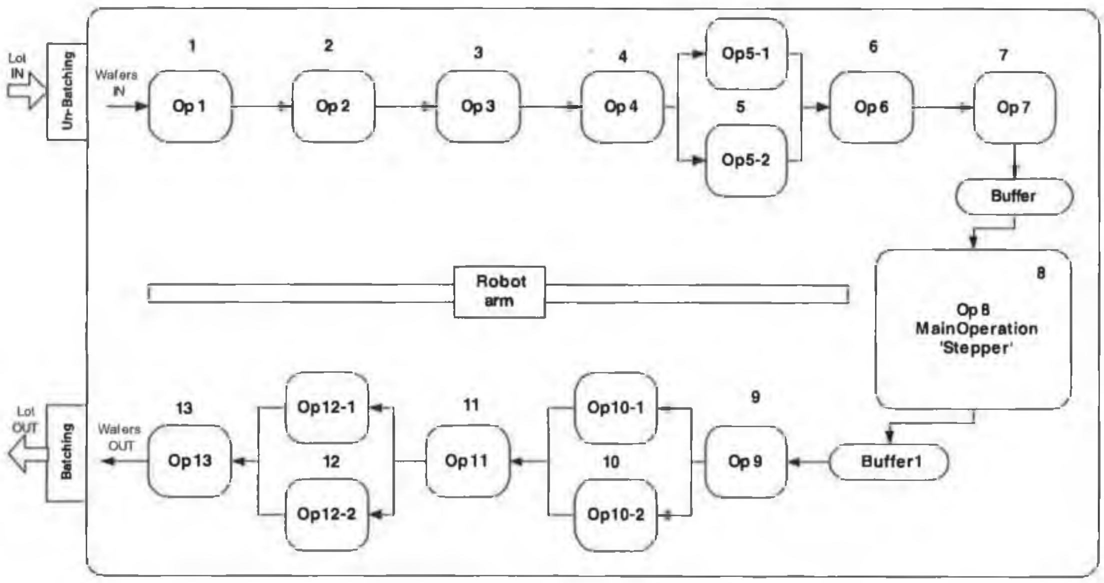


Figure 4.6: Detailed diagram of FMC in Photolithography Area

The upper arrow on the left side expresses the loading of the cell in lots that contain up to 25 wafers. The wafers go through the cell in a flow shop manner and leave the cell again as a lot. The MHS inside the cell includes a robot arm to deliver the wafer to each operating unit and some peripheral equipment to support the handling and the fixturing. The number of operating units varies in every cell based on the technology.

4.5.2 IDEF Model

The IDEF technique considered earlier is a standard tool in modeling. Although IDEF0 models provide a top-bottom (hierarchy) approach [173], the model here varies from top-bottom approach to bottom-top approach. Semiconductor manufacturing is based on multi-layer manufacturing that involves a more complex processing sequence consisting of several layers. The inner layers are fabricated first, and then the fabricated layers are bonded to allow further outer-layer processes. The model here represents photolithography tools that are involved in the fabrication of up to thirteen different layers.

A. Aggregate level of Photolithography Tool model

The top level of the IDEF0 model for Photolithography tools, B1, given in Figure 4.7, indicates the scheduling parameters; the inputs (process planning data and wafer in lots), the control (process characteristic, process factors), the mechanisms (tool, layer, product), and the output (processed wafers). B1 can then be decomposed into the second level of detail, Figure 4.8, which includes the main steps in each photolithography tool, B11 – B15. Each of these blocks is then decomposed into further blocks to describe the operations in the photolithography area in more detail.

B. Detailed level of Photolithography Tool model

Figure 4.9 describes loading operation operation in B11. The wafers arrive to the FMC in lots (boxes contains up to 25 wafers), loaded to the FMC with predetermined batch size. The cell handles the lot and un-batches it into wafers to start the operations.

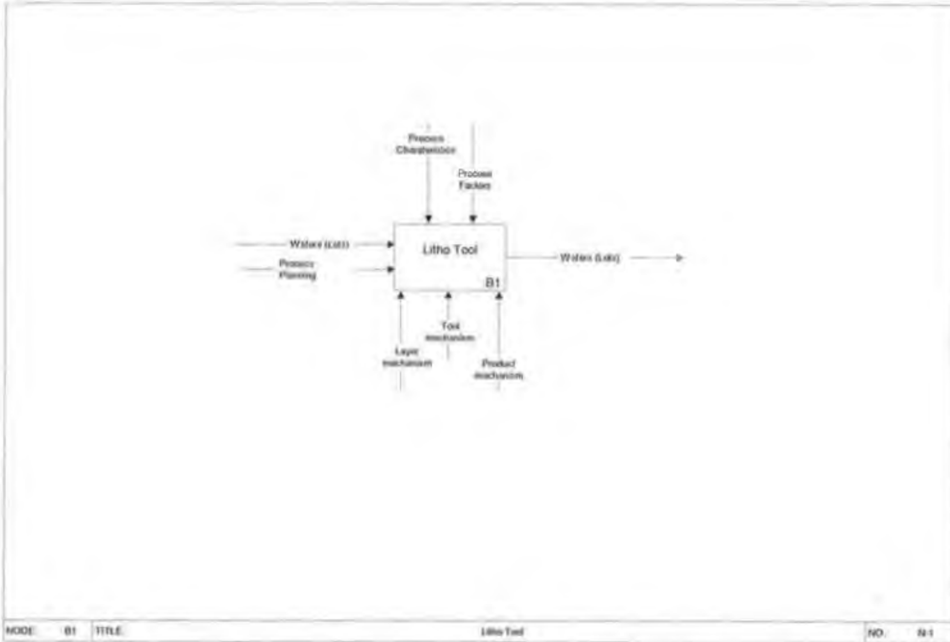


Figure 4.7: Top level of the developed model for Photolithography tools

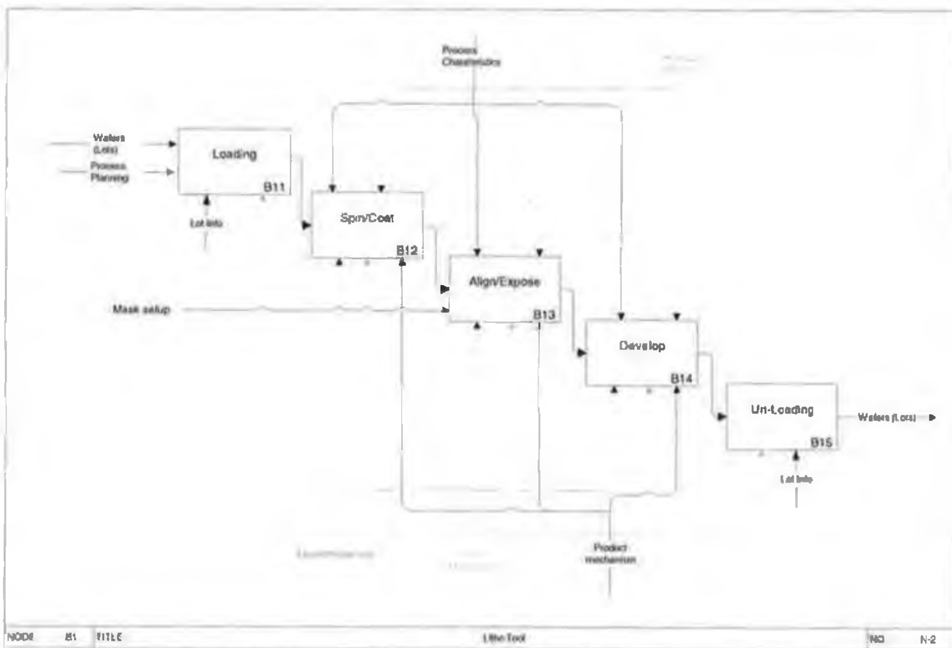


Figure 4.8: Second level of systematic developed model for photolithography tools

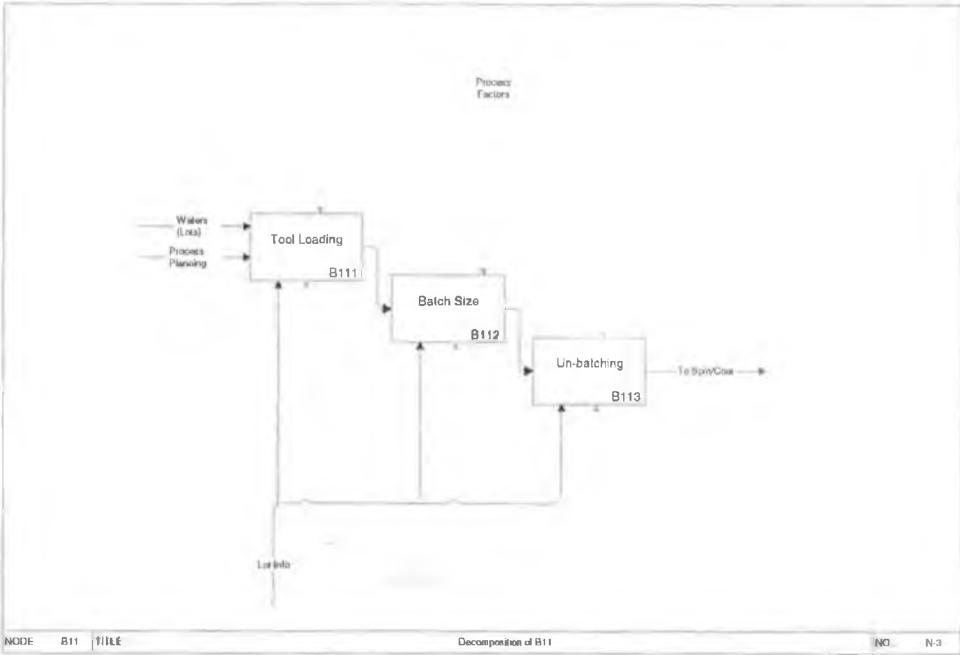


Figure 4.9: Loading operation in detail

C. Main operations in photolithography

Most photolithography processes have a similar process flow with limited variations. The process comprises three sets of operations "Spin/Coat", "Align/Expose", and "Develop". Figures (4.10 - 4.12) illustrate the main operations in the photolithography process in steps.

C.1. "Coat/Spin" Operations

In the "Coat/Spin" set of operations (Figure 4.10), the wafers surface needs to be free of moisture to prevent contamination and process hazards. Operation one is used to drive off moisture from wafers using very high temperatures. The second operation is assurance for the first step, where Hexamethyldisilazane (HMDS) vapor is applied to surface of the wafer to ensure no water is on the surface. This chemical promotes resist adhesion by binding with the silicon surface and presenting an adhesive friendly surface to the resist, thus preventing adhesion of water. The next operation is pre-coat chill where the wafers cool down to same temperature as the resist which is dispensed in the center of wafer and spun over the surface area. In addition to coating, top and bottom chemical Edge Bead Removal (EBR) is performed in this module with the application of a chemical, around the edges of the wafer to clean off the resist. The wafer is then baked gently to remove solvent from the resist coating, before being cooled to ambient temperature in order to transfer to the exposure operation.

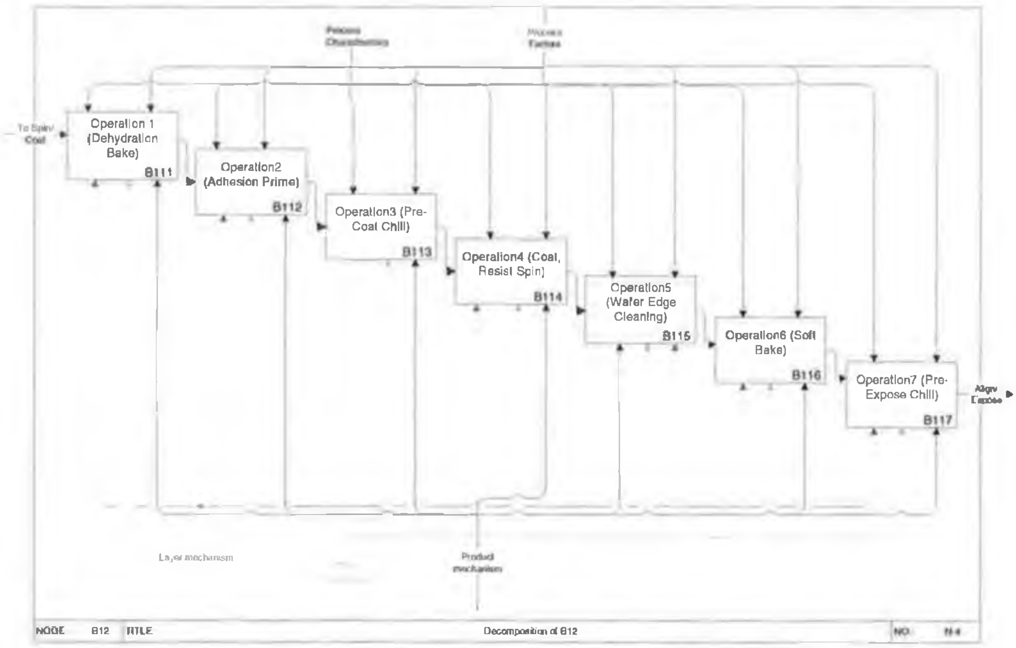


Figure 4.10: IDEF0 of “Coat/Spin” Operations

C.2. “Align/Expose” Operations

The wafer in this set of operations is exposed to ultraviolet (UV) radiation, which transfers the pattern onto the surface of the panel. This operation is the most critical one in photolithography. The sub-operations in this stage are mainly for alignment to ensure that the exposure will happen in the right way. The light from the illumination system goes through the reticle. Each reticle has to go through many adjusting steps during initial loading. For each layer there is a particular pattern held on a reticle must be placed and then aligned in the tool in these stages. The coarse alignment uses a single mark, the intermediate alignment uses four marks around the edge of the exposure field, while fine alignment uses 13 marks and ensures focus as well as position. These operations need only be carried out when a layer change occurs.

Each wafer, however, must also be positioned correctly before exposure. Here again a multi-stage alignement is used with the centre of the wafer and edge notch being located first as a coarse alignment. The wafer is then clamped in the chuck and an intermediate alignment over the full wafer performed before fine alignment to an individual IC device. The wafer is then exposed to UV light passing through the reticle under strict controls to ensure that the sub-micrometer features are transferred within tolerance, Figure 4.11.

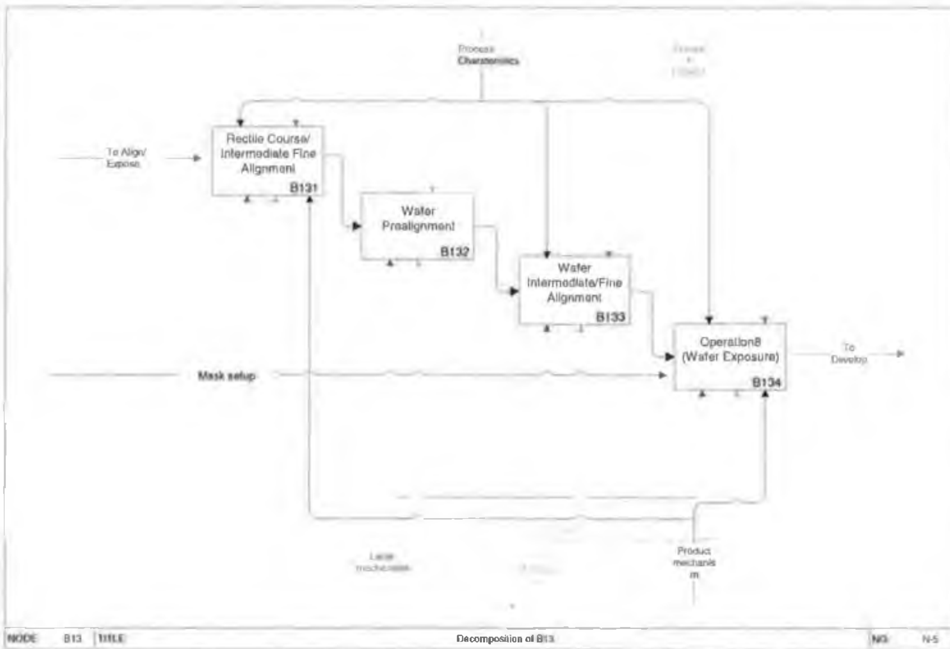


Figure 4.11: IDEF0 of "Align/Expose" Operations

C.3. "Develop" Operations

The wafer then goes through the 'Develop' set of operations starting with wafer edge clearing, where a small-scale exposure tool all around the edge of wafer creates a uniform cleared pattern. The 'Post Exposure Bake' is required to counter a mechanism known as Latent Image Decay. The exposed resist becomes slightly acidic while the unexposed remains slightly basic, thus causing a concentration gradient and diffusion with resulting degradation of the pattern. Application of heat at the 'post exposure bake' halts this process. Often parallel flows (dual ovens) are used here to increase run rates. The wafer must then be cooled to the same temperature as the developer chemical puddles, which dissolves exposed resist for positive resist and unexposed resist for negative resists. Resist wafers are then rinsed and dried by spinning. Finally, wafers then cool down to ambient temperature to prevent melting the lot box and centered to ensure proper wafer handling by robots, Figure 4.12.

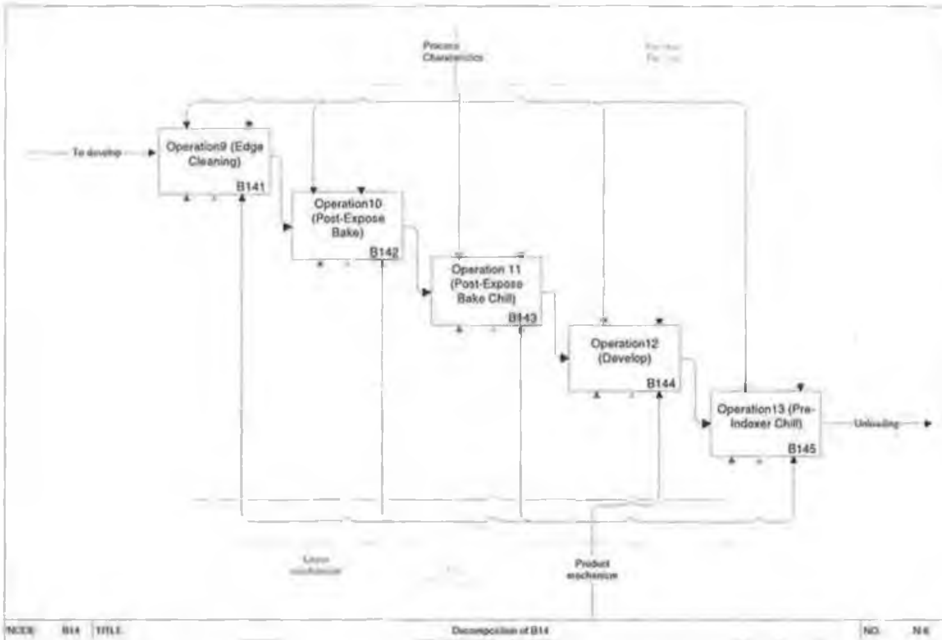


Figure 4.12: IDEF0 of "Develop" Operations

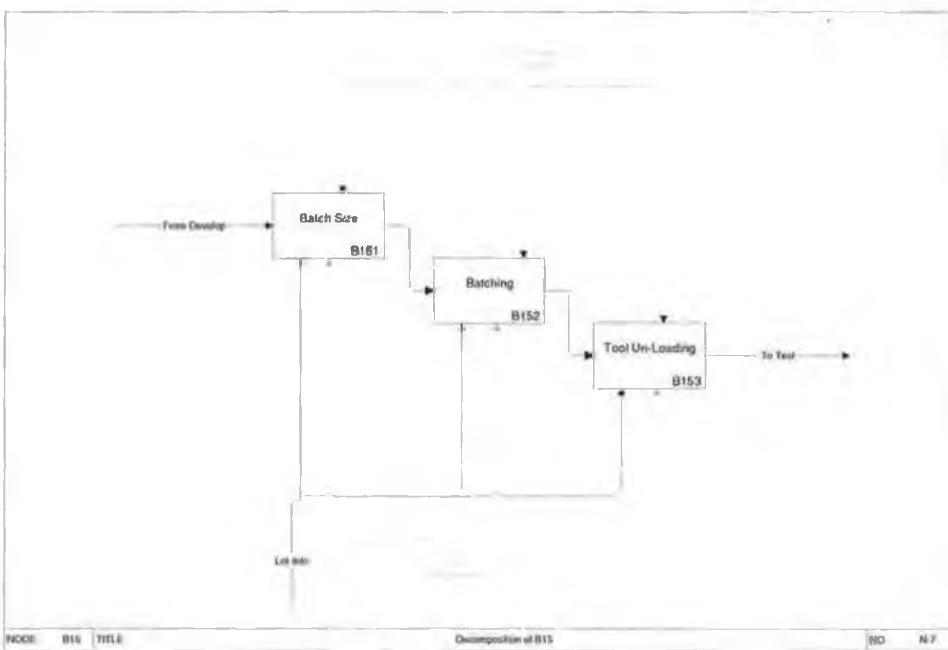


Figure 4.13: Unloading operation in detail

D. Un-Loading Photolithography Tool

Once the wafers' set of operations have been completed, the batch size is determined after the scrap or rework and batching operations will take place in order to unload the lot keeping its integrity, Figure 4.13.

4.5.3 Model Coding

Models for FMCs of the photolithography area were built on this hierarchical structure based using Extend software (ImagineThat, Inc.) [177]. The software supports building dynamic time models, blocks can be inserted from libraries provided with the software or customized. The model decomposes the process into blocks allowing nesting for top-bottom or bottom up modeling approaches. The hierarchical approach of modeling and coding is effective in managing the process complexity, see Figure 4.14.

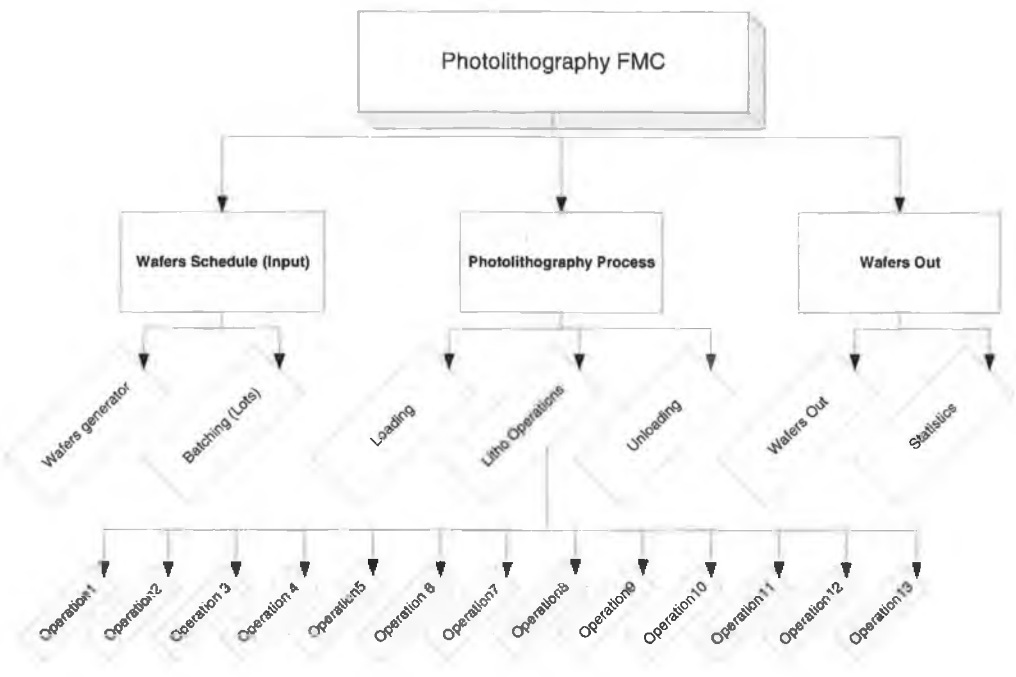


Figure 4.14: Hierarchical structure of FMC modeling

The model is composed of components (called "blocks") with connections between them, each used to represent a portion of the model. Some blocks may simply represent a source of information that is passed on to other blocks.

Other blocks may modify information as it passes through them while other blocks act like hierarchical blocks and contain groups of other blocks. Most blocks have a dialog associated with them to enter values and settings before executing the model. The most common use for a block is to pass information to other blocks or connect outputs to inputs graphically. Different types of connectors transmit different typye of information. The simulation model has successfully converted the real system into a computerized time-based model. The data required has been specified in a generic manner to help the manufacturing team. The model has considered some assumptions as described earlier in order to reduce the complexity of the process. Added to that, the level of detail which the model handles, was agreed with the manufacturing team of the industry partner.

The real manufacturing system may be subjected to random influences, and the simulation model has some randomization and statistical distributions to simulate somehow the actual manufacturing system.

In the photolithography FMC, the three hierarchical blocks in Figure 4.15 represent the flexible manufacturing cell, its input schedule and the output. Each of these blocks contains set of blocks and entities.

The first block on the left side (Wafer generation) simulates wafer arrival times at the beginning of the process. Moreover, it considers batching wafers into lots (default is 25 wafers) taking into account the lot integrity and the order constraints. Once the wafers arrived, this block set the initial production order settings such as product, layer, lot code, learn rate, ..etc.

The main block 'photolithography tool' represents the flexible manufacturing cell in detail, Figure 4.16. The photolithography FMC model simulates the 13 operations that the wafer has to go through in this process. The processing time of each operation is different. The processing time and material handling times as mentioned earlier were considered a normal distribution with a prespecified mean and standard deviation. This model reads the data from spreadsheets linked to the model.

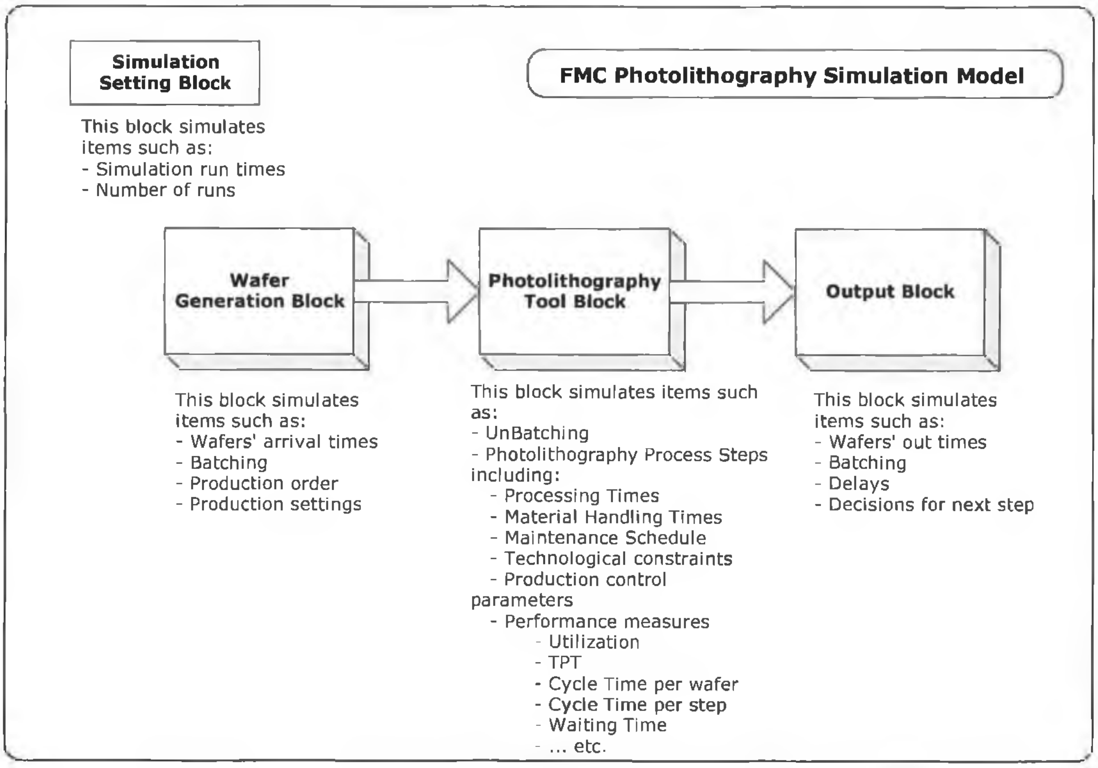


Figure 4.15: The simulation model for FMC of Litho

Once the wafers arrived at this block, the simulation model starts to process the wafer into the operations in the same sequence as shown in Figure 4.16. The setup times were considered in the model in some operations. The model can handle it automatically when the product or the layer changes.

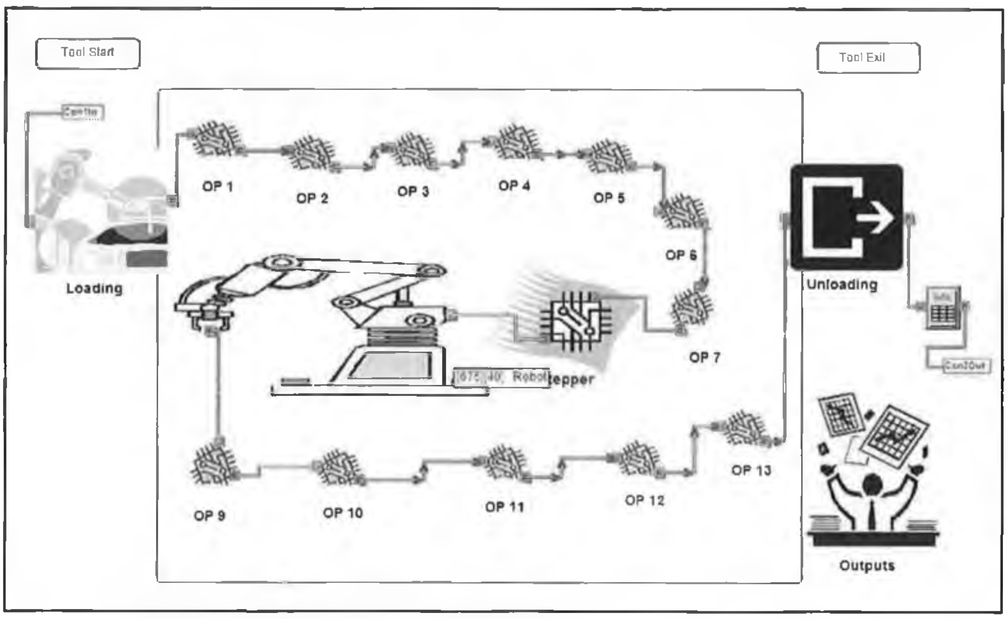


Figure 4.16: Inside 'Photolithography Tool' (main blocks of FMC)

Figures 4.17 – 4.20 use the new modeling approach 'SASM' to clearly illustrate the operations in photolithography process. Some of the operations are described in this section.

Operation one is presented using SASM shown in Figure 4.17. The blocks below represent the first operating unit in the FMC.

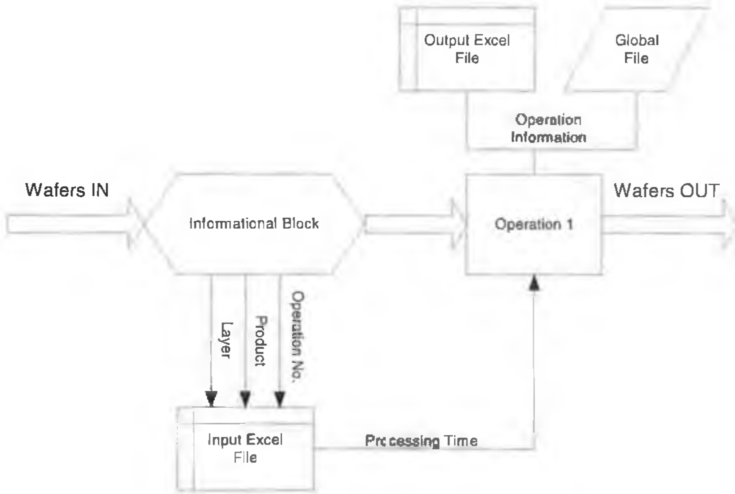


Figure 4.17: Schematic Diagram of Operation 1 (Detailed)

The information block helps the model to identify the nature of the wafers coming to be processed. The link between the model and the spreadsheet was established in order to allow the model to read data from spreadsheet rather than having manual entry of the parameters for each wafer. The machine block reads the processing time based on the wafer parameters such as product, layer, and operation number.

There are some similar operations in modeling, although some operations have a complicated logic, e.g. Figure 4.18. Here the'Operation five' block consists of two identical parallel machines; a wafer has to select one of the machines. The manufacturing team has set some rules to assign a machine for different jobs. The simulation model mimics the real system logic exactly in the selection criteria, and therefore the 'operation five' block looks complicated. There is mathematical formula set to assess the decision making for the machine dedicated to the incoming wafers.

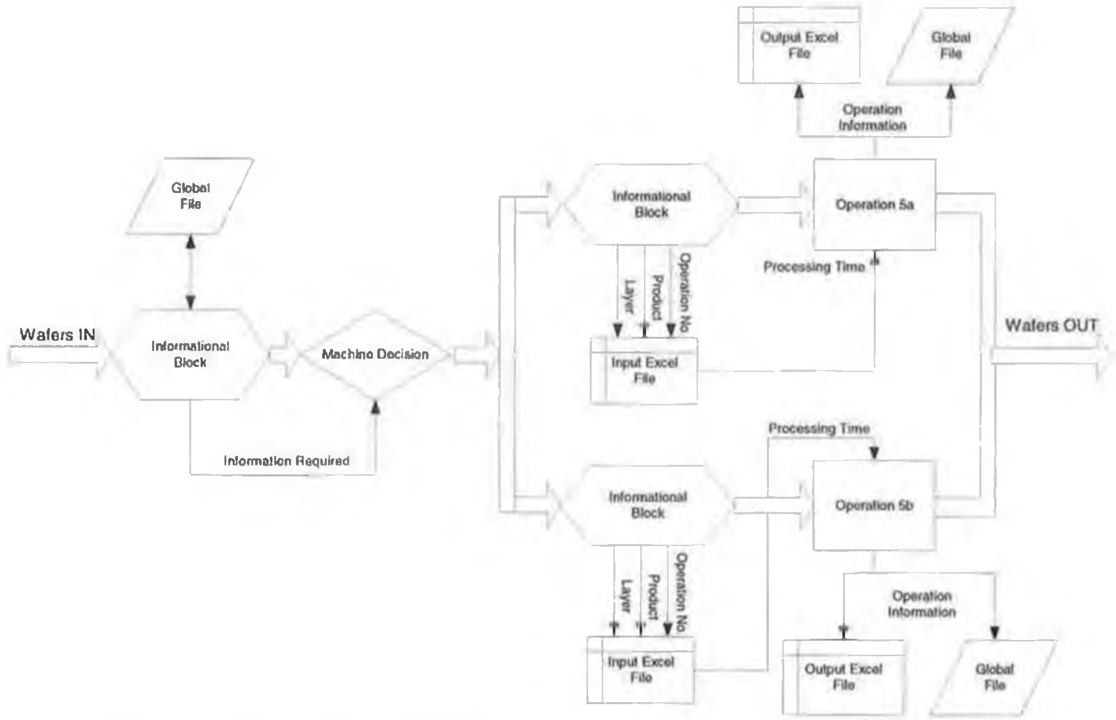


Figure 4.18: Schematic Diagram of Operation 5 (Detailed)

Operation eight 'Stepper', Figure 4.19, is the most important operation in the photolithography process as it is the critical dimensional operation. The stepper is responsible of exposing the wafer to the light in the photolithography process. This operation is setup dependent and is also product/layer sensitive. However, all operations in the cell are layer dependent operations.

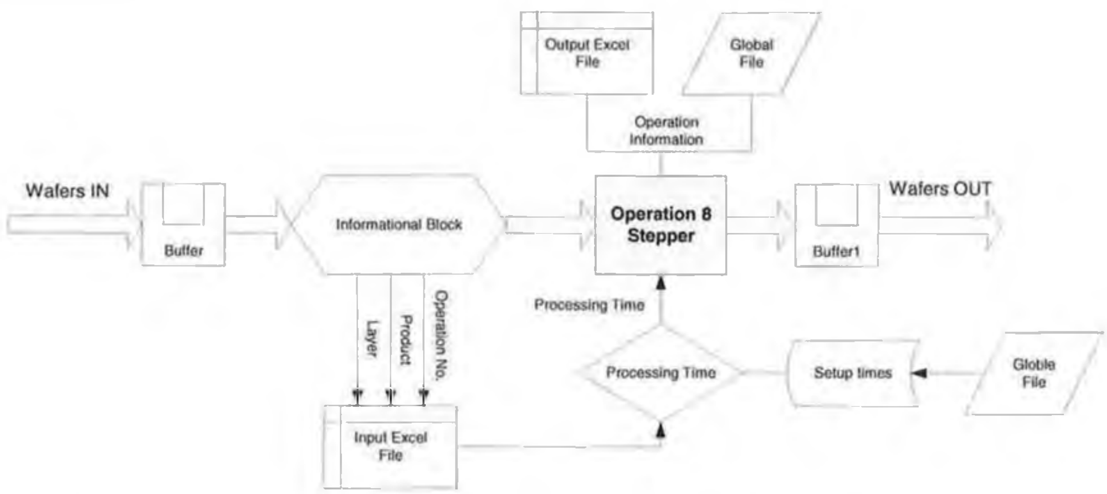


Figure 4.19: Schematic Diagram of Stepper Operation 8 (Detailed)

Some Operations – such as operations ten (Figure 5.18) and twelve – have two identical machines with simple selection logic which allows the first available machine to take the job. The global file was used as a catalyst in selecting the machine, Figure 4.20.

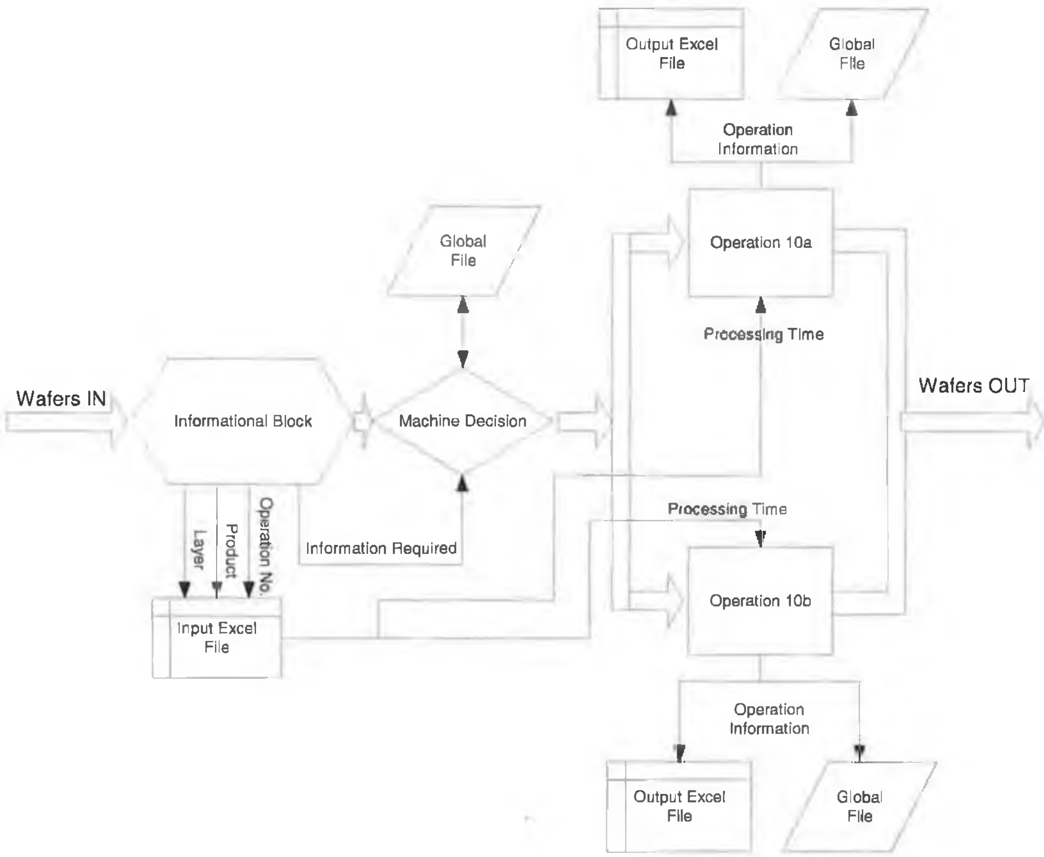


Figure 4.20: Schematic Diagram of Operation 10 (Detailed)

The model has considered all the parameters that the manufacturing team takes into account. The next step was to verify the model before the experimentation phase would take place.

4.6 Model Verification

The strength of decisions made based on the simulation model are a direct function of the validity of the obtained data [178], and hence the need for efficient and objective methods to verify and validate the model. The verification and validation of the model took place as a continuing process [111][175]. In the initial stages, the IDEF model was verified using manufacturing expertise. To ensure that the conceptual model is realistic, historical and clean (industrial expression) data have been examined for verifications sake.

The simulation model has been verified using three approaches. The first approach compares the output of the simulation model with actual data from the manufacturing floor and to other simulation models, although these models cannot provide same capabilities for high wafer starts. The second approach is to check the output through a trace file, which consists of detailed output representing the step-by-step progress of the simulation model over the simulated time. In addition, a decomposition approach has been applied to verify sub-blocks in the model. This approach detects the errors in the model efficiently and makes sure that every block functions as it should. Finally, the third approach is based on reasonableness of the model outputs. This approach relies on experts and manufacturing staff who are the reference to validate the model results based on reasonableness.

- ***First verification approach***

The outputs of the simulation model have been verified with actual data from the manufacturing floor and deterministic models that the industrial partner has used before. The main criteria for verification were the total completion time and cycle time. The preliminary verification was on one product using different number of wafers to be processed. Table 4.2, shows a sample of the verification experiments performed to examine the simulation model outputs.

The Simulation model output shows a comprehensive trend on throughput time criterion, up to 3000 wafers, the results were close to the floor data, although there is a gap between simulation model and reality as can be seen in Figure 4.21. The difference comes from neglecting the preventive maintenance times in the simulation model. The scheduled preventive maintenance is estimated to be 4 hours weekly and the nonscheduled is approximately 2 hours weekly. The gap between simulation results and real data varies from 3% to 4% in wafer starts greater than 3000 wafers per week. It's worth mentioning that the simulation run for 6000 wafer starts required a matter of few minutes on a personal computer (Processor Pentium III) to calculate the solution.

Table 4.2: Comparison between the simulation model output, the deterministic model, and actual data from the shop floor (Time in Seconds).

Wafers Start	Deterministic Model (Exisiting Models)	Simulation Model (New)	Actual Data (Floor Data)
100	11245.2	13221.4	13618.04
200	24739.4	25799.8	26444.8
500	58475.04	63533.5	63660.57
1000	97833.24	126423	128456
1500	146749.9	189313.2	178012.7
3000	293499.7	377983.4	393858.7
5000	430466.3	590123.3	618034.2
5500	516559.5	692433.4	730250.6
6000	513624.5	755323.4	798042.1

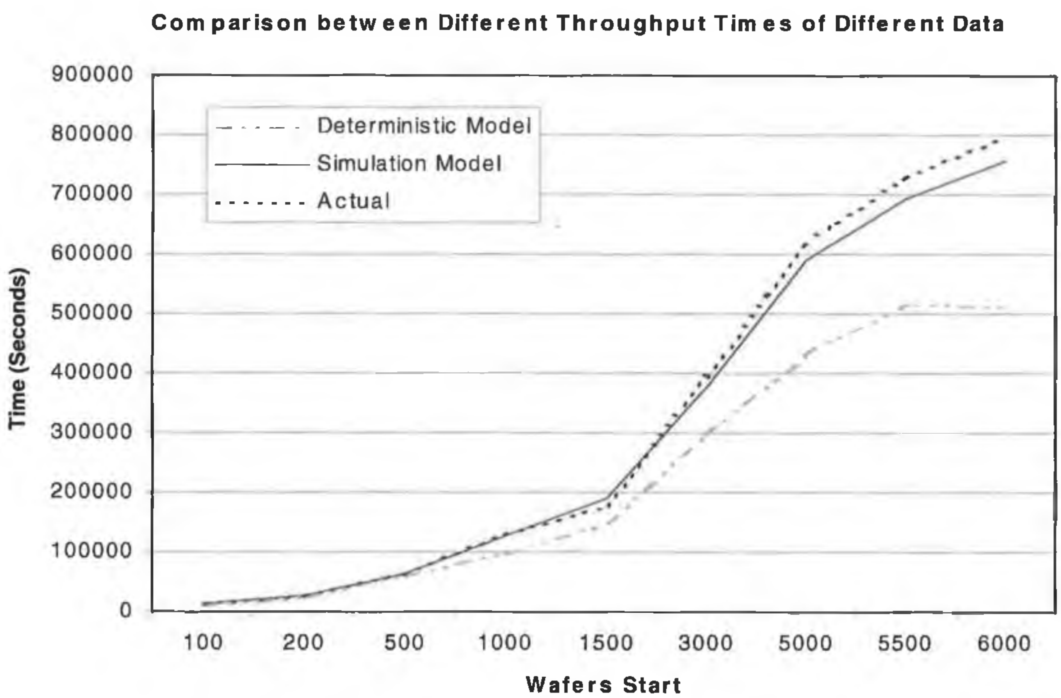


Figure 4.21: Comparison between different models based on throughput Time

The simulation model preliminary results are encouraging, as the model mimics the real manufacturing cell to the extent that the results were very close to the real system and much better than the existing deterministic models.

The simulation model could mimic the real complex process steps in photolithography and the way the manufacturing cell acts. The first verification procedure shows a good overall performance of the model. The second and third verification objectives are to confirm the details of the model.

- ***Second Verification Approach***

The model trace carries some useful information about the process, such as the effect of the use of two identical parallel machines in some operations and the impact of the buffer size on the total completion time. The model was verified throughout by tracing the file and the details and every module and sub-module has been tested separately (decomposition analysis).

- ***Third verification approach***

The third way is to check the output for reasonableness. Similar runs with different parameters were performed. Increasing the wafers at the beginning and changing the processing times of process steps has tested the model sensitivity to changes and verified the output. Reasonable output indicates correct logical and structural data assumptions for the model, and thus verifies the model. The measure for reasonableness on average cycle time per wafer and total throughput times (productivity measures) was checked by the manufacturing team. They reviewed the model outputs and agreed about the error limits.

4.7 Experimentation

There are five main steps in applying design of experiments in a research project and hera are shown below in Figure 4.22.

Logically the first step is the task of experiment planning. The purposes of planning are many, and how it should be carried out appropriately has been described in Roy [179] and Peace [183]. A few things that should be considered are on the right blocks (procedure) in Figure 4.22.

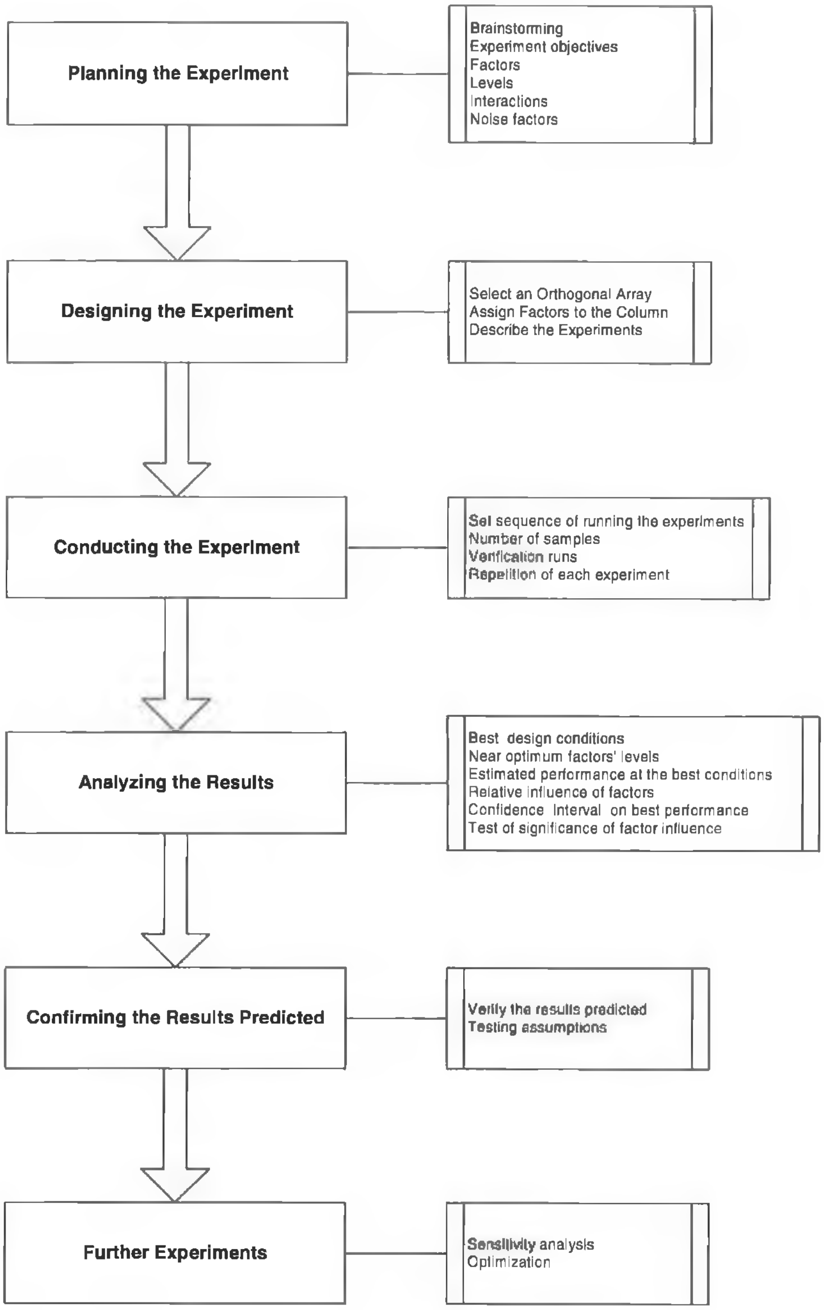


Figure 4.22: Design of Experiment (DOE) Steps

Selecting the orthogonal array, assigning factors to columns, and then describing the experimental combination constitute the experiment design step. "*Understanding of the experimental design technique and planning are necessarily the first and most steps in effective*", Roy mentioned. For the most part, experiments are to be carried out in a manner that best simulates the real-life application environment. The number of samples to test under each experimental condition is one of the key issues in conducting the experiments.

The results collected from the experiments contain information that delineates the primary reasons for the experiment, and more. Then, these results have to be analyzed to obtain key information with statistical validity. Analysis of means (ANOM) and analysis of variance (ANOVA) are two standard tools to analyze the data and determine the significance of each factor as well as the error percentage. Results verification and validation have to be part of the experimental design to confirm the model output. More additional experiments may be recommended for sensitivity analysis or optimization.

The experimental design here has employed two main approaches, Taguchi methodology [180] and full factorial experiments [181]. Taguchi's experimental design framework has been adopted for conducting the main simulation runs. Taguchi's experimental design procedure provides a convenient framework for establishing both the relative factor effects and the significance of the assumed factors [179]. Further, it helps identify suitable factor (level) combinations for finding near-optimal performance measure estimates. Full Factorial Experimentation (FFE) tends to find direct correlation between some assumed parameters. FFE has been applied for more comprehensive sensitivity analysis to study the performance of the cell under some specific criteria such as cycle time per batch (lot), operating unit utilization, WIP and other productivity measures.

4.7.1 Taguchi experimental design framework

The Taguchi experimental design paradigm is based on the technique of matrix experiments [182]. A matrix experiment consists of a set of experiments where the settings of the process parameters under study are changed from one experiment to another. The experimental data generated subsequently is analyzed to determine the effects of the various process parameters. In the statistical literature, matrix experiments are called *designed experiments* and the individual experiments in a matrix experiment are called *treatments*. Settings are also referred to as *levels* and parameters as *factors*.

Experimental matrices essentially are special orthogonal arrays, which allow the simultaneous effect of several process parameters to be studied efficiently. The columns of an orthogonal array are mutually orthogonal; that is, for any pair of columns all combinations of factor levels occur an equal number of times. This, called the *balancing property*, implies orthogonality [180]. The columns of an orthogonal array represent the individual factors under study, and rows represent the experiments to be conducted.

The purpose of conducting orthogonal experiments is twofold:

1) To determine the factor combinations that will optimize a defined objective function (i.e., to determine the optimal level for each factor)
2) To establish the relative significance of individual factors in terms of their effects on the objective function.

Taguchi suggests using a summary statistic, η, called the *signal-to-noise (S/N) ratio*, as the objective function for matrix experiments. Phadke discusses the rationale for using η as the objective function. Taguchi classifies objective functions into one of three categories: the smaller-the-better type, the larger-the-better type; and the nominal-the-best type. *S/N* ratios are measured in decibels (dB).

One important goal in conducting a matrix experiment is to determine optimum/ near optimum factor levels. The optimum/ near optimum level of a factor is that which results in the highest value of η in the experimental region. The effect of a factor level (also called the *main effect*) is defined as the deviation it causes from the overall mean. The overall mean value of η for the experimental region is shown in the formula below;

$$m = \frac{\left[\sum_{i=1}^{n} \eta_i\right]}{n}$$

where;

n = number of experiments performed

i = experiment number.

The process of estimating the main effects of each factor is called *analysis of means*. Taguchi makes a fundamental assumption in the method suggested for determining the optimal/near optimal factor combination (based on the optimal level for each factor) for a defined objective function. He assumed that the variation of η as a function of the factor levels is additive in nature; that is, cross-product terms involving two or more factors are not allowed. The assumption of additivity essentially implies the absence of significant interaction effects between factors. Taguchi suggests that a verification experiment (with factors at their optimum levels) be run to validate the additivity assumption. After running a verification experiment, Phadke [180] points out "if *the predicted and observed η are close to each other, then we may conclude that the additive model is adequate for describing the dependence of η on the various parameters.... On the contrary, if the observation is drastically different from the prediction, then we say the additive model is inadequate.... This is evidence of a strong interaction among the parameters*". In fact, Taguchi considers the absence of interactions to be the primary reason for using orthogonal arrays to conduct matrix experiments.

4.7.1.1 Standard orthogonal arrays

Taguchi has tabulated 18 basic orthogonal arrays, called *standard orthogonal arrays* [183]. To illustrate the notational scheme used for standard orthogonal arrays, consider as an example the $L_{25}(5^6)$ array, which has 25 rows with six 5-level factors. For brevity, the array $L_{25}(5^6)$ will be called the L_{25} array. The number of rows of an orthogonal array represents the number of experiments to be conducted. To be a viable choice, the number of rows must be at least equal the degrees of freedom required for the problem.

The number of columns of an array represents the maximum number of factors that can be studied using that array. Further, to use a standard orthogonal array directly, we must be able to match the number of levels of the factors with the number of levels in the columns in the array. Keeping one or more columns of an array empty does not lose orthogonality of the matrix experiment.

The real benefit in using matrix experiments is the economy they afford in terms of the number of experiments to be conducted. In the present study, because we need to experiment with four factors, each at five levels, a full factorial experiment would have required $5^4 = 625$ experiments. In contrast, it has found the L_{25} orthogonal array to be suitable for our purposes based on the number of factors and levels considered. Therefore, only 25 experiments were needed to run.

4.7.1.2 Experimental Array

To study the impact of the assumed factors within the FMC's considered, standard orthogonal array experiments are used. As mentioned earlier, Taguchi's standard L_{25} orthogonal array (Table 4.3) is found suitable for experimentation purposes.

Table 4.3: Taguchi's standard form of $L_{25}(5^6)$ Orthogonal Array

Standard $L_{25}(5^6)$ orthogonal array					
1	1	1	1	1	1
1	2	2	2	2	2
1	3	3	3	3	3
1	4	4	4	4	4
1	5	5	5	5	5
2	1	2	3	4	5
2	2	3	4	5	1
2	3	4	5	1	2
2	4	5	1	2	3
2	5	1	2	3	4
3	1	3	5	2	4
3	2	4	1	3	5
3	3	5	2	4	1
3	4	1	3	5	2
3	5	2	4	1	3
4	1	4	2	5	3
4	2	5	3	1	4
4	3	1	4	2	5
4	4	2	5	3	1
4	5	3	1	4	2
5	1	5	4	3	2
5	2	1	5	4	3
5	3	2	1	5	4
5	4	3	2	1	5
5	5	4	3	2	1

This enables simultaneous consideration of six factors at five levels. In the present case only four factors are considered, so the first four columns of the L_{25} orthogonal array are used, with the fifth and sixth columns being excluded for experimentation purposes without affecting the orthogonality of the matrix. The factors (parameters) under investigation have been set to five different levels, Table 4.4. The levels of each parameter have been set based on information from the manufacturing team. The Wafer Starts (WS) is the total number of wafers in front of the cell at the beginning of simulation run. Product-mix is the number of products selected randomly. Sequencing of wafers was done using one of the following five rules: First comes First served (FCFS), wafer with shortest total processing times (W-STPT), wafer with longest total processing times (W-LTPT), wafer with least/minimum layer number (W-FLN), and Random selection (Random).
Finally, stepper buffer size is variable and can be assigned to one of the five values 2, 3, 5, 8, or 13 wafers. The resulting matrix experiment table with the factor level details is shown in Table 4.5.
Throughput time is the selected criterion to measure the cell performance; it can be suitably modified into the corresponding *S/N* ratio for incorporation into the matrix experiment. It may be noted that, the real benefit in using *S/N* ratios is for situations where multiple repetitions are performed. For this case, Each experiment constitutes five repetitions. The number of repetitions was selected based on some statistical references such as Roy [179].

The equations below show the conventional notation in the Taguchi methodology of experimentation.

$$\eta_i = -10\, Log_{10}\,(MSD)$$

$$MSD = \frac{\sum_{i=1}^{n} Results^2}{n}$$

MSD = mean square deviation for smaller-the-better,

Results = the output or readings collected.

Table 4.4: Factor-level details used in the matrix experiment

Factor	Factor Level	Factor-level details
Wafer starts(WS)	1	750
	2	1500
	3	2250
	4	3750
	5	6000
Product-mix (PM)	1	1
	2	3
	3	5
	4	10
	5	15
Products Sequence (PS) (Dispatching Rule)	1	FCFS
	2	W-STPT
	3	W-LTPT
	4	W-FLN
	5	Random
Stepper Buffer Size	1	2
	2	3
	3	5
	4	8
	5	13

Table 4.5: Experimental table details

Exp.	WS	PM	PS	BS
1	750	1	FCFS	2
2	750	3	W-STPT	3
3	750	5	W-LTPT	5
4	750	10	W-FLN	8
5	750	15	Random	13
6	1500	1	W-STPT	5
7	1500	3	W-LTPT	8
8	1500	5	W-FLN	13
9	1500	10	Random	2
10	1500	15	FCFS	3
11	2250	1	W-LTPT	13
12	2250	3	W-FLN	2
13	2250	5	Random	3
14	2250	10	FCFS	5
15	2250	15	W-STPT	8
16	3750	1	W-FLN	3
17	3750	3	Random	5
18	3750	5	FCFS	8
19	3750	10	W-STPT	13
20	3750	15	W-LTPT	2
21	6000	1	Random	8
22	6000	3	FCFS	13
23	6000	5	W-STPT	2
24	6000	10	W-LTPT	3
25	6000	15	W-FLN	5

4.7.1.3 Matrix experiment results

Simulation experiments are performed using 'Extend' simulation software package, into which user-written C++ code is linked to capture the customization to be incorporated into the models. All wafers are assumed available at the start of the simulation run (i.e., wafer arrivals are not stochastically generated); although wafer arrivals into the system are dependent on signals coming from the cell or first operating unit. The processing times as well as product mix is assumed to be predefined before the simulation run. Finally, identical experimental testing conditions for each simulation scenario are ensured using the method of common random numbers.

Table 4.6: Matrix experiment simulation results

Experiment #	Avg. Throughput Time *(Seconds)*	Average Throughput per wafer *(Seconds)*	*S/N* Ratio (η_i) (dB)
1	88470	117.96	-41.4347
2	90239.4	120.3192	-41.6067
3	95893.5	127.858	-42.1346
4	96547.6	128.7301	-42.1936
5	99300	132.4	-42.4378
6	175436.1525	116.9574	-41.3606
7	178944.8756	119.2966	-41.5326
8	184207.9601	122.8053	-41.7843
9	189471.0447	126.314	-42.029
10	196733.5	131.1557	-42.3557
11	259775.9795	115.456	-41.2483
12	266464.7785	118.4288	-41.4691
13	269464.7785	119.7621	-41.5664
14	279153.5775	124.0683	-41.8732
15	290141.55	128.9518	-42.2085
16	420896.836	112.2392	-41.0029
17	430534.7727	114.8093	-41.1995
18	451332.1936	120.3553	-41.6093
19	466448.5829	124.3863	-41.8955
20	472018.4	125.8716	-41.9986
21	702054.36	117.0091	-41.3644
22	725018.7088	120.8365	-41.644
23	731500.8832	121.9168	-41.7213
24	761362.514	126.8938	-42.0688
25	770334.5	128.3891	-42.1706

The results obtained from the simulation model based on the matrix experiment are detailed in Table 4.6. The data analysis using the Taguchi experimental framework involves the analysis of means (ANOM) and analysis of variance (ANOVA). ANOM helps to identify the optimal/near optimal factor combinations, whereas ANOVA establishes the relative significance of factors in terms of their contribution to the objective function.

4.7.2 Analysis of means (ANOM)

The main factor effects, calculated using the formulas given in references (e.g. Phadke [180], Roy [179], and Peace [183]) are summarized in Table 4.7.
The notational convention adopted for analysis is

$$m_{jk} = \frac{\left[\sum_{i=1}^{k} \eta_i\right]}{k}$$

$$m = \frac{\left[\sum_{i=1}^{n} \eta_i\right]}{n}$$

m_{jk} = main factor effect for the k^{th} level of factor j , factor j is assigned the following factors: WS, PM, PS, BS.

η_i= observed *S/N* ratio for the i^{th} orthogonal experiment,

n= number of experiments performed,

m= overall mean value of η

Based on the analysis of means, the optimum/near optimum level for each factor resulting from matrix experiment is shown italicized in the throughput column of Table 4.7.

Table 4.7: Factor main effects for matrix experiment simulation study results

Factor-level main effects	*Applicable formula*	*Main effect value TPT per wafer*
m_{WS1}	$(\eta_1+\eta_2+\eta_3+\eta_4+\eta_5)/5$	-41.9615
m_{WS2}	$(\eta_6+\eta_7+\eta_8+\eta_9+\eta_{10})/5$	-41.8124
m_{WS3}	$(\eta_{11}+\eta_{12}+\eta_{13}+\eta_{14}+\eta_{15})/5$	-41.6731
m_{WS4}	$(\eta_{16}+\eta_{17}+\eta_{18}+\eta_{19}+\eta_{20})/5$	***-41.5411***
m_{WS5}	$(\eta_{21}+\eta_{22}+\eta_{23}+\eta_{24}+\eta_{25})/5$	-41.7938
m_{PM1}	$(\eta_1+\eta_6+\eta_{11}+\eta_{16}+\eta_{21})/5$	***-41.2822***
m_{PM2}	$(\eta_2+\eta_7+\eta_{12}+\eta_{17}+\eta_{22})/5$	-41.4904
m_{PM3}	$(\eta_3+\eta_8+\eta_{13}+\eta_{18}+\eta_{23})/5$	-41.7632
m_{PM4}	$(\eta_4+\eta_9+\eta_{14}+\eta_{19}+\eta_{24})/5$	-42.012
m_{PM5}	$(\eta_5+\eta_{10}+\eta_{15}+\eta_{20}+\eta_{25})/5$	-42.2342
m_{PS1}	$(\eta_1+\eta_{10}+\eta_{14}+\eta_{18}+\eta_{22})/5$	-41.7834
m_{PS2}	$(\eta_2+\eta_6+\eta_{15}+\eta_{19}+\eta_{23})/5$	-41.7585
m_{PS3}	$(\eta_3+\eta_7+\eta_{11}+\eta_{20}+\eta_{24})/5$	-41.7966
m_{PS4}	$(\eta_4+\eta_8+\eta_{12}+\eta_{16}+\eta_{25})/5$	***-41.7241***
m_{PS5}	$(\eta_5+\eta_9+\eta_{13}+\eta_{17}+\eta_{21})/5$	-41.7394
m_{BS1}	$(\eta_1+\eta_9+\eta_{12}+\eta_{20}+\eta_{23})/5$	-41.7305
m_{BS2}	$(\eta_2+\eta_{10}+\eta_{13}+\eta_{16}+\eta_{24})/5$	***-41.7201***
m_{BS3}	$(\eta_3+\eta_6+\eta_{14}+\eta_{17}+\eta_{25})/5$	-41.7477
m_{BS4}	$(\eta_4+\eta_7+\eta_{15}+\eta_{18}+\eta_{21})/5$	-41.7817
m_{BS5}	$(\eta_5+\eta_8+\eta_{11}+\eta_{19}+\eta_{22})/5$	-41.802

It may be noted that the main effects values are measured in decibels because they refer to *S/N* ratios. Accordingly, the predicted factor level combination that should optimize (i.e., minimize) the throughput time per wafer is WS4, PM1, PS4, BS2, which easily is interpreted to mean the wafer starts = 3750 wafers, the product-mix is one product, the product sequence is wafer with the least layer number first, and the stepper buffer size = 3. Interestingly, the predicted best setting corresponding to row number 16 in the matrix experiment.

Figure 4.23 plots the main effects of each factor level. The near optimum for each factor can be easily identified as the level that results in the highest value of η in the factor-level range (η is negative, and hence the lowest points are the best). Note that the prediction of the near optimum factor level combination is conditioned by the variation of η as a function of the factor level, satisfying the additivity assumption.

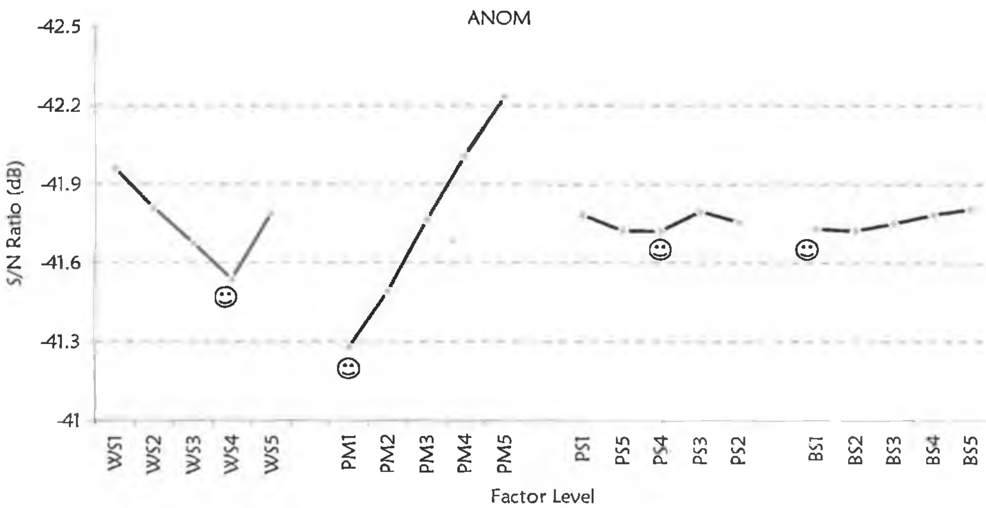


Figure 4.23: Analysis of means (ANOM) plot of factor main effects

To justify the validity of this assumption, we need to carry out a verification experiment with the near optimum setting obtained.

The ANOM plot shown in Figure 4.24 reveals the relative magnitude of effects by factors on the throughput time per wafer. The product-mix is seen to affect the throughput time the most, followed by wafers start. The effect of both of the control rules is seen to be relatively less pronounced based on the selected criterion (throughput time). However, a better feel for the relative effects is obtained by conducting an analysis of variance (ANOVA) test, which is also needed for estimating the error variance for the factor effects and the variance validity of the prediction error, to provide the necessary input for justifying the additivity assumption.

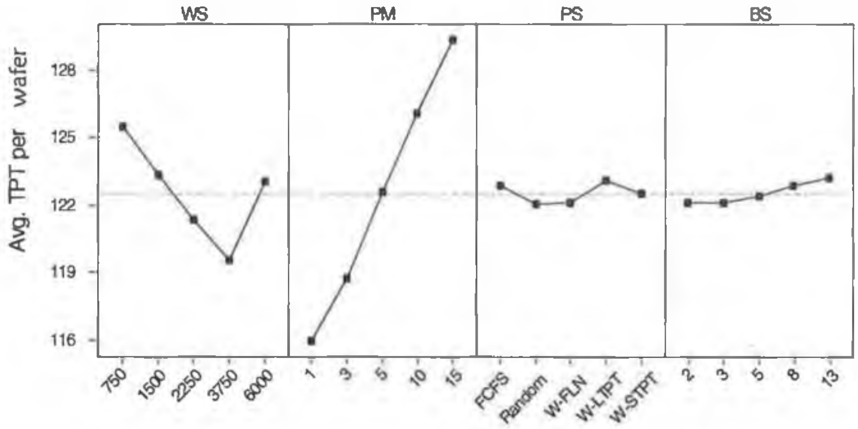


Figure 4.24: ANOM plot of factor main effects on TPT per wafer

4.7.3 Analysis of Variance (ANOVA)

The main formulas that have been used in conducting the ANOVA test are as follows:

$$SST = SSB + SSE$$

where,

SST = Total sum of squares,

SSB = Sum of the sums of squares due to various factors,

SSE = Sum of squares due to error.

Also,

$$SST = GTSS - SSM$$

where,

$GTSS$ = Grand total sum of squares and can be calculated as $GTSS = \sum_{i=1}^{n} \eta_i^2$,

SSM = Sum of squares due to the mean and can be calculated as $SSM = n \times m^2$.

Now,

$$GTSS = 43593.49(\mathrm{dB})^2$$
$$\mathrm{SSM} = 25\times(-41.7564)^2 = 43589.92(\mathrm{dB})^2$$

Therefore,

$$SST = GTSS - SSM$$
$$SST = 43593.49 - 43589.92 = 3.57(dB)^2$$

$$SSB = \sum_{j=1}^{c}\left[l_j \sum_{k=1}^{l_j}(m_{jk} - m)^2 \right]$$

where,

c = number of factors,

l_j = number of levels for factors j.

or

$$SSB = SSB_1 + SSB_2 + SSB_3 + \ldots.. + SSB_c$$

In the case under study,

$$SSB = SSB_{WS} + SSB_{PM} + SSB_{PS} + SSB_{BS}$$

where for example,

$$SSB_{WS} = 5\left[(m_{ws1} - m)^2 + (m_{ws2} - m)^2 + \ldots + (m_{ws5} - m)^2\right]$$

Now,

$SSB_{WS} = 0.499298\,(\mathrm{dB})^2$, $SSB_{PM} = 2.946854\,(\mathrm{dB})^2$,

$SSB_{PS} = 0.023777\,(\mathrm{dB})^2$, $SSB_{BS} = 0.023888\,(\mathrm{dB})^2$

So,

$$SSB = 3.493817\ (\mathrm{dB})^2$$

$$SSE = SST - SSB$$
$$SSE = 3.57 - 3.493817 = 0.076183(dB)^2$$

The error variance (σ_e^2), defined as

$$\sigma_e^2 = \frac{SSE}{ErrorDOF}$$

$$\sigma_e^2 = \frac{0.123848}{16} = 0.00774(dB)^2$$

Table 4.8: ANOVA using the simulated results estimated as S/N ratios

Factor	Degree of Freedom (DOF)	Sum of squares (SSB)	Mean square (SSB/DOF)	F
WS	4	0.499298	0.124825	16.126
PM	4	2.946854	0.736714	95.1765
PS	4	0.023777*	0.005944	
BS	4	0.023888*	0.005972	
Error	8	0.076183*	0.009523	
Total	24	3.57		
(Error)	(16)	(0.123848)	(0.0077405)	

* Indicates the sum of squares added together to estimate the pooled error sum of squares, indicated by parentheses. The F ratio is calculated using the pooled error mean square.

Phadke [180] suggests using F ratio resulting from ANOVA only to establish the relative magnitude of the effect of each factor on the objective function and to estimate the error variance. However, probability statements regarding the significance of the individual factors are not made. From the ANOVA table, Table 4.8, the relative effects of the factors product-mix and the number of wafers are seen to be important, followed by product sequence and stepper buffer size. This is in agreement with the ANOM results.

Table 4.9: ANOVA using the original simulation results

Factor	Degree of Freedom (DOF)	Sum of squares (SSB)	Mean square (SSB/DOF)	F
WS	4	98.97987	24.74497	15.466
PM	4	585.837	146.4593	91.537
PS	4	4.040589*	1.010147	
BS	4	4.701041*	1.17526	
Error	8	16.84222*	2.105278	
Total	24	710.4007		
(Error)	(16)	(25.58385)	(1.6)	

The statistical significance of the impact of individual factors on the throughput time per wafer is highlighted in Table 4.9 using the original simulated results (i.e., without converting to S/N ratios). The resulting F ratios (F-calculated) confirms the fact that the product-mix and the number of wafer starts are significant, at 95 % confidence level, Tablulated F0.5, 4, 16 is 3.01.

4.7.4 Testing of additivity

To validate the assumption of additivity, a verification experiment needs to be conducted with the optimal/near optimal factor settings. The result of the verification experiment then is compared with a predicted optimal/near optimal value, resulting in a prediction error. If the prediction error happens to fall within a two-standard-deviation confidence limit of the variance of prediction error, the additivity assumption can be assumed justified [179]. Validation of the additivity assumption essentially implies the absence of significant interaction effects between factors.

- ***Verification experiment***

A verification experiment was performed with the optimal/near optimal factor combination (WS4, PM1, PS4, BS2). The observed optimal/near optimal throughput time per wafer was 112.2392 seconds; that is, $\eta_{obs.opt}$ = -41.0029 dB. The following equation was then used to predict the optimum/near optimum performance measure value:

$$\eta_{pre.opt} = m + (m_{WS4} - m) + (m_{PM1} - m)$$
$$= -41.7564+(-41.5411+41.7564)+(-41.2822+41.7564)$$
$$= -41.0669 \text{ dB}$$

Note that $\eta_{pre.opt}$ is calculated using only significant optimal (main effects) factor-level values. The prediction error then becomes

$$\textit{Prediction error} = \eta_{obs.opt} - \eta_{pre.opt}$$
$$= -41.0029-(-41.0669)$$
$$= 0.064 \text{ dB}$$

The variance of prediction error σ_e^2 is calculated as:

$$\sigma_{e\ pred}^2 = \left(\frac{1}{n_0}\right)\sigma_e^2 + \left(\frac{1}{n_r}\right)\sigma_e^2$$

where,

n_0 = equivalent sample size for the estimation of $\eta_{pre.opt}$.

n_r = number of repetitions of the verification experiment.

In the present case, n_0 is given by

$$\left(\frac{1}{n_0}\right)=\left(\frac{1}{n}\right)+\left(\frac{1}{n_{WS}}-\frac{1}{n}\right)+\left(\frac{1}{n_{PM}}-\frac{1}{n}\right)=\left(\frac{1}{25}\right)+\left(\frac{1}{5}-\frac{1}{25}\right)+\left(\frac{1}{5}-\frac{1}{25}\right)=\frac{9}{25}$$

where,

n = number of rows in the matrix experiment = 25,

n_{WS} = number of times factor WS was repeated in the matrix experiment = 5,

n_{PM} = number of times factor PM was repeated in the matrix experiment = 5.

Further, because in this study n_r = 5, the variance of the prediction error is calculated to be

$$\sigma^2_{e\ pred}=\left(\frac{1}{n_0}\right)\sigma^2_{e\ TPTperWafer}+\left(\frac{1}{n_r}\right)\sigma^2_{e\ TPTperWafer}$$

$$\sigma^2_{e\ pred}=\left(\frac{9}{25}\right)\times 0.0077405+\left(\frac{1}{5}\right)\times 0.0077407$$

$$\sigma^2_{e\ pred}=0.004335(dB)^2$$

The corresponding two-standard-deviation confidence limits for the prediction error are $\pm 2\times\sqrt{\sigma^2_{e\ pred}}=\pm 0.132(dB)$.

The prediction error of 0.064 dB is well within the calculated confidence limits, so the additivity assumption is justified.

4.7.5 Results Analysis

The Taguchi experimental design paradigm has been used to gain a better understanding of the significance of each parameter on the system performance. Based on the ANOVA detailed in Tables 4.8 and 4.9, the main control parameters (i.e., the number of wafer starts and product-mix) that have a statistically significant impact on the throughput time are determined. In contrast, the parameters such as product sequence and stepper buffer size are not seen to be statistically significant. The results suggest that experimentation should focus attention on the alternatives available for the product-mix and wafer starts and only then the other parameters for improving the global performance. The results are noteworthy, because manufacturing teams have typically been tempted to experiment with different production plans. The use of the Taguchi experimental design procedure provides an efficient platform for quickly focusing in on the parameters that need to be given priority.

The ANOM plots in Figure 4.23 and 4.24 provide two useful insights with regard to wafers start: an increase in the WS from 1 to 3 reduces the TPT per wafer as the utilization of the machines were getting higher, until it reaches the best TPT per wafer at WS4 (3750). After that, for 6000 wafer starts the performance starts to drop as a result of wafer blocking in the cell (increasing waiting times). Moreover, the ANOM plot confirms our intuition with respect to the impact of increasing the product-mix on cell performance and in particular TPT per wafer.

It is important to say that the compromise between product-mix and wafer starts poses a more challenging problem. For the experimental framework, attention is drawn to the planning conditions in increasing/decreasing product-mix and the impact of it on the FMC performance. Other issues such as a sensitivity analysis of batch size of each product on total completion time need to be investigated.

4.8 Sensitivity Analysis

The complexity of FMC planning in the photolithography area has encouraged a more in depth study of the effect of increasing the number of products in the FMC. Further experiments have been conducted to focus on the impact of increasing product-mix on cell performance. The Taguchi methodology has efficiently provided an insight into the impact of product-mix on the TPT per wafer criterion. All the experiments have employed the same operating conditions and five repetitions for every single experiment were performed.

A number of simulation sensitivity analyses were performed. These included experiments to analyze:

- The impact of changing the quantity of product batch size on wafer cycle time
- The impact of increasing product-mix on loading of the FMC
- Comparison of cycle times through each of the photolithography steps in order to determine the bottlenecks in the photolithography FMC.

a) <u>Impact of product batch size on wafer cycle time</u>

The number of lots of each product in a batch can be adjusted to reduce wafer cycle time. However, the manufacturing team do not have any confident methodology to find the optimum size. The simulation model was used to determine the effect of changing the number of each product in the product-mix on wafer cycle time (CT/wafer). In this case, wafers were introduced to the system according to their product type. For the five products selected for simulation, equal numbers of each type were used with all of product 1 being introduced before the first lot of product 2,..etc. Each product requires processing of the same layer so that the main impact on set up times are the reticle changes associated with a product change. The results of 20 experiments on a sample of two different layers, with five different products, are shown in Figure 4.25.

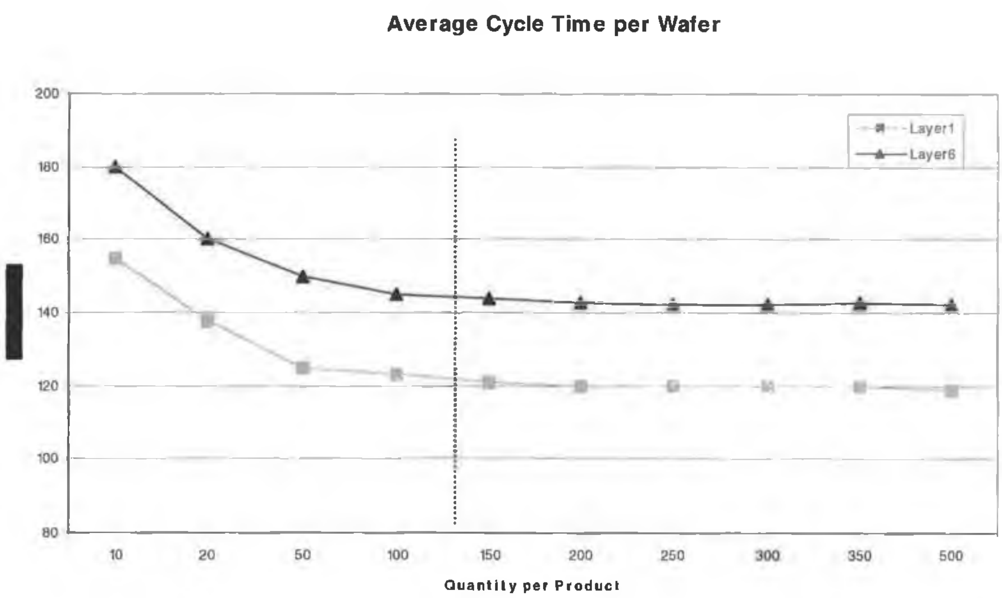


Figure 4.25: Impact of product batch size on average CT/wafer

It has been concluded that the minimum start volumes permitted on any given product should be 150 wafers to allow the cell performance to reach an acceptable level.

b) Variation in product batch size (cascade) due to product-mix increase

The objective here was to examine the impact of increasing product-mix and wafer starts on the number of batch sizes to meet planned production. A set of 36 experiments was performed in order to gain insight on the effect of increasing the product-mix on batch size, detailed in Table 4.10. Figure 4.26, shows that for more than 10 products in the cell, the minimum batch size to meet production demands had a consistent value near two. These results are have many significant for the manufacturing team, as the batch sizes currently are four lots based on their deterministic models and verified by previous sensitivity analysis results.

Table 4.10: Matrix experiment

Exp.	PM	WS	Exp.	PM	WS	Exp.	PM	WS
1	1	1000	13	10	1000	25	20	1000
2	1	2000	14	10	2000	26	20	2000
3	1	3000	15	10	3000	27	20	3000
4	1	4000	16	10	4000	28	20	4000
5	1	5000	17	10	5000	29	20	5000
6	1	6000	18	10	6000	30	20	6000
7	5	1000	19	15	1000	31	25	1000
8	5	2000	20	15	2000	32	25	2000
9	5	3000	21	15	3000	33	25	3000
10	5	4000	22	15	4000	34	25	4000
11	5	5000	23	15	5000	35	25	5000
12	5	6000	24	15	6000	36	25	6000

It is clear that, as the product-mix requirements are increased, the performance of the cell is decreased due to the increase in setup times. The cell takes about 300 seconds to change reticle for every layer or product change. The analysis indicated that average batch size tends to two lots in each batch. However, in order to achieve four or five lots with current FMC and operating conditions, number of product-mix *should not exceed four or five at most*. For illustration sake, one layer has been examined in this case to show the effect of changing product-mix ratio.

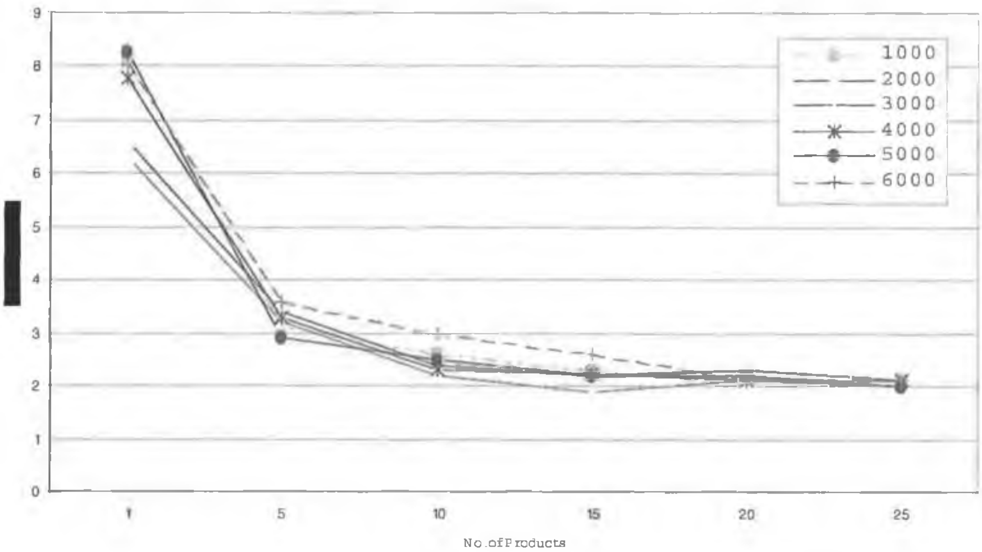


Figure 4.26: variation in product batch size due to increase in product-mix

c) Comparison of Cycle times for each photolithography step

Experiments were designed to analyze the cycle time through each of the photolithography steps. The primary objective here was to identify those steps contributing to high variances in FMC throughput times. The experiments have been carried out for many combinations of wafer starts, layer-mix, product-mix, and product sequences as shown in Table 4.11. The results of simulation, Figure 4.27, were used to detect the bottleneck steps in the cell and hence more alternatives can be proposed to enhance the cell performance.

The sensitivity analysis of cycle times through each step in the photolithography FMC shows the bottleneck steps in the cell.

Table 4.11: Experiments setting matrix

Experiment No	No. of Products	No. of Layers	Total wafers start
1	5	1	6000
2	1	1	6000
3	15	1	6000
4	10	1	6000
5	5	2	6000
6	5	5	6000
7	5	10	6000
8	5	13	5200
9	3	5	2500

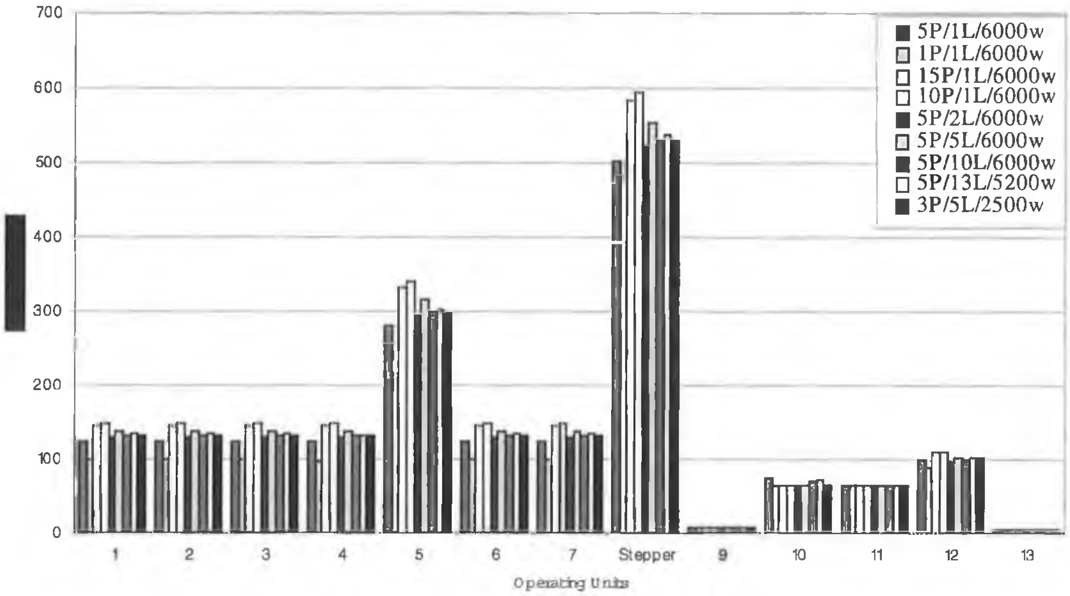


Figure 4.27: Comparison of average cycle times for each step in a photolithography FMC

Figure 4.27 shows that the stepper and step number five may be classified as the critical steps (bottlenecks) in the cell. The simulation output shows the variation of stepper utilization across all the steps is significantly high as the stepper is sensitive to any changes in layer or product. In addition, minor variations can be seen within each step because of applying different scenarios. This might be worth another sensitivity study.

4.9 Conclusion

Integration of simulation and Taguchi experimental design paradigm provides effective, quick, and efficient results. Moreover, sensitivity analysis gives a better understanding of the photolithography flexible manufacturing cell. Analysis of the performance of an existing or planned FMCs has usually been achieved by means of deterministic spreadsheets. Unfortunately, such deterministic models ignore such critical real world phenomena as system variability.

The quality of a technique's solution is measured generally in at least two dimensions: (1) how close the solution comes to the optimal solution if it can

be measured; and (2) how much computer time is required to solve problems of a given size.
The simulation model has shown reasonable results and gives better understanding of the cell behavior under various operating conditions. The quality of the output has been verified with actual floor data under similar conditions. The computer time required to run the simulation model for one experiment was so economic (less than five minutes).

A number of conclusions can be made based on the simulation results and the analysis.

1) There are many factors that affect the performance of the photolithography FMC performance including product-mix, product ratios, product sequence, wafers start, buffer size, and layers.
2) The Taguchi methodology for experimental analysis shows that the product-mix is the most significant factor in all the controllable parameters, followed by the wafers start.
3) Increasing product-mix significantly increases the wafer throughput time and therefore diminishes the cell performance.
4) Since product-mix cannot be set to one, it is suggested to keep the product-mix as low as possible.
5) The wafer starts show a high impact on wafer throughput time. It reaches an optimum at 3750 wafers.
6) The stepper buffer size and the sequence of products did not have a significant effect on wafer throughput time in comparison to other parameters. Nevertheless, the stepper buffer size and in particular the product sequence may have a significant effect on other performance measures such as machine utilization and WIP. This will be explored in chapter six.
7) The Taguchi methodology has provided useful insights with regard to near optimum combination of the selected parameters.
8) The higher the number of each product, the better the wafer cycle time per wafer. The sensitivity analysis shows that the minimum start

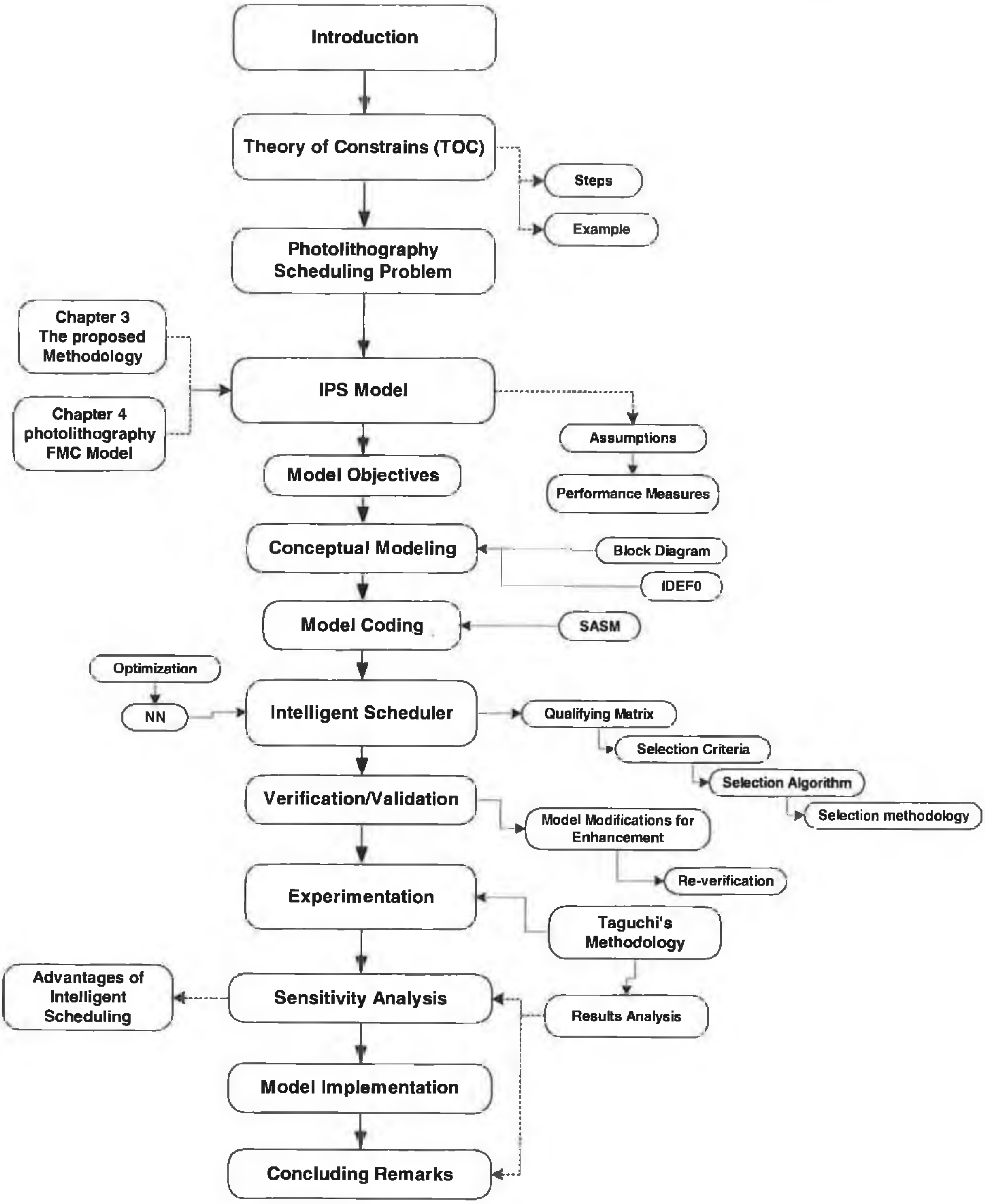
volumes permitted on any given product should be at least *150 wafers* (6 lots) to allow the cell performance to reach an acceptable level.

9) The product-mix has been shown to be the critical factor for normalizing batch size, as the increase of product-mix (more than 10 products) can lead to very low batch size (two or less) which is not acceptable for the production planning team.

10) The bottlenecks of the FMC have been identified to be the stepper, followed by step number five as they have relatively high processing times. In addition, the stepper requires significant setup times with layer and product changes.

Chapter 5

Intelligent Scheduling of Photolithography in Semiconductor Manufacturing



Chapter 5

Intelligent Scheduling of Photolithography in Semiconductor Manufacturing

5.1 Introduction

The dynamic nature of semiconductor scheduling adds more complexity in finding solutions. Traditional industrial engineering analyses and techniques using mathematical/ deterministic models are simply not powerful enough to analyze these complex manufacturing areas [184]. As an alternative, a simulation-based technique can help towards providing an effective tool for scheduling. However, simulation only provides a prediction based on input scenarios and does not inherently optimize the inputs. Further the simulation sometimes for complex models can have a significant computation time. It has also been proved inadequate in modeling complex systems which include some element of human decision-making [186].

Artificial Intelligence turns out to be one of the most effective approaches to handle problems with dynamic natures [150][185]. Moreover, incorporating it within the simulation environment enables the development of intelligent systems to evaluate complex systems such as semiconductor manufacturing. Neural networks have shown good promise for solving combinatorial optimization and constraint satisfaction problems [187]. The intelligent-agent based approach is also capable of providing assistance in decision making [149].

This chapter presents an intelligent model for scheduling lots through the photolithography toolset (IPS Model) including a problem oriented interface, which allows the user to define the parameters of the model, execute the simulation run, collect the results and perform the analysis to set the appropriate (near optimum) lots schedule before real production takes place. The model has linked intelligent-agent based scheduling and a neural network to the simulation model to optimize the lot selection criteria and hence enhance the quality of results. The development of this hybrid model employs the methodology proposed in Chapter 3. This chapter uses the model in chapter five and expands it to include the whole group photolithography toolset. The theory of constraints is the framework of semiconductor management and hence a brief description of the theory is presented first. An overview of the photolithography scheduling problem in semiconductor manufacturing along with quick definitions of the process flow is also discussed. Finally, the results from the model are analyzed and their use as the basis for modification/enhancement of the performance of the toolset are shown.

5.2 Theory Of Constraints (TOC)

The Theory of Constraints (or TOC as it is called) is a relatively recent development in the practical aspect of making organizational decisions in situations in which constraints exist. A constraint is anything in an organization that limits it from moving toward or achieving its goal. There are two basic types of constraints: physical constraints and non-physical constraints. A physical constraint is something like the physical capacity of a machine. A non-physical constraint might be something like demand for a product or a corporate procedure.

The theory Of Constraints was first described by Dr. Goldratt in his novel, *The Goal* [188]. TOC emphasizes a systematic management approach to discovering the uncertain factors hindering development and suggests the global deployment of resources. The Theory of Constraints has been used at three different levels:

Production Management - TOC was initially applied here to solve problems of bottlenecks, scheduling, and inventory reduction.

Throughput Analysis - Application of TOC has caused a shift from cost-based decision making to decision-making based on continuous improvement of processes in which system throughput, system constraints, and statistically determined protective capacities at critical points are key elements.

Theory of Constraints Logical Processes - This third level is the general application of TOC reasoning to attack a variety of process problems within organizations. TOC logic is applied to identify which factors are limiting an organization from achieving its goals, developing a solution to the problem, and then getting the individuals in the process to implement the requisite changes for themselves.

The steps in applying TOC are as follows:

1. *Identify the system's constraints.* Of necessity this includes prioritization so that the ones that most limit system progress toward the goal are clearly identified. In the case under study, it is used to find the FMC with lowest production capacity.
2. *Decide how to exploit the system's constraints.* Once it has been decided how to manage the constraints within the system, keep the constraint operational 100% of its available time at all costs and always operate it at full production rate.
3. *Subordinate everything else to the above decision in Step 2.* Here it is the step to manage the non-constraints resources so that they just provide what is needed to match the output of the constrained resources. These resources should never supply more output than needed because doing so moves the system no closer to the goal. Priortize material closer to completion for re-entrant tools, e.g. photolithography tools. This will generate throughput in the shortest possible time. There is a method called 'Drum-Buffer-Rope', or DBR, that allows for subordination. It will be explained later using an example.
4. *Elevate the system's constraints.* If we continue to work toward breaking a constraint (also called elevating a constraint) at some point the constraint will no longer be a constraint. The constraints are not acts of God. The constraints can be broken.
5. *If the constraint is broken, beware of inertia, return to Step 1.* When that happens, there will be another constraint, somewhere else in the system that is limiting progress to the goal. There is always a weakest link in any chain.

Over the past several years, many manufacturing systems have successfully employed the concepts of Goldratt's Theory Of Constraints (TOC) to improve the performance of their capacity management.

Example: TOC in Manufacturing

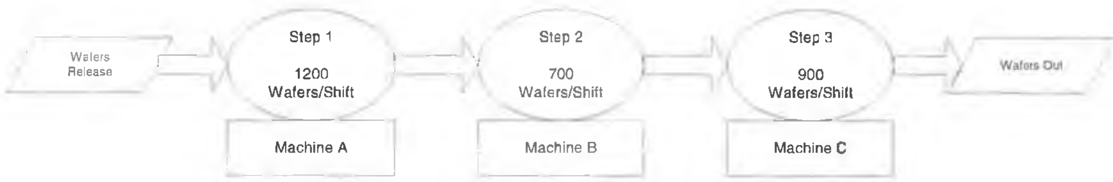


Figure 5.1: a simple factory

Consider the simple semiconductor factory shown in Figure 5.1. There are three processing steps, each with a flexible manufacturing cell, and an average run rate in wafers per shift. Since Step 2 has the lowest capacity of all of the resources in the system, it is identified as the factory throughput constraint. The factory cannot produce any more than this step can run, and any time this step is idle, factory capacity is irreversibly lost. As part of the exploitation process, its rate is identified as the "drumbeat" with which to synchronize the rest of the production line. To fully exploit the capacity of the constraint, it must have three things available at all times: material to work on or work-in-progress (WIP), a manufacturing cell, and a skilled operator. Subordination of the rest of the resources of the factory involves ensuring the constraint has its requirements satisfied at all times. If the factory capacity is to be raised, the capacity at Step 2 must be raised by improvement projects or equipment acquisition, and if it is raised beyond 900 wafers per shift, then Step 2 is broken as the constraint and Step 3 takes its place.

The first requirement for exploitation of the constraint is WIP, and other resources must be subordinated to ensure that the constraint is always fed. One cause of the constraint starving is the inevitable breakdown of Machine A (variable availability). One way to protect the constraint is to place a WIP buffer between Machine A and Step 2. The size of the buffer is based on the historical distribution of times to repair Machine A. Machine A is subordinated to Step 2 by always being run in such a way as to maintain the correct level in the buffer. Too much WIP in the buffer increases overall factory throughput time (TPT) but does not raise output. Too little WIP in the buffer risks factory capacity.

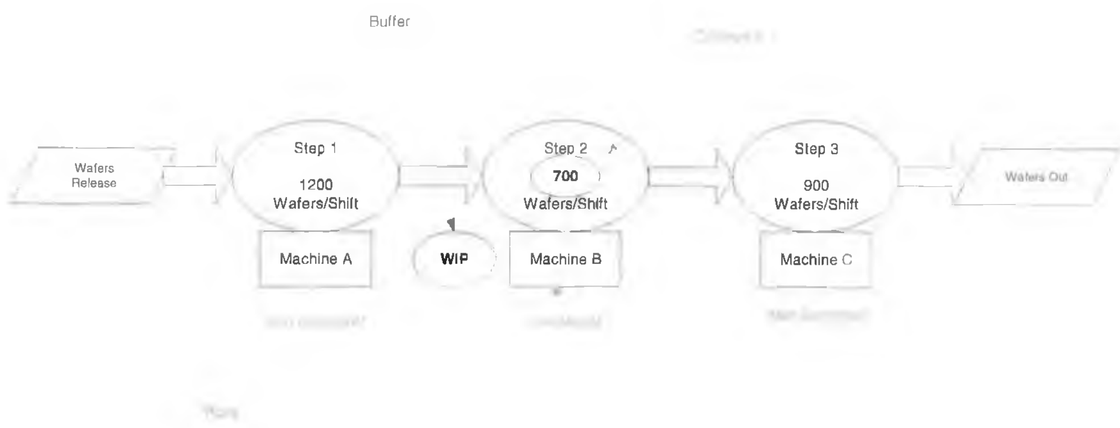


Figure 5.2: the Drum-Buffer-Rope 'DBR' factory

Another way to ensure that WIP is fed to the constraint is to control material release. Subordinating material release to the constraint involves allowing the constraint to pull for work it requires. This concept is described in TOC as tying the "rope" between the constraint and material release so that the constraint can "pull in" the work it needs as it needs it.

The TOC-based approach to WIP management gets its name from the combination of these ideas: drum-buffer-rope or DBR (Figure 5.2). Consistent use of these ideas drives the factory towards maximum throughput at minimum throughput time in the face of any variability in the availability of equipment.

5.3 Photolithography Scheduling Problem

Photolithography is a complex manufacturing area and is the constraint process, so care must be taken in scheduling to prevent severe reduction in product yield. In this respect, the impact of the photolithography cluster performance on successful manufacturing and the economical stability of a factory is crucial.

The dynamics of scheduling photolithography are not limited to unpredicted downtime of tools and sudden changes to product demands during production, there are also predictable changes that occur over a variety of time-scales. The equipment pool, as one of the most critical pools in production, changes as older tools (flexible manufacturing cells) which can produce a reduced set of layers as determined by the qualifying matrix work in parallel with new tools having improved technology and capabilities. These changes, in addition to new processes being introduced, cause complexity in scheduling a specific lot for a specific tool at a certain point of time.

The purchase of more equipment is a primary solution, but it is not cost effective when the manufacturing cell costs more than 15 million US dollars. Therefore, the maximum utilization of the equipment is essential. Good scheduling is essential to provide the best process parameters as well as production plan. Therefore, understanding the performance of photolithography has been a key focus for several years. Many deterministic models were developed to estimate the behaviour of the toolset under different loading conditions. These models of capacity calculations, processing times, and production plans have been the foundation and source of the data that is input to the model.

The toolset under investigation has 21 FMCs (tools) and can process 13 layers in a product menu of 15 products. The key questions which have driven the development of this IPS model are:

i) Which tool is best placed to process the next lot arriving in this area ?

ii) Is the capacity of this set of tools sufficient to produce enough wafers to fulfil demand?

The model merges the photolithography tools (FMC) performance data (based on the model in chapter four), intelligent scheduling, and practical deployment restrictions to come up with a prediction of toolset performance under different loading conditions.

5.3.1 Nature of the Photolithography Area

The following are the main constraints in the photolithography cluster in semiconductor manufacturing:

1. <u>Complex Product Flows</u>

   The complexity of product flows within photolithography has been discussed earlier.

2. <u>Random Yields</u>

Process yields are uncertain and vary due to environmental conditions, problems with production equipment or material. Yields for well-established products may be predicted using historical data, but the constant introduction of new products and technologies makes yield estimation a major problem. Cunningham [189] provides a survey and comparison of statistical yield estimation models in use in industry.

3. Diverse Equipment

   In well-established Factory, due to the use of more than two or three technologies in the photolithography area along with new products developing, the photolithography toolsets have grown in complexity with several distinct tool types. As a result, the manufacturing team had set a deployment practices summary sheet (qualifying matrix), which documented the tool technical restriction states. These tool states document the level of qualification of a photolithography tool for running certain layers. Each tool in photolithography has multiple generations having different capability restrictions.

4. Equipment Downtime

   The production equipment used in semiconductor manufacturing is extremely sophisticated. It requires extensive preventive maintenance and calibration, and is still subject to unpredictable failures. It is estimated that the main cause of uncertainty in semiconductor operations is due to unpredictable (unscheduled) equipment downtime, which is itself also cited as major problem. For bottleneck machines or flexible manufacturing cells, this time is particularly sensitive as it affects the total throughput time of the system.

5. Production and Development in Shared Facilities

   Due to the constant development of new products and processes, very often same equipment is used for both production lots and engineering test and qualification lots.

6. Data Availability and Maintenance

   The sheer volume of data in a semiconductor manufacturing facility makes data acquisition and maintenance an extremely time-consuming and difficult task. For each operation a product undergoes, information like processing times and yields has to be recorded. The constant introduction of new product types to keep up with the changing markets further complicates this problem, which is also compounded as one moves from the front-end towards the final testing stages due to multiple packaging and co-production possibilities.

In order to address the complexities of the photolithography cluster in the semiconductor manufacturing, a list of the parameters that affect on the throughput time per lot/wafer within the photolithography manufacturing area is shown below:

- Product type
- Batch size
- Previous product
- Tool setup time (layer/product)
- Product priority
- Preventive maintenance
- Unscheduled maintenance
- Number of qualified tools available
- WIP inventory
- Material Handling System (MHS)

An enormous amount of time is needed to examine and schedule all these parameters.

To summarize the problem, the photolithography area considers as a factory bottleneck. However, fabs that employ simulation often model the area in less detail and treat each individual toolset as a black box. The decisions made in pholithography toolset can have an influential impact on the total TPT. That urges to find a methodology to model and simulate this complex area in order to experiment and optimize the performance of the photolithography area.

5.4 Intelligent Photolithography Scheduling Model (IPS)

The proposed IPS model was built to assist decision making in the photolithography area and to improve the performance of the photolithography toolset using intelligent scheduling approach. The hybrid model was configured to receive information from production files and then use the data to make the selection decision. The model can learn more by inserting data in the neural network module. The model development methodology was shown in Figure 3.8 (Chapter 3).

5.4.1 Model Objectives

As the photolithography area has been identified as a factory constraint cluster, TOC steps two and three provided the insights about the model objectives. The model aims to exploit the photolithography tools within the system. The proposed IPS model objectives apply to three different planning levels:

I. Tactical (Strategic) planning
II. Intermediate planning
III. Shop floor planning (scheduling)

On the tactical level, the model output is used to

- Predict the approximate number of tools needed in the coming period.
- Provide a robust stochastic model to mimic the floor in the Photolithography area.

On the intermediate planning level, the model results could be used to

- Test the impact of product-mix / volume on the performance of Photolithography area.
- Perform sensitivity analysis on planning parameters in the manufacturing area.
- Determine the average utilization of every tool.
- Find the WIP inventory along the production flow.
- Examine different production schedules in order to estimate the effect of changes before the implementation.

- link the existing models to the new model so as to import data.
- Explore the impact of production flow dynamics on delivery time.
- Provide the planner with infomation about the constraint equipment (bottlenecks) and their impact on lead-time.

Finally, at the floor level, the model simulates one of the main issues on the floor, which is lot selection. The model provides the shop floor

- Near optimum solution of the sequence of the lots to be processed in order to increase the productivity and reduce lead times.
- Operational recommendations to improve the performance in the photolithography area.
- Impact of the selection criteria on the process performance.

5.4.2 Model Assumptions

The model assumptions are similar to previous FMC model in chapter four. (see Table 4.1). The new assumptions in the IPS model are:

- Tools are set to perform specific layers based on technical qualifications.
- Tools can be qualified to perform more than one layer.
- Lots sequence in the actual floor is random.
- The tools are down for maintenance after completing the wafer (no interruption of the operations).

The industrial partner has verified all the assumptions.

5.4.3 Performance Measures

The selection of performance measures depends on many things such as the application, nature of the production system and administration requirements. In the case under study, the performance measures were set within the TOC framework. The model was designed to allow the use of various performance measures, although this thesis has concentrated on the following.

TOC I: Production Management

This level concerns bottlenecks, scheduling and inventory reduction. The selection of measures of performance selected to evaluate these criteria are

1. Product/Layer Cycle Time
2. Machine Utilization
3. Total Waiting Time
4. Average Delays (Tardiness)

TOC II: Throughput Analysis

This level concerns system throughput time, system constraints, and production capacities. The selection of measures of performance which were considered to evaluate these criteria are:

1. Machine Idle Time
2. Production Rate
3. Throughput Time

TOC III: Logical Processes

This third level is the general application of TOC reasoning to identify the factors that are significant in developing a robust solution for scheduling. Statistical analysis tools are used such as ANOM, ANOVA, and Taguchi methodology on the measures from TOCI and TOCII.

5.4.4 Conceptual Modeling Approaches

The IPS conceptual modeling phase have used block Diagrams, IDEF0, and SASM modeling approaches to the photolithography toolset.

5.4.4.1 Block Diagram

The overall blocks of the model of the photolithography toolset in semiconductor manufacturing are shown in Figure 5.3.

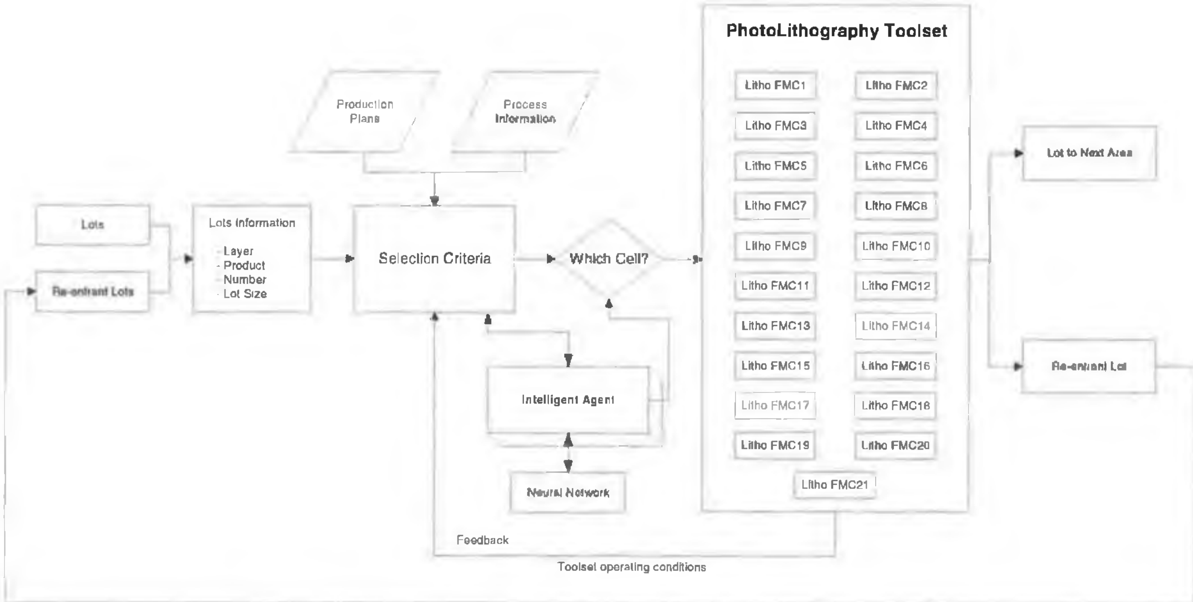


Figure 5.3: block diagram of the flow of lots in the model

Scheduling of photolithography typically starts when there are a number of lots of wafers positioned at specific steps in the process (some waiting, some in process) and a number of active resources (some idle, some processing, some down for maintenance). The model starts to schedule lots in the block before the toolset as shown in Figure 5.3.

5.4.4.2 IDEF0

A. Aggregate level of model A1

Figure 5.4 shows the top level of the developed IDEF0 model for wafer fabrication and indicates the inputs (wafers, process planning, and process preparations), the control (process characteristics and process factors), the mechanisms (layer, tool, product, and CAM software) and output (finished product) of the system. The A1 has been decomposed into second level of detail, (Figure 5.5) which includes the processes that the wafer should go through to be fabricated.

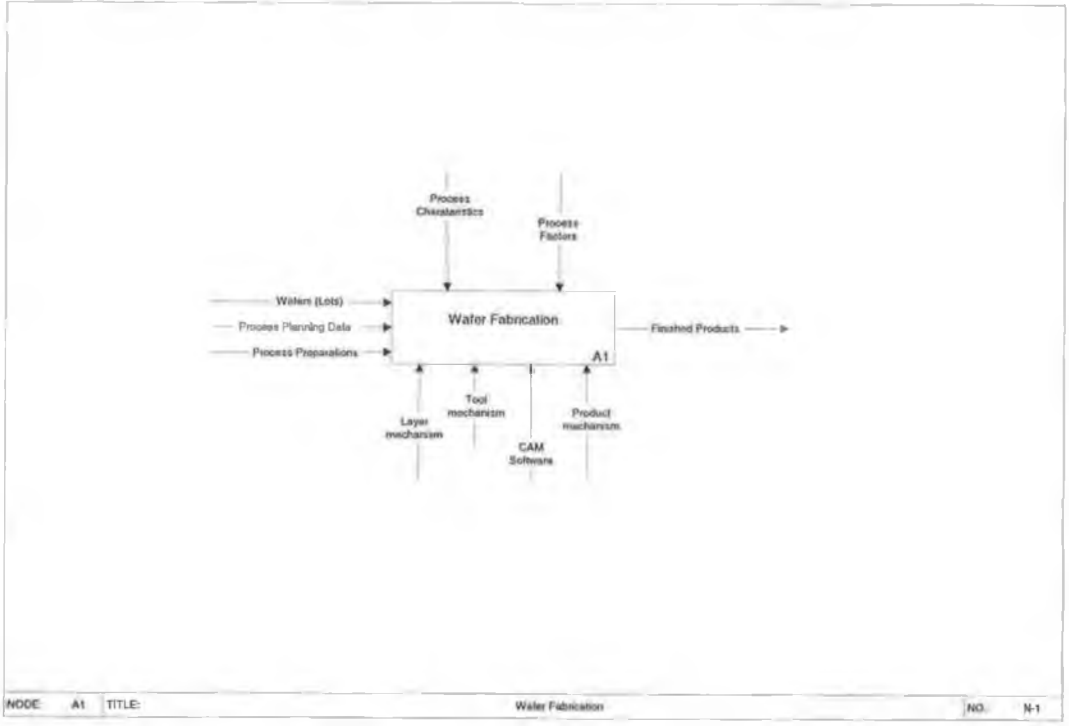


Figure 5.4: Top level of the developed model for wafer fabrication - A1

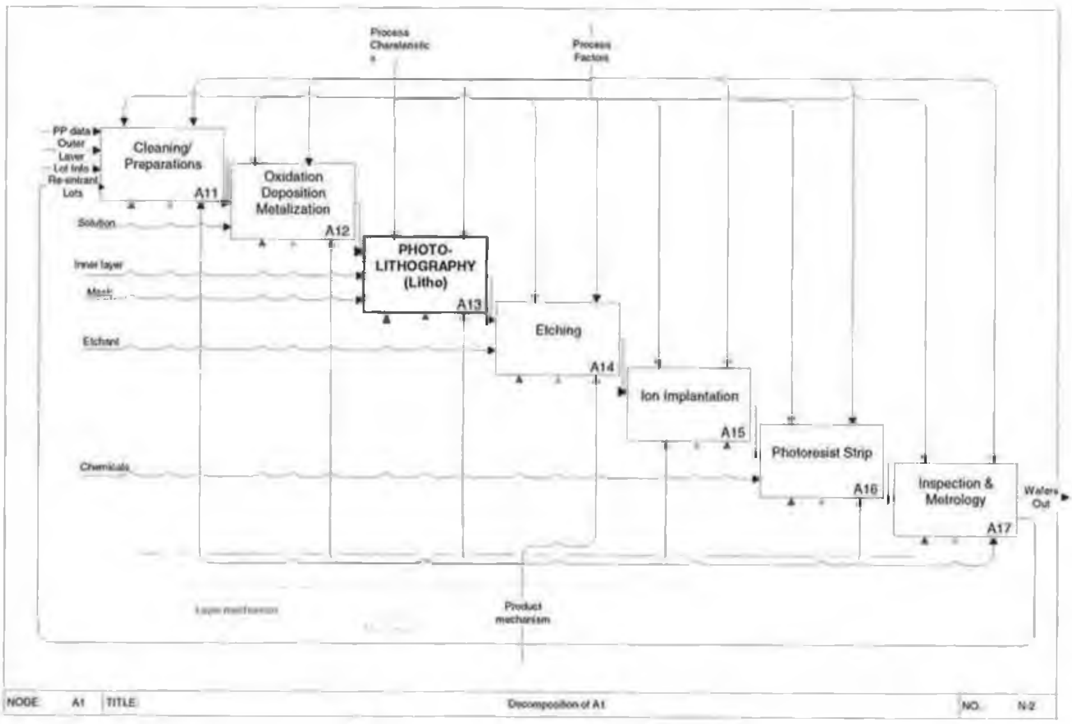


Figure 5.5: Second level of systematic developed model for wafer fabrication

B. Photolithography - A13

The photolithography process has described in detail in chapter five. The flows of lots and information in the model are shown in Figure 5.6. Wafers come in lots to the photolithography area where the information about the lots has to be identified. The model has a set of selection criteria (A132) that have been verified by the manufacturing team. The tool selection (A133) involves a more complex processing sequence using an intelligent agent to assist the selection. The photolithography tool or FMC (A134) has been decomposed in detail earlier in Figure 4.8. Once the layer has been processed, the lots move to next step based on the production settings (A135) of the lot that assigns the incoming layer and the next manufacturing area.

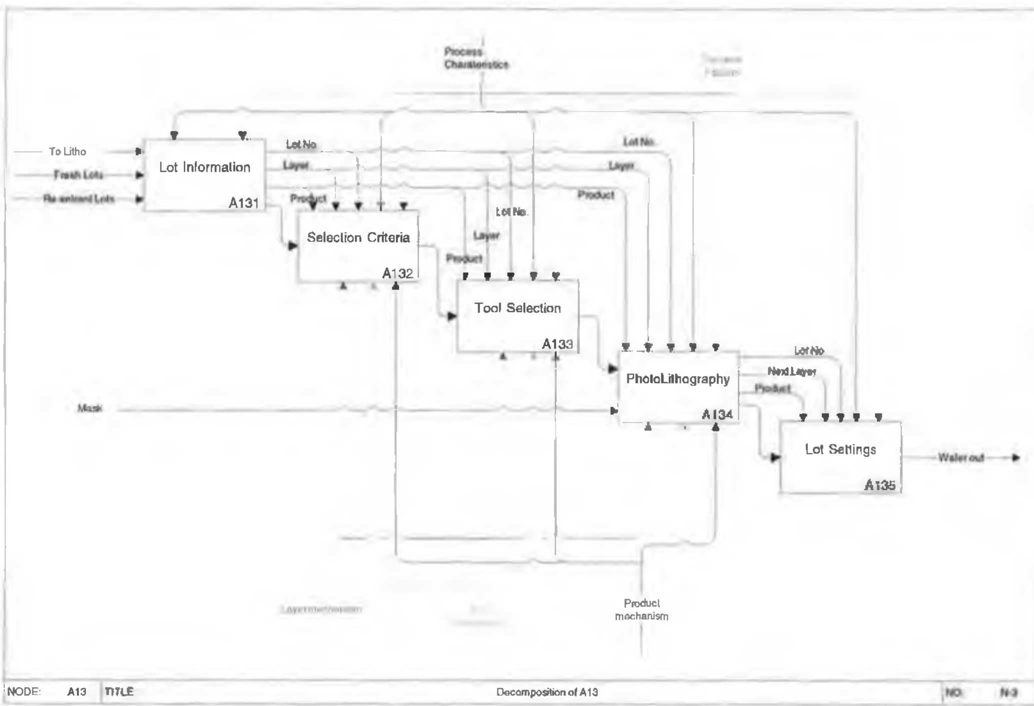


Figure 5.6: photolithography model steps in detail

5.4.5 Model Coding

A hierarchical-structured approach has been used to develop the model of photolithography toolset using simulation software package EXTEND (ver.5 Imaginethat, Inc.). The model focuses on the scheduling problem, and the major aim of this study, process optimization. There are three main components in the top level of the hierarchy including production schedule, Photolithography toolset, and output as shown in Figure 5.7.

For a better and more comprehensive illustration, the model has been modeled using SASM to represent the lots flow in the model. The production schedule of the model (left-most block in Figure 5.7) has been shown schematically in Figure 5.8. The production planning settings including number of wafers to start the production, product-mix, required wafers of each product, dispatching rule, priority, the first layer, and maintenance schedule have to be entered into the model. The model handles the data in two ways spreadsheets and global arrays. Wafers are normally generated using statistical distribution (Exponential distribution) or time-based schedule. The information about each lot come is recorded using the global array, which allows it to be updated during the simulation run. Meanwhile, the spreadsheet keeps records of the information required for analysis. Based on the schedule, the generated wafers are assembled in lots of 25 wafers unless the product or layer changes.

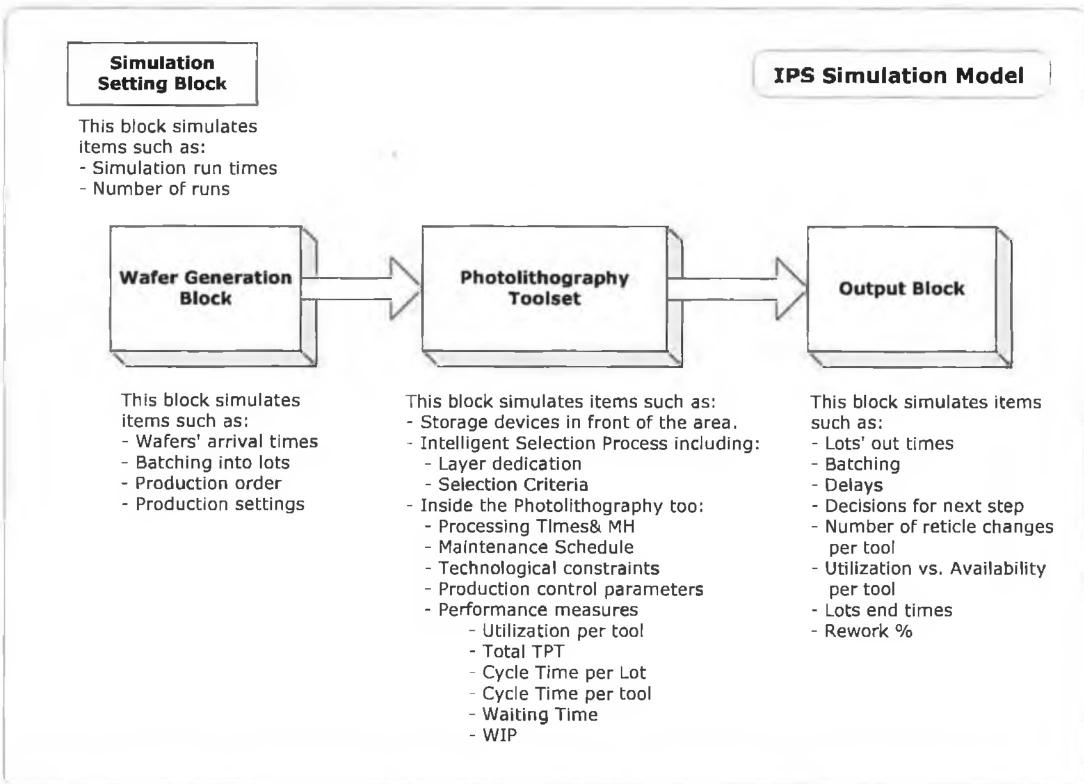


Figure 5.7: Photolithography simulation model blocks

Some lots can be less than 25 wafers and the model has been customized to handle this. Once completed, the lots pass the photolithography toolset block.

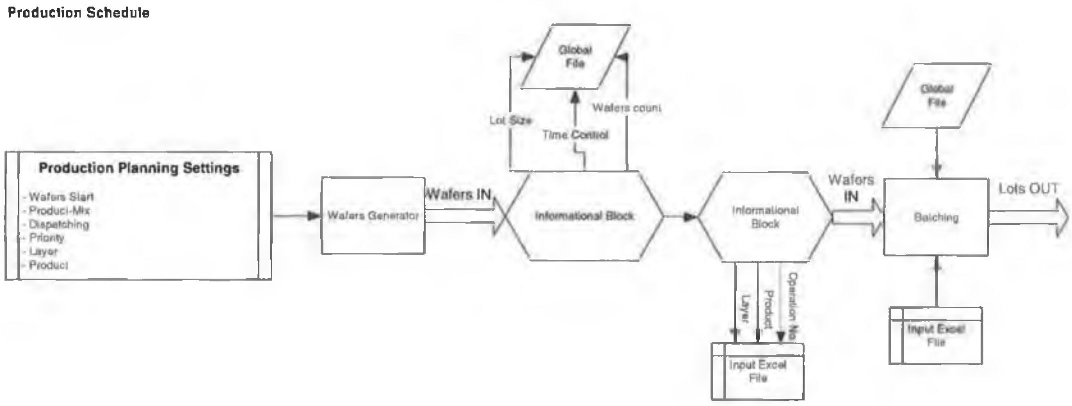


Figure 5.8: Schematic Diagram of Production Schedule Block

The lots come to toolset block, in Figure 5.7, where there is a buffer at the start of this step followed by information retrieval blocks, see Figure 5.9. The model decides based on this information which layer to be processed for the lot. Every layer has its own path in the model flow followed by buffer. The scheduling of lots is the major aim of the incoming step. Intelligent agent and set of selection criteria have been considered to arrange the lot scheduling. Once the decision have been made and a tool is selected to process the lot, information concerning the process is retrieved automatically and the tool condition information is updated in both the intelligent agent block and the data stores elements (global array and spreadsheets). Every tool has been individually modeled and simulated as a photolithography FMC model discussed in chapter four. When the lot is processed, either it goes to another manufacturing processes or re-entery to the photolithography toolset to build a new layer (re-entrant lots).

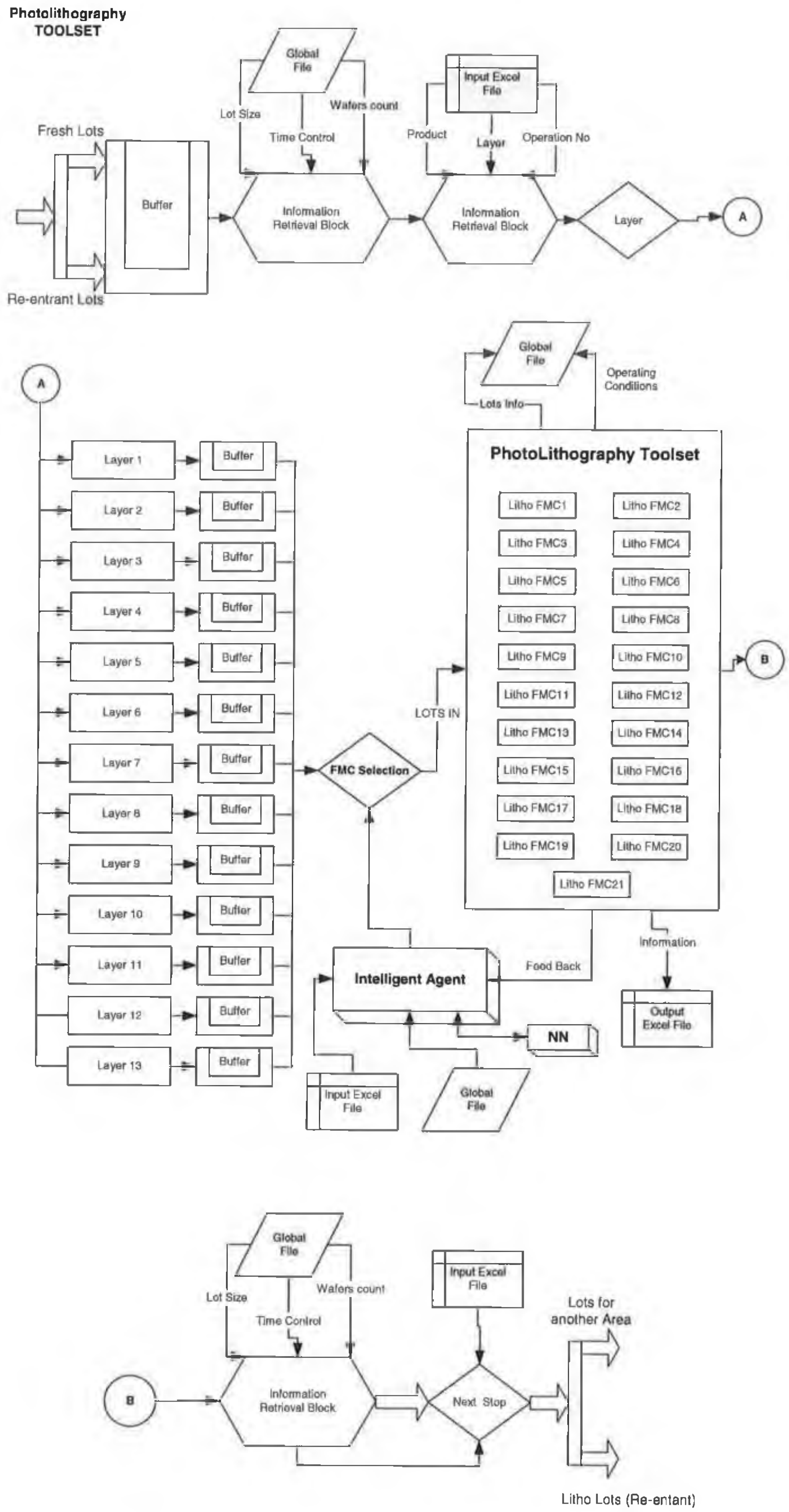


Figure 5.9: Schematic Diagram of Photolithography Toolset Block

The third block (Figure 5.7) collects the results needed for analysis and evaluation. This block gives the signal to stop the simulation once the production of the 'wafers started' has been completed. The data is sent to the output files for graphing and presentation purposes, Figure 5.10.

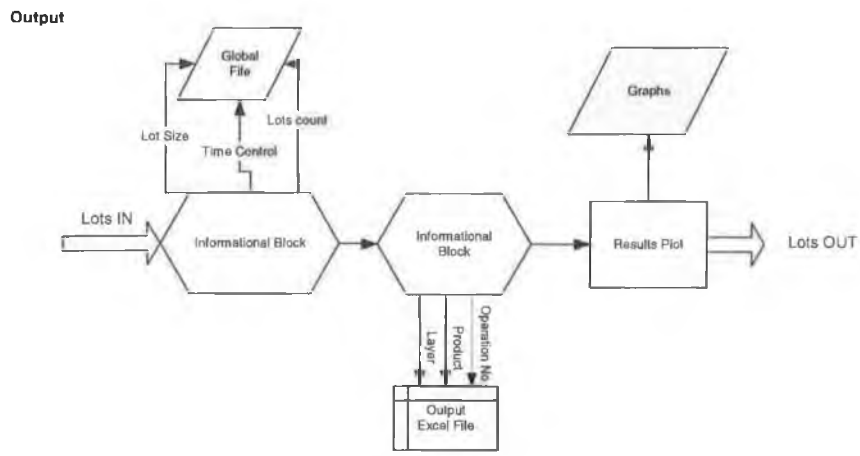


Figure 5.10: Schematic Diagram of Output Block

5.5 Intelligent Scheduling Methodology

The intelligent-agent based module associated with the IPS model provides decision support at control level as well as production/tactical planning level. The previous attempts had provided some insights into the functioning of the photolithography area but were not able to obtain a quick robust solution to handle the lots scheduling problem. The solution space is large due to the number and range of controllable variables and assignable parameters. Moreover, the complex nature of the process flow leads to numerous conflicts in decision-making. This makes the evaluation and testing of all potential solutions in a search space rather difficult, risky, and uneconomic. The intelligent-agent evaluates the major decision making criteria using a heuristically established set of rules. The parameters assumed were a mixture between dynamic parameters such as tool status and buffer status and static parameters such as priority products or layers and product sequence.

It is worth saying that most of the FMS decisions are generally affected by decisions made earlier, and attempts to isolate individual decisions and treat them as independent decisions are likely to be problematic.

This thesis puts forward a model that provides support for decision-making in a large and complex domain through selective exploration of the solution space for a single toolset, as well as the assessment of the effects of other related decisions. The concept of the generalized simulation model is key to the research objective. The following sections discuss the main elements in the intelligent agent that have been used to solve scheduling within the photolithography toolset.

- ***Qualifying Matrix***

The manufacturing team uses a qualifying matrix (QM) that declares which tool is capable of processing each layer. The factory cannot replace older equipment as long it is still functioning and the replacement period is not due. The qualifying matrix is updated periodically based on manufacturing policies. Table 5.1 illustrates a sample of the qualifying matrix showing the tools and the layers on which they are able to perform.

Table 5.1: Sample of the Qualifying Matrix

Tool No. \ Layer No.->	1	2	3	4	5	6	7	8	9	10	11	12	13
X01	✓	✓											
X02	✓	✓			✓								
X03	✓	✓											
X04					✓								
X05					✓					✓			
X06												✓	✓
X07			✓	✓									
X08						✓	✓						
X09						✓	✓						
X10							✓			✓		✓	
X11												✓	✓
X12												✓	✓
X13						✓	✓						
X14								✓	✓				
X15								✓	✓				
X16									✓	✓	✓		
X17										✓	✓		
X18										✓	✓		✓
X19			✓	✓				✓	✓				
X20			✓	✓									✓
X21				✓									

The QM is a two dimensional matrix which ignores the particular product to concentrate on layers and tools. Therefore, it is assumed that all the tools can process a qualified layer on any product. For example, the manufacturing team always assigns the new tools to perform the hard/complex layers as the older tools may not be capable of achieving the required quality in a timely manner.

- ***Selection Criteria***

The manufacturing team has a significant input in assigning a short list of the major constraints that photolithography tools encounter within the lots flow. Technically, the problem of scheduling lots is a problem of assignment. There are several criteria that will affect the decision to select a tool to perform the incoming lot. The schedule generated for a manufacturing run is highly dependent on the particular criteria used in the scheduling process. These criteria can be either process-oriented or wafers-oriented (Figure 5.11):

A. Process oriented criteria are the criteria that concern the equipment itself such as technology, maintenance, .. etc.

B. Wafer oriented criteria are the ones which concern the lot information such as product, layer, .. etc.

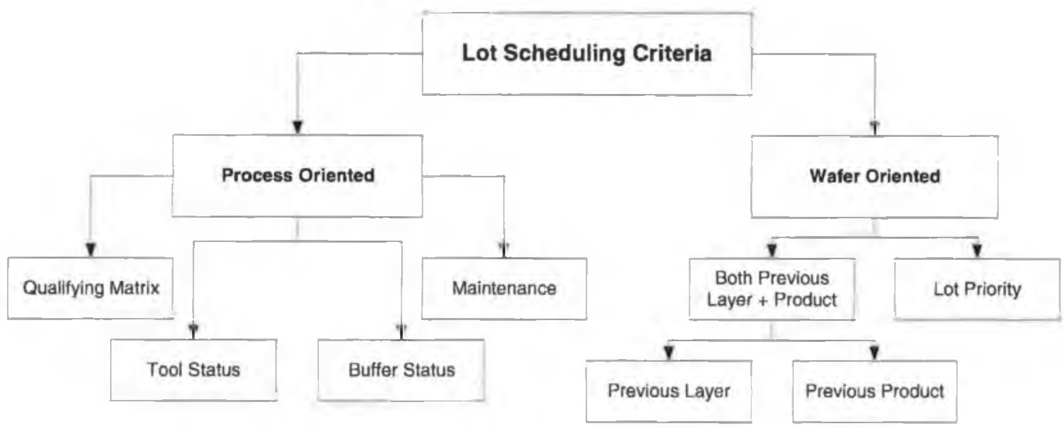


Figure 5.11: Lots scheduling criteria

The criteria shown in Figure 5.11 are the most effective for decision-making based on manufacturing team. The main goal in this step of modeling scheduling intelligently is to set weights for these criteria based on their importance. The goal is to minimize setup times per tool due to change in layers and products. The following order of evaluating these criteria has been established with the manufacturing team.

- Qualifying Matrix
- Maintenance
- Tool status
- Buffer status
- Lot priority
- Previous layer & product
- Previous layer
- Previous product

This means the selection of the best tool for incoming lot will be taken through a series of arguments. First, the selected tool must qualify to build the required layer for the lot. Second, it should not be in maintenance either preventive or unscheduled. After that, the model has to check if the tool is idle and if not, check the buffer status in front of the tool. The lot priority can be set to high (hot) or regular priority. Finally, the previous lot characteristics have an important effect to the scheduling because a tool already processing same layer or product can be given priority to process the incoming lot.

- ***Selection Algorithm***

The major issue for scheduling lots is the algorithm/approach for selecting the tool to process the lot. The model uses a weighted-score approach for evaluating the possible alternatives. The description of the algorithm and some issues involved is given in this section.

The scheduling problem is defined as follows:

Assume an incoming lot O_{ij} has to process layer 'j' for product 'i',

where $i = 1, 2, \ldots\ldots, n_i$, and $j = 1, 2, 3, \ldots\ldots, n_j$.

The scheduling problem is to assign a specific tool to process the incoming lot. The tool with highest score is the optimum for the selected criteria, as shown in the formula below;

$$Max\left|S_m\right|_{m=1}^{m=N_{QT}}$$

where,

S_m is the score of tool 'm' and N_{QT} is the number of qualified tools for this layer.

The score of tool 'm' can be calculated based on the following equations:

$$S_m = \sum_{c=1}^{n}\sum^{r} K_c \pi_r$$

where,

π_r is the value of selection criteria 'r', $r = M, TS, BR, xl, xp, H$

n is the number of selection criteria, and

K_c is binary variable (0-1) to set if the criterion is applied (1) or not (0).

The selection criteria for the case under investigation are described below:

Maintenance criterion (M):

The criterion has a value of w_{MR} if the tool is down and w_{MO} if tool is ready.

$$\pi_M = w_{MR}K_M + w_{MO}(1-K_M)$$

where,

π_M is the value of the maintenance function,

w_{MR} is the weighting if tool is down for maintenance,

w_{MO} is the weighting if tool is ready, and

K_M is the maintenance flag (1= down, 0= ready).

Tool Status criterion (TS):

The criterion has a value of w_{TSR} if the tool is busy and w_{TSO} if tool is idle.

$$\pi_{TS} = w_{TSR}K_{TS} + w_{TSO}(1-K_{TS})$$

where,

π_{TS} is the value of the tool status function,

w_{TSR} is the weighting if tool is busy,

w_{TSO} is the weighting if tool is idle, and

K_{TS} is the tool flag (1= busy, 0= Idle).

Tool Buffer Status criterion(BR):

The criterion has different values based on how many lots are in the buffer in front of the tool.

$$\pi_B = w_{BR} + \left[(w_{BO} - w_{BR}) * \left[1 - \frac{N_o}{SB} \right] \right]$$

where,

π_B is the value of the tool buffer status function,

w_{BR} is the weighting if tool buffer is full,

w_{BRO} is the weighting if tool buffer is empty,

N_o is the number of lots inside the buffer, and

SB is the size of the buffer.

Previous Layer criterion (xl):

The criterion has a value of w_{xlR} if the tool is processing same layer and w_{xlO} if tool is not.

$$\pi_{xl} = w_{xlR} K_{xl} + w_{xlO}$$

where,

π_{xl} is the value of the previous layer function,

w_{xlR} is the weighting if layers are same,

w_{xlO} is the weighting if different layers, and

K_{xl} is the layer flag (1= same, 0= different).

Previous Product criterion (xp):

The criterion has a value of w_{xpR} if the tool is processing same product and w_{xpO} if tool is not.

$$\pi_{xp} = w_{xpR} K_{xp} + w_{xpO}$$

where,

π_{xp} is the value of the previous product function,

w_{xpR} is the weighting if products are same,

w_{xpO} is the weighting if different products, and

K_{xp} is the layer flag (1= same, 0= different).

Lot Priority criterion (H):

The criterion has a value of w_{HR} if the lot has high priority (hot lot) and w_{HO} if lot is not.

$$\pi_H = w_{HR} K_H + w_{HO}$$

where,

π_H is the value of the lot priority function,

w_{HR} is the weighting if lot priority is high,

w_{HO} is the weighting if lot is regular, and

K_H is the priority flag (1= high, 0= regular).

- ***Scheduling Methodology***

The scheduling methodology used in this thesis is a dynamic dispatching technique where, every time a lot becomes ready to be processed, a decision about which tool to process has to be made. However, here information about lots, tools, and the system is updated simultaneously to exploit toolset conditions and consider that in the selection decision. This methodology enables the manufacturing team to explore the alternatives as well as examine the production scenario before the real production takes place.

The intelligent-agent based approach has been integrated with a simulation model to help in scheduling the lot selection. It starts with the grouping of the tools inside the toolset. The tools have been grouped based on a qualifying matrix so every group has some common layers which it can process, however ability to process a particular layer is not exclusive to the group. The purpose of this grouping is to save more time in the scheduling process by applying the qualifying matrix at an early stage. The scheduling starts once the lot is about to enter the photolithography toolset, and in the early stages the model put the lots into the path based on the layer to be processed.

Lot information has to be read (product, layer, and lot number) before they reach the tool selection block. Scheduling is connected directly to the intelligent-agent based block where the scores of each qualifying tool are calculated, and the tool with the highest score is identified. Then the model sends the lot to the selected tool. Figure 5.12 shows the logic of selection criteria.

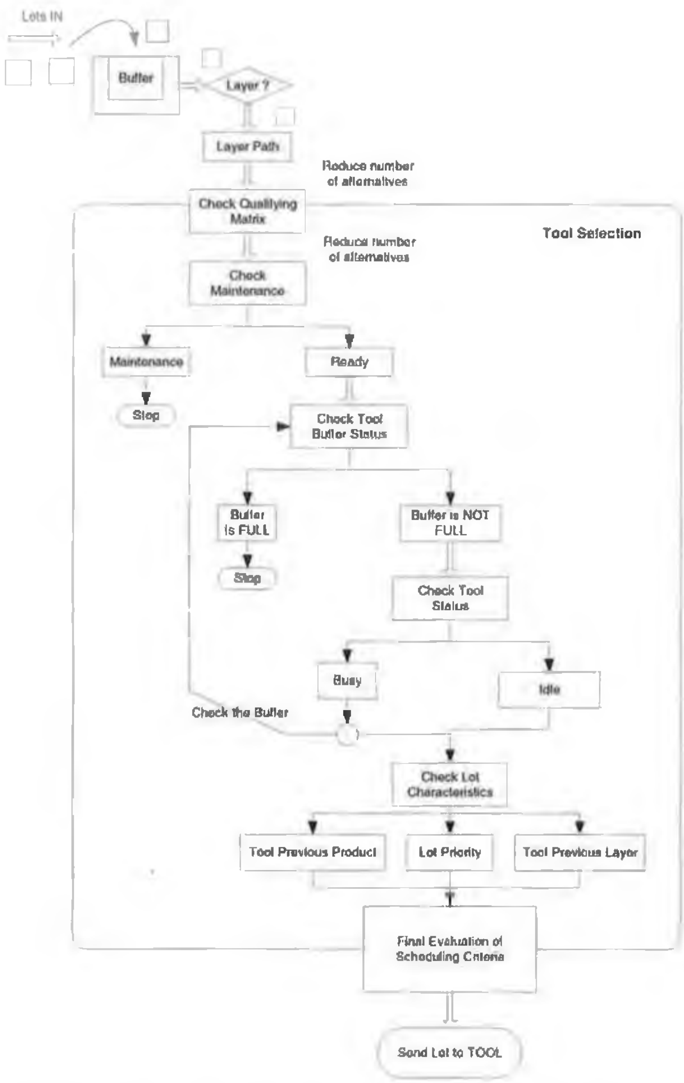


Figure 5.12: The Scheduling Criteria Logic

5.5.1 Model Optimization

In most cases, neural network approaches hold significant advantages especially when suitable training data is available with adequate quality level. An Artificial Neural Network (ANN) presents at least an alternative to the more conventional optimization techniques. ANN has been applied in order to optimize the selection criteria weights. Neural network has been discussed in chapter three.

ANN used takes the advantage of regression and attempts to predict input-output transformation functions based on restructuring the weights of the nodes. It is important to note that it is not necessary to develop complex algorithms, and once trained the ANNs have an ability to generalize data and produce outputs for previously unseen inputs. Optimization software "NeuroShell Predictor, Ward System Group Inc." [147] was used to build the ANN model. Figure 5.13 shows how the ANN is used with the simulation model to form a hybrid system.

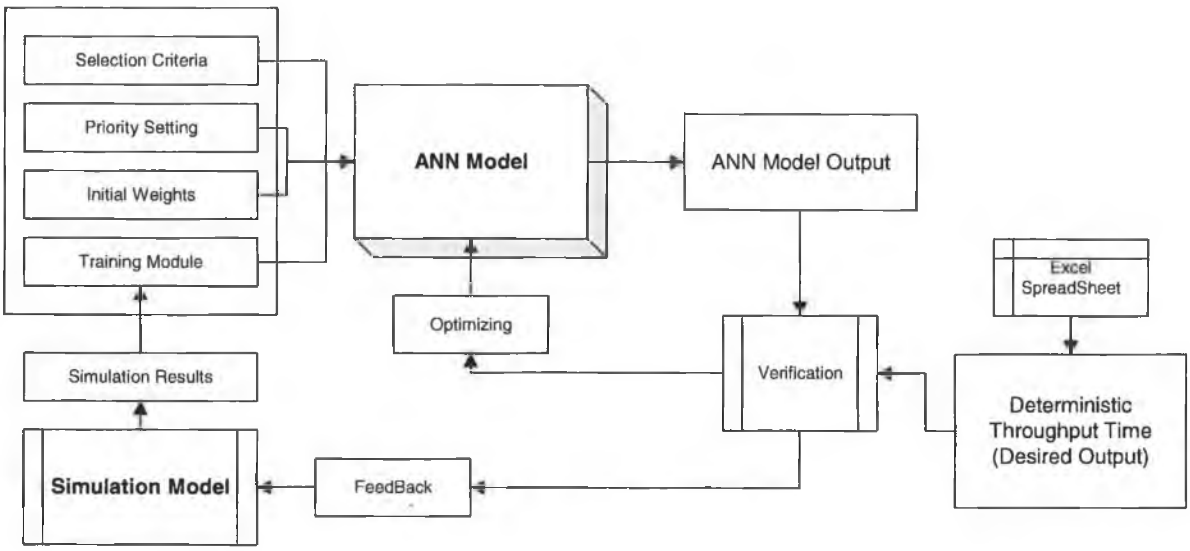


Figure 5.13: Architecture of ANN scheduling model

- ***Optimization Process***

The network model of continuous variables for optimization problems by Hopfield and Tank [143] is the most popular. The rationale behind optimization using neural networks is that a neural network can act as a goal seeking dynamic system and the equilibrium state of a neural network can minimize an abstract energy function [190]. The central part of the network is the formulation of an energy function based on an objective function and constraints of a given problem. Most of these approaches focus on the optimization of a given problem by iteratively searching for the minimum energy point. They do not provide a method to utilize the acquired knowledge to solve many other problems that share similar characteristics [191].

The major advantage of the ANN approach is that it utilizes the acquired weights of criteria in making better balance or sequence decisions. It involves three stages, which may be sequential:

1. the initial setting stage;
2. the learning (training) stage;
3. the knowledge refinement stage.

In the initial setting stage, the desired inputs and outputs have to be defined. The selected measures of the toolset performance were the number of reticle changes (RtC) and total throughput time (TPT). The model was initially trained using fixed operating conditions of 6000 wafer starts, 13 layers, and 10 products. The weights for use in the training are defined in a spreadsheet format or text. Each row in the spreadsheet represents a set of weights for the scheduling criteria as shown in Figure 5.14.

Show data

Path name of file: C:\... Settings\Amr\My Documents\My

Initial label row detected: yes

Number of columns read: 13

Number of data rows read: 32

	Exp	Mt On	Mt Off	Idle Tool	Busy Tool	Buffer Full	Buffer not	X-layer On	X-layer off	X-product On	X-product Off	Output (RtC)	Output TPT
1	1	-5	0	4	1	-1	4	5	0	5	0	1187	485
2	2	-5	1	6	1	-5	6	5	0	2	0	1159	440
3	3	-10	1	6	1	-5	6	5	0	2	0	1159	442
4	4	-240	25	20	10	-10	20	50	15	50	15	645	258
5	5	-245	25	100	25	-50	100	50	15	50	15	1137	257
6	6	-120	10	50	20	-10	50	60	10	60	10	1212	268
7	7	-120	10	60	10	-10	60	70	5	50	5	1134	294
8	8	-200	25	10	10	-10	10	70	5	50	5	512	268
9	9	-200	25	10	2	-20	10	80	5	50	5	512	268
10	10	-200	25	20	2	-30	20	60	5	40	5	876	264
11	11	-200	25	20	5	-20	20	60	5	40	5	878	264
12	12	-150	10	30	20	-10	30	70	5	50	5	602	284
13	13	-150	20	30	10	-5	30	80	5	60	5	602	284
14	14	-200	20	100	50	-50	100	150	5	120	5	1134	294
15	15	-200	20	15	5	-10	15	60	10	50	10	602	284
16	16	-200	20	10	0	-15	10	60	5	50	5	602	285
17	17	-120	10	10	5	-15	10	60	10	50	10	604	287
18	18	-120	10	10	5	-15	10	30	5	20	5	880	272
19	19	-120	10	10	5	-5	10	30	5	20	5	831	263
20	20	-120	10	10	5	-5	10	60	5	40	5	448	253
21	21	-120	10	15	0	-10	15	60	5	40	5	602	285
22	22	-120	10	15	5	-5	15	60	5	40	5	512	268
23	23	-150	25	20	5	-5	20	60	5	40	5	602	285
24	24	-120	10	20	5	-10	20	60	5	40	5	604	286
25	25	-120	10	10	5	-5	10	40	10	30	10	602	285
26	26	-120	10	10	5	-5	10	30	5	20	5	269	831
27	27	-120	10	10	5	-5	10	70	10	50	10	602	285
28	28	-120	10	15	10	-10	15	60	5	40	5	602	285
29	29	-120	10	10	5	-5	10	40	5	60	5	760	281
30	30	-120	10	10	5	-5	10	70	5	50	5	448	253
31	31	-120	10	12	5	-7	12	80	10	60	10	448	254

Total data rows: 32 Selected rows: 32 (from 1 to 32)

Figure 5.14: Sample of ANN training data format

In the learning stage, the data file is used to train the network. The parameter columns (inputs) and the output column have to be defined to the ANN before executing training. Once the model is run, the knowledge is stored in an output file and graphs can be obtained to make decisions.

The method employed in training the ANN depends on supervised learning. This implies that there is prior or available knowledge of the system to be modeled in the form of inputs-output pairs [165].

A simple module of NN was built to predict the near optimum weights of the selection criteria. The module has limits up to 20 input nodes in the input layer and maximum of 20 hidden nodes in the hidden layer. The output of the model was 2 nodes as mentioned earlier (RtC and TPT). The module learns to make predictions by learning patterns in the given data file. The software used facilitates integration of the NN with other programs, which helps to read the data from spreadsheet directly and export the result out to the same spreadsheet.

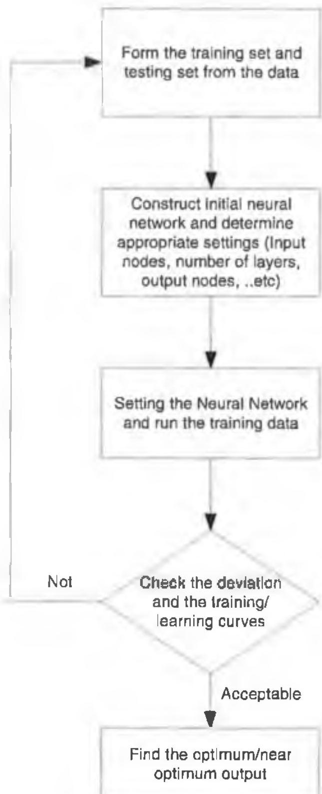
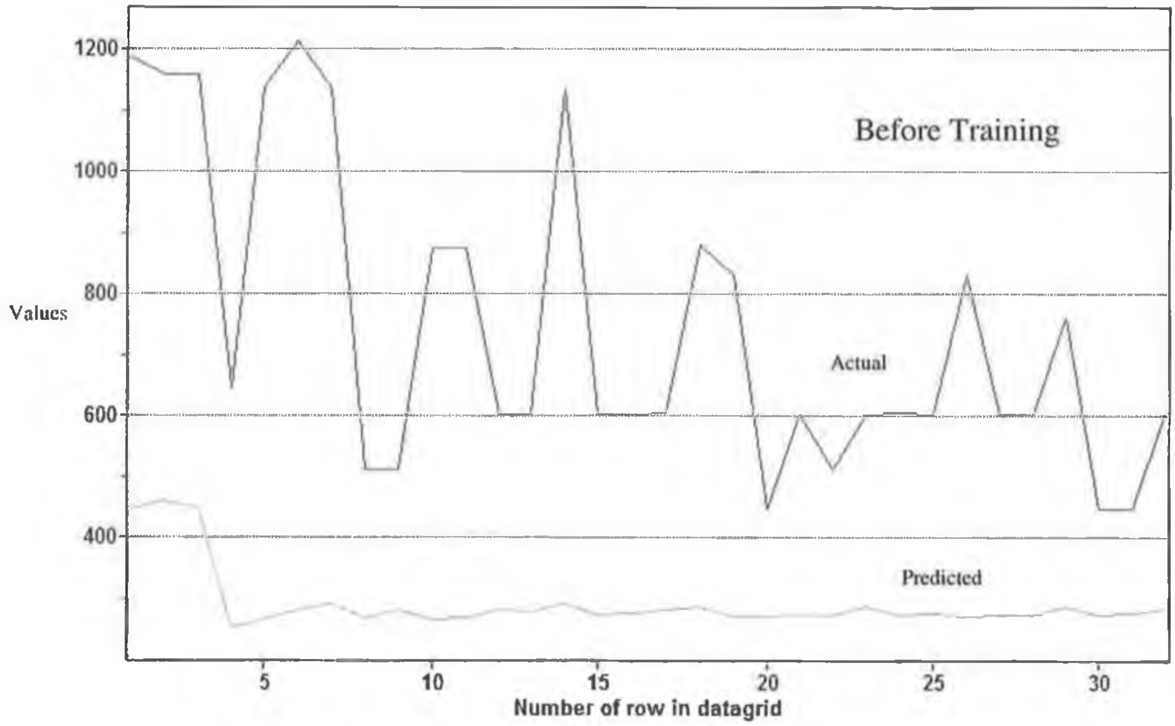
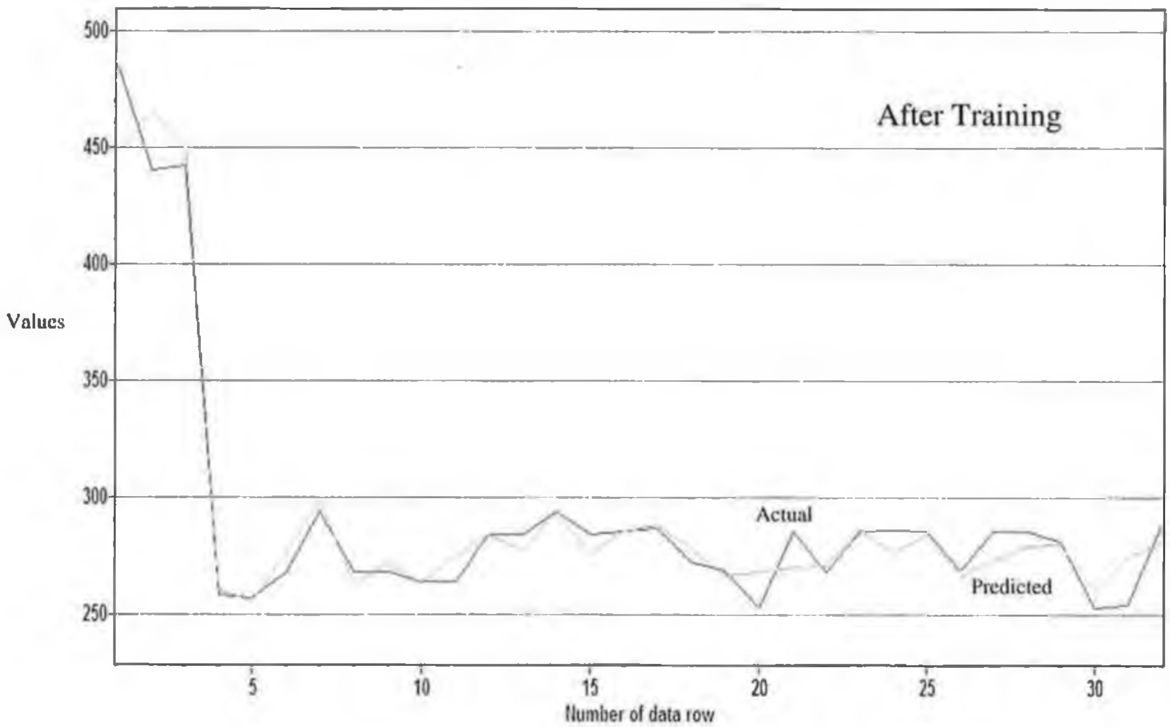


Figure 5.15: ANN training steps

Input patterns are propagated through the ANN to produce an output pattern that will initially be different than the desired output. The desired output has been assumed to be the theoretical sum of the total processing times and transportation time of lots in the model. This desired output is therefore considered to be a non-achievable target.



(a)



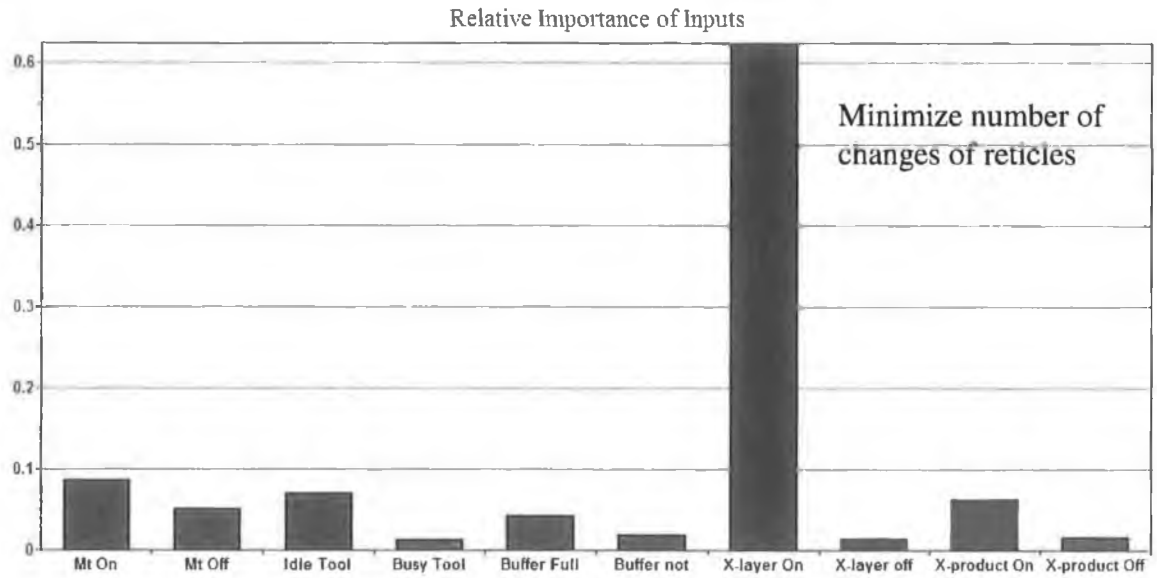
(b)

Figure 5.16: The training curves (a)before training,(b) after training

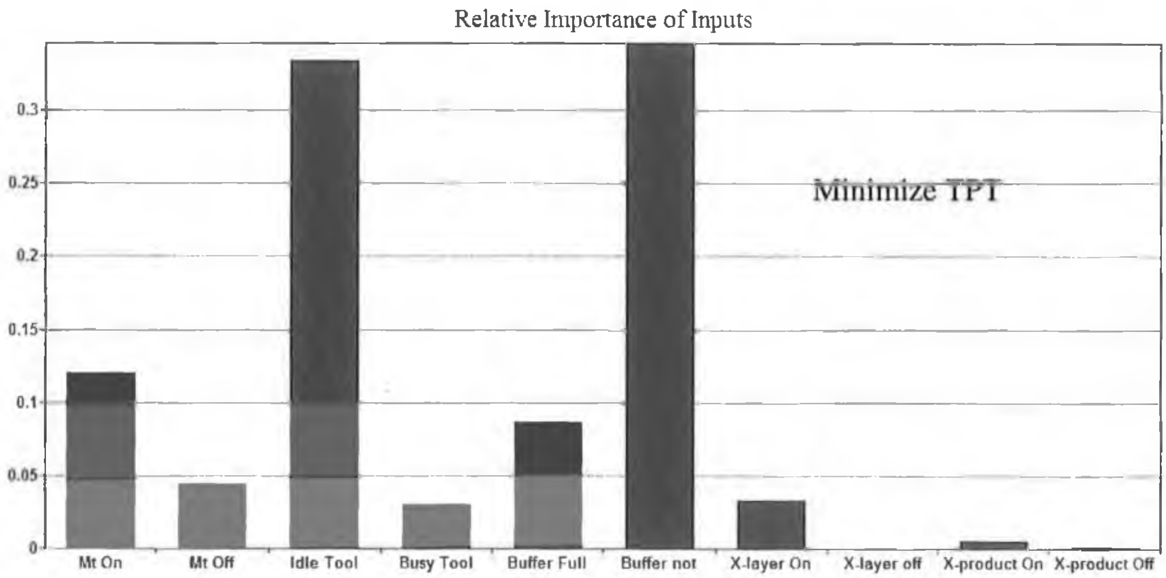
However, the training process consists of adjusting the weight values to produce minimum error of the outputs. Error is defined simply as the difference between the model output and the desired output (see Figure 5.15). The ANN has been developed in two-phases that each incrementally train the model for one objective function alone to assure the near optimum weights of the selection criteria. The improvement in the prediction is obvious in Figure 5.16(a,b) showing the performance of the ANN before the training phase and the output curve of the trained ANN module. The ANN module can provide a number of graphic displays such as actual output versus the network predictions, the relative importance of inputs, and the network learning curve.

The software also calculated the sensitivity of the output (error) to changes in the inputs. Figure 5.17 shows the relative importance between the selection criteria based on the assigned objective function. The previous layer seems to be the most significant criterion for RtC as a measure of performance, while the tool buffer status (full or not) and tool status (idle or busy) seem to be the most significant criterion for TPT as a measure of performance.

In summary, based on the knowledge gained from learning using a set of training exemplars, the ANN model could significantly improve the solution quality to find near optimum weights of selection criteria. The ANN has recommended two combinations of weights for the selection criteria based on TPT and RtC. These combinations provide the minimum TPT and RtC values repectively. Therefore, there are two alternatives for weighing scores for these criteria. The ANN outputs have been used to update the weights in the simulation model manually. If more training is given to the ANN, the quality of results is expected to increase. The performance of the ANN model is evaluated using comparison with the results of other simulation model outputs.



(a)



(b)

Figure 5.17: The relative importance of selection criteria based on (a)RtC, (b)TPT

- ***Numerical Example***

Assume that a lot comes with these configurations (Product '5', Layer '2', Normal priority). In this configuration, the qualifying matrix has three tools that can build this layer for that product, i.e. tool '01', tool '02', and tool '03'. First, the model will follow the logic indicated in Figure 5.12 by checking if any of these tools is out for maintenance reason. Then, let's assume that the

three tools are busy (Tool Status criterion), the model will examine the buffer of each tool along with the lot information (layer, product) to evaluate the tool scores. After that the model will apply the selection algorithm calculations to every qualified tool and come up with three scores S_{401}, S_{402}, and S_{403}.
For illustration sake, presume that S_{01} = 120, S_{02} = 85 and S_{03} = 65.
The model will send the lot to tool '01' because it gets the highest score.

5.5.2 Data Management

The model was built to support two levels of authorization, the administrative level and the operational level. The administrative settings allow the planners to set the values of the critical factors such as the qualifying matrix and maintenance schedule. Whilst the operational level is the experimental level for sensitivity analysis as well as the floor control level and at this level the engineers can change the other parameters such as the number of wafer starts, lots priorities, .. etc. The diagram in Figure 5.18 shows the different levels of data control levels and the inputs that the model needs to run.

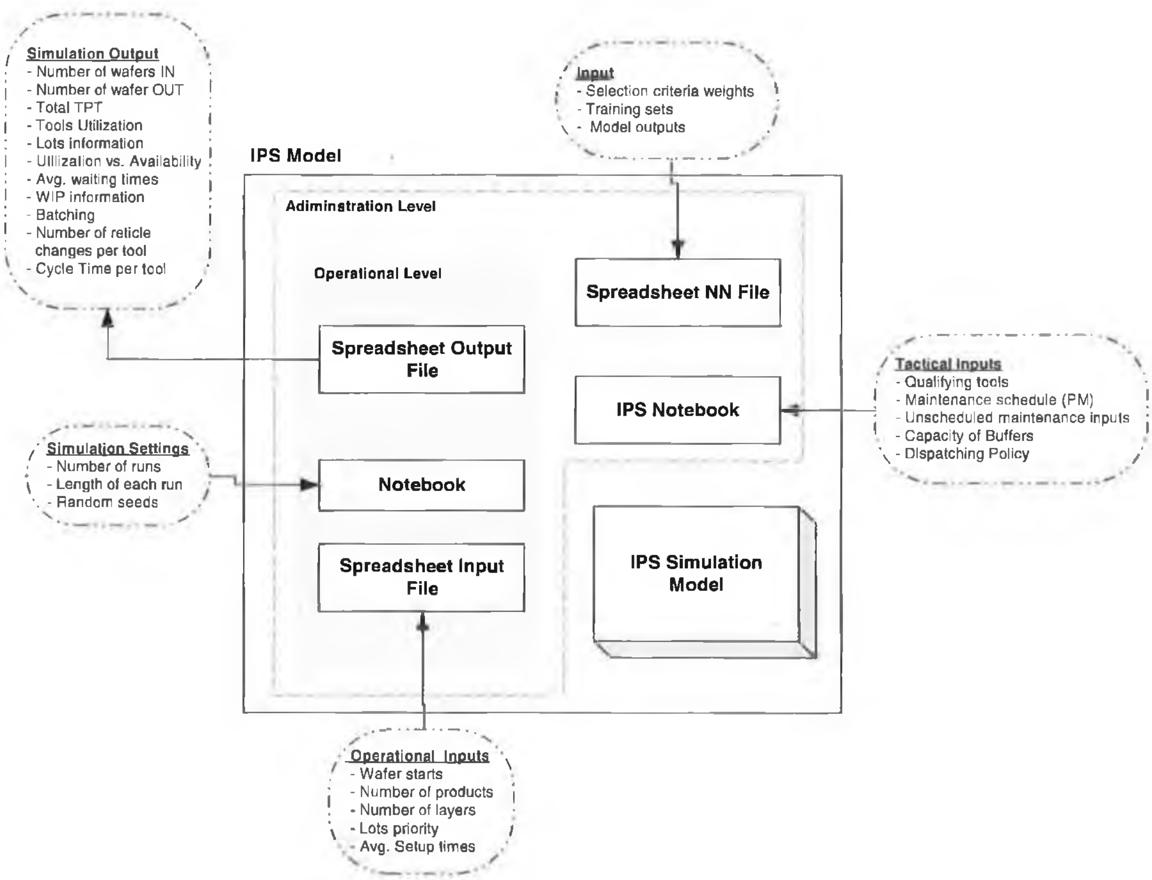


Figure 5.18: The IPS data management diagram

5.5.3 Model Verification and Validation

Many decisions are made based on data derived from simulation models. The strength of these decisions is a direct function of the validity of this data [178]. It is evident that validation is an integral part of simulation model building right from input data collection through model development to output data analysis. Integration of verification and validation with model development is crucial. The goal of this section is to verify that the model of the photolithography toolset is valid. A number of approaches were taken to confirm the status of the model.

- ***First verification approach***

This approach can be called in quality terms an 'internal audit'. The software used for simulation produced a trace file, which consists of detailed output representing the step-by-step progress of the model over time. This allowed detection of subtle errors. The trace file displayed that some stations had higher utilization values than would be expected from production. The numbers would be misleading if this were not corrected. To ensure that times were not overlooked, this verification was checked by people other than the modeller to confirm that the correct logic was followed for each event type.

- ***Second verification approach***

The output was checked for reasonableness, similar to an 'external audit'. A set of different runs that were given by the manufacturing team was simulated to ensure that the throughput time and cycle time levels were near to the actual levels. This indicates that the logic and assumptions in the model are correct.

- ***Third verification approach***

A decomposition approach to verification can be applied throughout the complex manufacturing simulation model because the method of construction allows the tracing of each small part of the model separately. This approach established greater confidence in the model as the errors can be detected quickly. Moreover, this approach was also successful in the previous FMC models.

Verification was carried out for clean data (this is an industrial expression means data without chaos or irregularities) and the simulation results were compared to the actual data, Table 5.2. The arithmetic average of deviation varied between 2 % - 16 % (see model implementation and remarks end of this section). The intelligent-agent based module was turned off to simulate what is happening in the actual production. The first check was for the tool processing times (overall) and to agree with the experts that the overall throughput time is reasonable. While part of the model was verified in chapter four, after including the complexity of the FMS as a whole, it needs reverification.

The verification process gave many useful insights for model building and execution. The simulation runs were performed on a PC Pentium III; the time needed to finish one simulation run was about 8 hours for 6000 wafer starts (real time is more than 3 weeks). This time is comparatively small to the other existing models in use by the manufacturing team. However, the coming section has further analysis to reduce the simulation running times.

Table 5.2: Results of model verification

No.	Scenario	Simulation Results (Hours)	Actual Data (Hours)
1	1 product, 13 layers, 3000 wafer starts, random scheduling	330.8459	291.3205
2	1 product, 13 layers, 6000 wafer starts, random scheduling	542.6478	570.8261
3	5 product, 5 layers, 3000 wafer starts, random scheduling	137.4299	156.0054
4	10 products, 1 layer, 4000 wafer starts, random scheduling	61.6568	74.12381
5	10 products, 5 layers, 4000 wafer starts, random scheduling	89.78155	98.75971
6	15 products, 1 layer, 3750 wafer starts, random scheduling	124.8965	145.7323
7	15 products, 1 layer, 6000 wafer starts, random scheduling	82.45752	98.70811
8	25 products, 2 layers, 6250 wafer starts, random scheduling	81.92122	80.24792

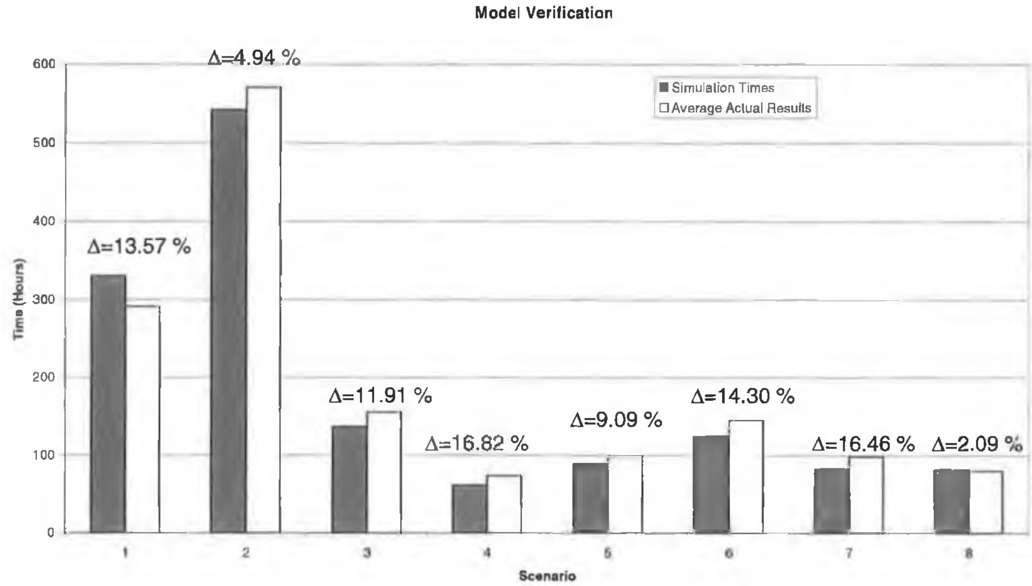


Figure 5.19: Comparison between simulation results vs. actual results of average TPT

The information in Table 5.2 is shown in graphical form in Figure 5.19 and this was considered to mimic production data closely by the manufacturing team.

* *Remarks on verification phase:*

The error (variation) in the preliminary results was slightly high in some scenarios. In fact, this variation happened due to:

- Shortage of clean data in some scenarios
- Simplifications and assumptions were assumed by manufacturing team to handle the difficulties to collect actual floor data in some cases
- Unscheduled maintenance yields
- High priority lots (in actual production)
- Random lot selection.
- Intelligent-agent module is off.

However, the preliminary results were considered acceptable and the model has been verified. The latter sections in this chapter compare the outputs of the model with the actual floor data.

5.5.4 Simulation Time Trade-off

In general, the quality of simulation models can be measured in at least two dimensions [171]:

(1) How close the model mimics the real system if it can be measured?
(2) How much computer time is required to solve problems or obtain results?

Due to wide differences in software, platform, problem size, experimental design and reporting, it is very difficult to compare the performance of different simulation models directly. However, the simulation execution time is still one measure of the implementation success of the simulation technique that should be considered. It is a compromise situation between model details and efficiency in running time. The model running time reaches 8 hours for 6000 wafers start. The actual production time (real time) for processing 6000 wafers can be more than three weeks, and hence the simulation time was acceptable. To enable further investigations of several scenarios and sensitivity analysis, shorter run time of the model is required.

- ***Model modifications***

The model has included the flexible manufacturing cell model in each tool. That means when the model starts, based on the hierarchical approach, it calls the FMC module. The FMC simulation model had simulated the process wafer by wafer and the time step resolution was in seconds while, in this IPS model (toolset), the resolution requirements are lots instead of wafers and hours instead of seconds.
To improve the speed of calculation the original FMC module was replaced with another hierarchical block that functions with less detail. First the data collected from FMC model and fed into spreadsheet file. A statistical analysis of the data shows that the normal distribution is the best fitting for the average processing times per lot. The normal distributions of the processing times with predefined mean and standard deviation were saved into a new spreadsheet file (input file).

The IPS model reads the processing times from the input spreadsheet as shown in Figure 5.20.

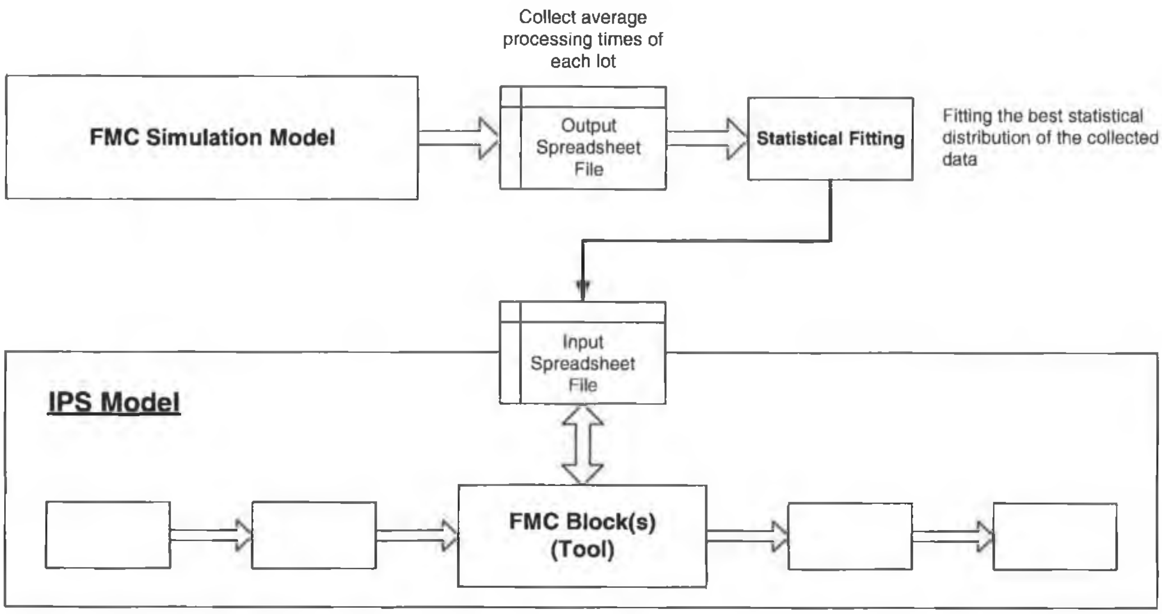


Figure 5.20: The model modification to reduce simulation run time

The trade-off between the simulation run time and the level of detail in the model is a crucial issue and time-consuming process. Guaranteeing the finding of an optimum solution is nolonger a possible choice for complex systems, however obtaining a satisfying solution is the new target. The modified model can run 6000 wafer starts in <u>*less than 4 minutes*</u>, approximately 160 times the reduction in simulation run time. The model was then re-verified and re-validated.

- ***Model re-verification***

The outputs for the new model were verified using the old model scenarios. Random scenarios were selected to examine the new model and compare results between the old model and the new one as shown in Table 5.3. The tool buffer size was assumed to be five lots and number of layers is 13 layers for every lot. The deviation in percent between the old model and the modified one is presented in Figure 5.21 and Figure 5.22. The order of the experiment is the only difference between the two figures. It is clear that for increasing wafer starts and increasing product-mix, the deviation between the two models increases slightly.

Table 5.3: the experiment matrix

Experiment	Product-Mix	Wafer Starts	Experiment	Product-Mix	Wafer Starts
1	1	2000	17	10	6000
2	1	4000	18	15	1125
3	1	5000	19	15	1875
4	1	5000	20	15	4125
5	5	1000	21	15	5250
6	5	1000	22	15	5250
7	5	2000	23	20	1000
8	5	2000	24	20	3000
9	5	6000	25	20	3000
10	5	6000	26	20	4000
11	10	2000	27	20	5000
12	10	2000	28	20	6000
13	10	3000	29	25	3125
14	10	4000	30	25	4375
15	10	5000	31	25	6000
16	10	5000			

On close examination of Figure 5.21 and Figure 5.22, it becomes apparent that the effect of applying the modified model is reasonable as the mathematical average of error between the two models averages 4% of the performance measure. The manufacturing team were satisfied about this deviation as they considered it small compared to the benefits of having quick outputs of the simulation model. The deviation mean was 4.128% and standard deviation is 1.568 %, this supports our conjecture that applying the modified model will save time and the error is traceable and detectable. A statistical software package was used to evaluate the deviation (Qualitek 4.0, Nutek Inc. [192]).

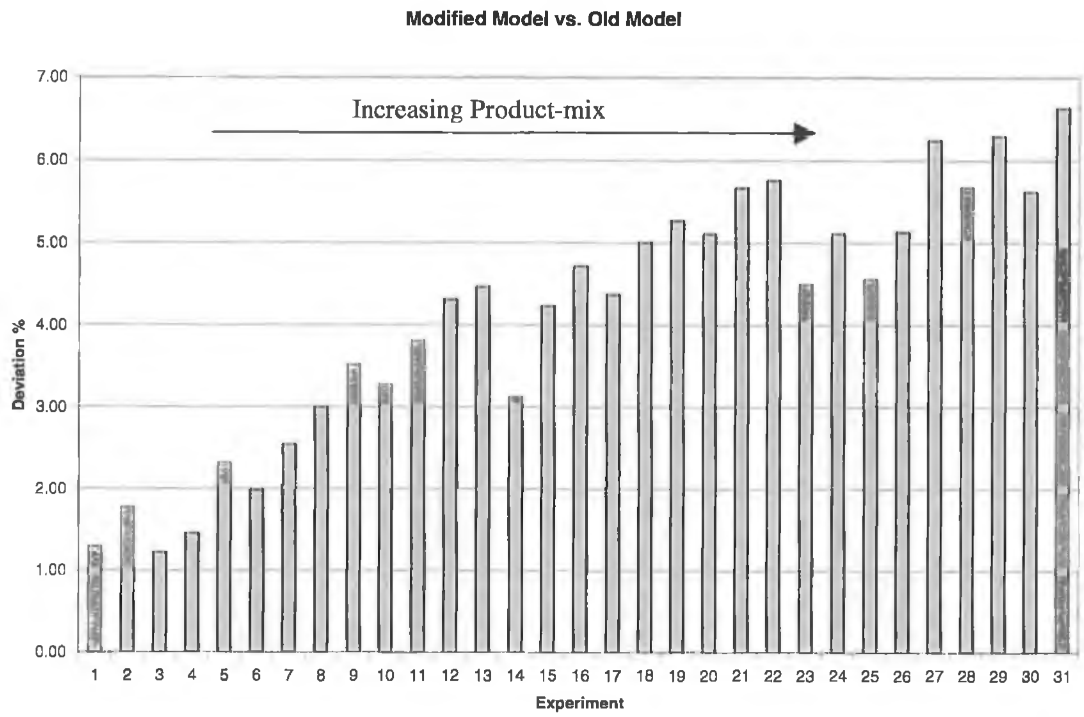


Figure 5.21: The deviation percentage between the two models (order of experiment based on product-mix)

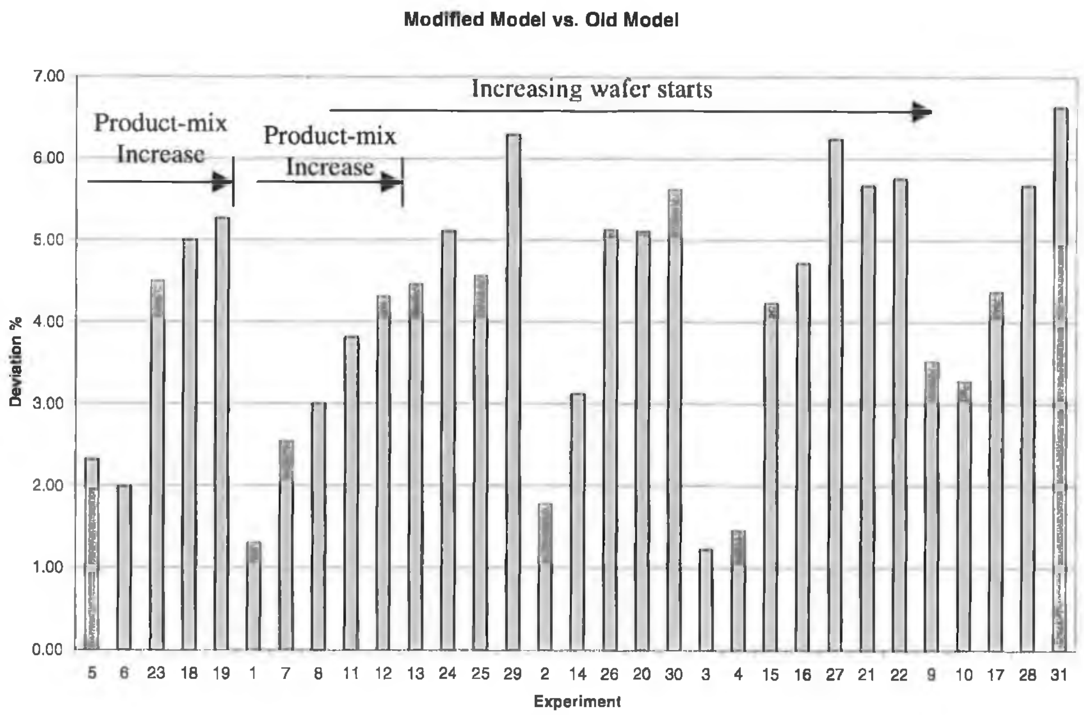


Figure 5.22: The deviation percentage between the two models (order of experiment based on wafers start)

5.6 Experimentation

Experiments will be much simpler if they can always have only one factor influencing the output [179]. Unfortunately, given the complexity of semiconductor manufacturing and in particular the photolithography area the challenge is to decide how many parameters to select to study and how many levels should be set for each? The answers of all these questions come into the design of experiments, which already was explained in chapter four, sec 4.7.

5.6.1 Experiment Repetitions

There are generally two types of experimental error: between-experiment error and within-experiment error. Between-experiment error is the error in the results associated with different setups. Within-experiment error is the error associated with repeating experiments (repetitions) under the same conditions [179]. The way to keep both these types of error to a minimum is to conduct experiments in an efficient manner.

The repetition of experiments is an important issue to be considered in order to obtain accurate output especially when there is more than one objective that a product or process is expected to satisfy. Moreover, a stochastic element in the manufacturing environment is quite common in many industrial clusters. Each experiment has to use the same number or range of repetitions under the same running condition in order to evaluate the results; five repetitions have been for each experiment. For comparison purposes, the average is the most common criterion as long the dispersion is not significant.

The standard deviation and range of variation in the results were calculated. The module considers quality characteristics, which selected to be the smaller is better with target of zero. The results distribution was Normal distribution curve with standard deviation in TPT of 5.641 hours. The manufacturing team has considered the dispersion of preliminary output acceptable, taking into account the stochastic nature of the production system.

5.6.2 Process Parameters

The manufacturing team has selected the following parameters (*factors*):

1. Wafer starts per product (WS)

2. Product-mix (PM)
3. Tool Buffer Size (BS)
4. Number of Layers (NL)
5. Scheduling agent (SA)

The simulation model was developed to focus on the effect of these parameters on the defined objective functions. The performance measure of interest is throughput time (total completion time). Sensitivity analysis has included more performance criteria that give support to decision-making.

Each of the assigned parameters has been set to a different level of operation based on the manufacturing team recommendations, Table 5.4.

Table 5.4: Factors assumed and their levels

Factor	***Factor Level***	***Factor-level details***
Wafer starts(WS)	1	2250
	2	3750
Total number of wafers at the beginning of production	3	4500
	4	6000
Product-mix (PM)	1	1
	2	5
Number of products	3	10
	4	15
Tool Buffer Size (BT)	1	1
	2	3
Size of the buffer in front of each tool	3	5
	4	8
Number of Layers (NL)	1	5
	2	8
Number of layers required to be processed	3	10
	4	13
Scheduling-agent based (SA)	1	0
Either random scheduling or intelligent-agent based	2	1

The wafer starts was assumed to be fixed per each product, for example, if there is 1000 wafer starts and the product-mix is 5, that means there is 200 wafers of each product in this scenario. The product-mix can be set for four different levels 1, 5, 10, 15 products at the beginning of the production.
The buffer size in front of each tool in the toolset is affected using one of the following four sizes: 1, 3, 5, 8. The effect of varying the number of layers (four different levels) on the tool performance, was one of the experimental objectives. Finally, the scheduling-agent based can be either on or off. In other words, there is no systematic approach in lots scheduling for actual production rather the experience of experts working on the shop floor.

5.6.3 Taguchi Methodology

Taguchi experimental design procedures have been applied to the experiments in chapter four and it allowed the team to understand the behavior of the flexible manufacturing tools under different production scenario. Therefore the same procedure has also been used here. Experimental matrices were used based on the standard orthogonal arrays, which allow the simultaneous effect of the selected process parameters to be studied efficiently. The purpose of conducting orthogonal experiments is twofold:

1. To determine the factor combinations that will optimize a defined objective function (i.e., to determine the optimal level for each factor)
2. To establish the relative significance of individual factors in terms of their effects on the objective function.

Quality characteristic (QC) is considered as the sense of desirability. There could be three different types of QC as mentioned earlier in chapter five: smaller is better (QC=S), bigger is better (QC=B), and nominal is best (QC=N).

- *Smaller the better*

  Here, as the major objective function is throughput time, the smaller the better magnitude of the results is always preferred over the others. The theoretical target is the minimum, or in other words, the lowest achievable value that can be obtained.

Summary statistics have used average throughput time (TPT) per lot (to normalize the output) as the objective function for matrix experiments. S/N ratios could not be measured in this study due to the scale of the output.
The overall mean value of average TPT for the experimental region is shown in the formula below;

$$m = \frac{\left[\sum_{i=1}^{n} \eta_i\right]}{n}$$

where;

η_i= observed *S/N* ratio for the i^{th} orthogonal experiment,

n = number of experiments performed,

i = experiment number,

m= overall mean value of η

- *Standard Orthogonal Array*

The selection of the orthogonal array is based on the number of factors and their levels. Here, there are five factors with different levels. There is one two-level factor and four four-level factors. The standard orthogonal array L_{32} $(2^1, 4^9)$ modified was selected and then the factors assigned to the columns. Table 5.5 shows the standard for of L_{32} orthogonal array as Taguchi designed.
This array enables the simultaneous consideration of 1 factor (at two levels), and 9 factors (at four levels). In the present case only one factor (at two levels) and four factors (at four levels) are considered, so the first five columns of L_{32} modified are used with the remaining columns being excluded without affecting the orthogonality of the matrix.
To study the impact of the factors listed earlier within the photolithography toolset considered, the orthogonal array has been adjusted based on number of factors used and the levels dedicated. Table 5.6 distributes the factor levels into the orthogonal array form for experiments.
The arithmetic mean equation and other equations of the Taguchi methodology were similar to these used in chapter four.

Table 5.5: Standard L_{32} $(1^2, 4^9)$ modified (mixed level) orthogonal array

Exp. No.	Column									
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3	3	3
4	1	1	4	4	4	4	4	4	4	4
5	1	2	1	1	2	2	3	3	4	4
6	1	2	2	2	1	1	4	4	3	3
7	1	2	3	3	4	4	1	1	2	2
8	1	2	4	4	3	3	2	2	1	1
9	1	3	1	2	3	4	1	2	3	4
10	1	3	2	1	4	3	2	1	4	3
11	1	3	3	4	1	2	3	4	1	2
12	1	3	4	3	2	1	4	3	2	1
13	1	4	1	2	4	3	3	4	2	1
14	1	4	2	1	3	4	4	3	1	2
15	1	4	3	4	2	1	1	2	4	3
16	1	4	4	3	1	2	2	1	3	4
17	2	1	1	4	1	4	2	3	2	3
18	2	1	2	3	2	3	1	4	1	4
19	2	1	3	2	3	2	4	1	4	1
20	2	1	4	1	4	1	3	2	3	2
21	2	2	1	4	2	3	4	1	3	2
22	2	2	2	3	1	4	3	2	4	1
23	2	2	3	2	4	1	2	3	1	4
24	2	2	4	1	3	2	1	4	2	3
25	2	3	1	3	3	1	2	4	4	2
26	2	3	2	4	4	2	1	3	3	1
27	2	3	3	1	1	3	4	2	2	4
28	2	3	4	2	2	4	3	1	1	3
29	2	4	1	3	4	2	4	2	1	3
30	2	4	2	4	3	1	3	1	2	4
31	2	4	3	1	2	4	2	4	3	1
32	2	4	4	2	1	3	1	3	4	2

Table 5.6: Experimental table details

Exp.	SA	PM	WS	BT	NL	Exp.	SA	PM	WS	BT	NL
1	0	1	2250	1	5	**17**	1	1	2250	8	5
2	0	1	3750	3	8	**18**	1	1	3750	5	8
3	0	1	4500	5	10	**19**	1	1	4500	3	10
4	0	1	6000	8	13	**20**	1	1	6000	1	13
5	0	5	2250	1	8	**21**	1	5	2250	8	8
6	0	5	3750	3	5	**22**	1	5	3750	5	5
7	0	5	4500	5	13	**23**	1	5	4500	3	13
8	0	5	6000	8	10	**24**	1	5	6000	1	10
9	0	10	2250	3	10	**25**	1	10	2250	5	10
10	0	10	3750	1	13	**26**	1	10	3750	8	13
11	0	10	4500	8	5	**27**	1	10	4500	1	5
12	0	10	6000	5	8	**28**	1	10	6000	3	8
13	0	15	2250	3	13	**29**	1	15	2250	5	13
14	0	15	3750	1	10	**30**	1	15	3750	8	10
15	0	15	4500	8	8	**31**	1	15	4500	1	8
16	0	15	6000	5	5	**32**	1	15	6000	3	5

5.6.4 Matrix Experiment Analysis

Simulation experiments are performed using Extend (*Imaginethat Inc.* [177]), into which C++ code is linked to capture the customization incorporated into the models. All wafers are assumed available at the start of the simulation run (i.e., wafer arrivals are not stochastically generated); although part arrivals into the system are dependent on signals from the cell or first operating unit.

Table 5.7: Matrix experiment simulation results

Experiment #	Avg. Throughput Time *(Hours)*	Average Throughput per Lot *(Hours)*	Experiment #	Avg. Throughput Time *(Hours)*	Average Throughput per Lot *(Hours)*
1	58.295	0.648	17	55.844	0.620
2	167.951	1.120	18	99.974	0.666
3	247.951	1.378	19	206.235	1.146
4	319.554	1.331	20	300.290	1.251
5	96.287	1.070	21	65.383	0.726
6	82.842	0.552	22	78.020	0.520
7	276.578	1.537	23	239.657	1.331
8	351.985	1.467	24	301.123	1.255
9	131.195	1.458	25	92.882	1.032
10	287.510	1.917	26	241.343	1.609
11	130.399	0.724	27	96.342	0.535
12	246.545	1.027	28	200.779	0.837
13	153.090	1.701	29	103.442	1.149
14	233.906	1.559	30	219.610	1.464
15	238.114	1.323	31	198.405	1.102
16	206.597	0.861	32	182.014	0.758

The processing times and product mix are assumed to be predefined before the simulation run.

Finally, identical experimental testing conditions for each simulation scenario are ensured using the method of common random numbers. The results obtained from the simulation model based on the matrix experiment are detailed in Table 5.7.

- ***ANalysis Of Means (ANOM)***

Based on the analysis of means, the optimum/near optimum level for each factor resulting from the matrix of experiments is shown italicized in the throughput column of Table 5.8. Accordingly, the predicted factor level combination that should optimize (i.e., minimize) the throughput time per lot is SA2, PM1, WS1, BT3, and NL1, which is interpreted as scheduling with the wafer starts = 2250, the product-mix is one product, The tool buffer size is five lots, and the number of layers is five. The predicted best setting does not match to any of the rows in the orthogonal matrix used.

Table 5.8: Factor main effects for matrix experiment simulation study results

Factor-level main effects	*Applicable formula*	*Main effect value TPT per lot*
m_{PM1}	$(\eta_1+\eta_2+\eta_3+\eta_4+\eta_{17}+\eta_{18}+\eta_{19}+\eta_{20})/8$	***1.02004***
m_{PM2}	$(\eta_5+\eta_6+\eta_7+\eta_8+\eta_{21}+\eta_{22}+\eta_{23}+\eta_{24})/8$	1.057251
m_{PM3}	$(\eta_9+\eta_{10}+\eta_{11}+\eta_{12}+\eta_{25}+\eta_{26}+\eta_{27}+\eta_{28})/8$	1.142369
m_{PM4}	$(\eta_{13}+\eta_{14}+\eta_{15}+\eta_{16}+\eta_{29}+\eta_{30}+\eta_{31}+\eta_{32})/8$	1.239765
m_{SW1}	$(\eta_1+\eta_5+\eta_9+\eta_{13}+\eta_{17}+\eta_{21}+\eta_{25}+\eta_{29})/8$	***1.050581***
m_{SW2}	$(\eta_2+\eta_6+\eta_{10}+\eta_{14}+\eta_{18}+\eta_{22}+\eta_{26}+\eta_{30})/8$	1.175963
m_{SW3}	$(\eta_3+\eta_7+\eta_{11}+\eta_{15}+\eta_{19}+\eta_{23}+\eta_{27}+\eta_{31})/8$	1.134501
m_{SW4}	$(\eta_4+\eta_8+\eta_{12}+\eta_{16}+\eta_{20}+\eta_{24}+\eta_{28}+\eta_{32})/8$	1.098379
M_{BT1}	$(\eta_1+\eta_5+\eta_{10}+\eta_{14}+\eta_{20}+\eta_{24}+\eta_{27}+\eta_{31})/8$	1.167132
M_{BT2}	$(\eta_2+\eta_6+\eta_9+\eta_{13}+\eta_{19}+\eta_{23}+\eta_{28}+\eta_{32})/8$	1.112854
M_{BT3}	$(\eta_3+\eta_7+\eta_{12}+\eta_{16}+\eta_{18}+\eta_{22}+\eta_{25}+\eta_{29})/8$	***1.021268***
M_{BT5}	$(\eta_4+\eta_8+\eta_{11}+\eta_{15}+\eta_{17}+\eta_{21}+\eta_{26}+\eta_{30})/8$	1.158171
M_{NL1}	$(\eta_1+\eta_6+\eta_{11}+\eta_{16}+\eta_{17}+\eta_{22}+\eta_{27}+\eta_{32})/8$	***0.652441***
M_{NL3}	$(\eta_2+\eta_5+\eta_{12}+\eta_{15}+\eta_{18}+\eta_{21}+\eta_{28}+\eta_{31})/8$	0.983932
M_{NL4}	$(\eta_3+\eta_8+\eta_9+\eta_{14}+\eta_{19}+\eta_{24}+\eta_{25}+\eta_{30})/8$	1.344714
M_{NL5}	$(\eta_4+\eta_7+\eta_{10}+\eta_{13}+\eta_{20}+\eta_{23}+\eta_{26}+\eta_{29})/8$	1.478338
M_{SA1}	$(\eta_1+\eta_2+\eta_3+\eta_4+\eta_5+\eta_6+\eta_7+\eta_8+\eta_9+\eta_{10}+\eta_{11}+\eta_{12}+\eta_{13}+\eta_{14}+\eta_{15}+\eta_{16})/16$	1.229493
M_{SA2}	$(\eta_{17}+\eta_{18}+\eta_{19}+\eta_{20}+\eta_{21}+\eta_{22}+\eta_{23}+\eta_{24}+\eta_{25}+\eta_{26}+\eta_{27}+\eta_{28}+\eta_{29}+\eta_{30}+\eta_{31}+\eta_{32})/16$	***1.000219***

The main effects of each factor level are shown in Figure 5.23. The near optimum value for each factor can be easily identified as the level that results the lowest value and hence the lowest points are the best. Note that the prediction of the near optimum factor level combination is conditioned by the additivity assumption. To justify the validity of this assumption, a verification experiment using the near optimum combination needs to be carried out. The ANOM plot shown in Figure 5.23 reveals the relative magnitude of effects of factors on the throughput time per lot. The number of layers (NL) is seen to affect the throughput time the most, followed by applying intelligent scheduling of lots. The effect of both factors is seen to be relatively less pronounced than the other parameters based on TPT criterion.

However, a better feel for the relative effects is obtained by conducting the analysis of variance. ANOVA is also needed for estimating the error variance for the factor effects and the variance of the prediction error, which provide justification for the additivity assumption. Moreover, the experiments may be extended to evaluate the effect of the significant factors on different performance criteria.

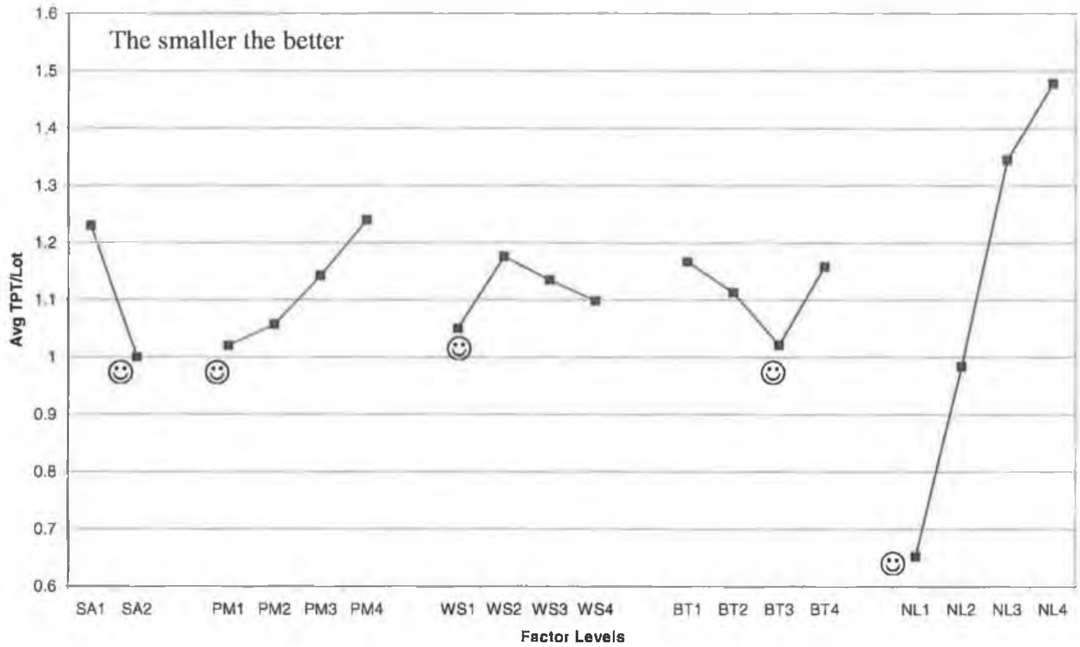


Figure 5.23: Analysis of means (ANOM) plot of main effects

- ***<u>ANalysis Of VAriance (ANOVA)</u>***

The main formulas used in conducting the ANOVA in chapter five were used again in this analysis, and the results are shown in Table 5.9. The error variance (σ_e^2), calculated and found to be 0.1419.

$$\sigma_e^2 = 0.1419\ (\mathrm{Hr})^2$$

Taguchi methodology suggests using the F ratio resulting from ANOVA mainly to establish the relative magnitude of the effect of each factor on the objective function and to estimate the error variance. From the ANOVA tableau, Table 5.9, the most significant factors are the number of layers and the application of intelligent scheduling of lots.

Table 5.9: ANOVA using the simulation results

Factor	***DOF***	***SSB***	***SSB/DOF***	F_{cal}	$F_{tabl.}$
SA	1	0.42136	0.42136	*23.48*	4.4139
PM	3	0.22913	0.07638	4.26	3.1599
WS	3	0.06818	0.02273	1.27	3.1599
BT	3	0.10688	0.03563	1.99	3.1599
NL	3	3.32899	1.10966	*61.83*	3.1599
Error	18	0.32304	0.01795		
Total	31	4.47758			

The product-mix is seen to be important, followed by wafer starts and tool buffer size (taking into consideration that F-tabulated (at α=0.5, i.e., $F_{0.5,\ 3,\ 18}$= 3.1599 and $F_{0.5,\ 1,\ 18}$ = 4.4139)).

5.6.5 Testing of Additivity

A verification experiment, similar to chapter five, was performed with the near optimal factor combination (SA2, PM1, WS1, BT3, and LN1). The observed optimal/near optimal throughput time per lot was 0.5925 hours.

The following equation was then used to predict the optimum/near optimum performance measure value:

$$\begin{aligned}\eta_{pre.opt} &= m+(m_{LN1}-m)+(m_{SA2}-m)+(m_{PM1}-m)\\ &= 1.1149+(0.6524-1.1149)+(1.00022-1.1149)+(1.02004-1.1149)\\ &= 0.44286\ Hr\end{aligned}$$

Note that $\eta_{pre.opt}$ is calculated using only significant optimal (main effects) factor-level values. The prediction error then becomes

$$\begin{aligned}\textit{Prediction error} &= \eta_{obs.opt} - \eta_{pre.opt}\\ &= 0.5925 - 0.44286\\ &= 0.15\ Hr\end{aligned}$$

The variance of prediction error σ_e^2 is calculated using same formulas as in chapter five:

$$\sigma^2_{e\ pred} = 0.0626 Hr^2$$

The corresponding two-standard-deviation confidence limits for the prediction error are $\pm 2\times\sqrt{\sigma^2_{e\ pred}} = \pm 0.5(Hr)$.

The prediction error of 0.15 *hour* happens to be well within the calculated confidence limits, so the additivity assumption is justified.

5.6.6 Results Analysis

The Taguchi methodology has been applied to gain better understanding of the significance of the selected parameters. Taguchi methodology has a remarkable save in number of experiments needed to analyze the performance.

The ANOM plot in Figure 5.23, provides useful insights with regard to the severe impact of increasing the number of layers on the average TPT. It is interesting to note the benefit obtained from using ANOM to give quick and effective indication for the factors. The results recommend 2250 wafer starts which is not possible to set in many cases of actual production. However, the planning team has to keep the wafer starts as minimum as 2250 wafers, if possible. The impact of increasing wafer starts can be easily examined using the model. The tools buffer size has been suggested to be 5 lots per tool.

Based on the ANOVA Table 5.9, the number of layers and application of intelligent scheduling of lots, then product-mix have a significant impact on the throughput time per lot. The effect of the remaining factors has less impact on throughput as an objective function. Testing of additivity has been conducted to verify the experimental design assumptions and confirm the errors are within acceptable limits.

To prevent changes in the number of layers masking the effects of the other parameters the number of layers in all products was fixed at 13 layers for the following experiments.

More experiments were performed to study the effect of other factors on different toolset performance criteria. Sensitivity analyses have been set to provide better understanding of the system performance under various production scenarios.

5.7 Sensitivity Analysis

To evaluate the performance of the toolset using other measures of performance than throughput time, further experiments were conducted. A number of simulation sensitivity analyses were performed. These included experiments to analyze the impact of:

- Increasing product-mix on batch size.
- Changing the Buffer size in front of each tool on RtC.
- Changing selecsted parameters on RtC using ANOM.

a) Effect of product-mix increase on batch size for toolset

Product-mix increase is one major problem that faces manufacturing team and has an impact on the toolset performance. The objective here was to find the impact of increasing the number of products on the batch size needed to meet planned production. A combination of experiments (36 experiments) was performed and the number of wafers per product is shown in Table 5.10.

Table 5.10: The experiment matrix

	WS	**Wafers Start**					
	PM	1000	2000	3000	4000	5000	6000
Product-Mix	1	1000	2000	3000	4000	5000	6000
	5	200	400	600	800	1000	1200
	10	100	200	300	400	500	600
	15	75	125	200	275	350	400
	20	50	100	150	200	250	300
	25	50	75	125	175	200	240

The buffer size was fixed at five lots and the number of layer is 13 layers. The results have been handled based on counting the total number of changes occurred in the toolset, and the changes per batch size has been determined as well. Figure 5.24, shows the variation in product batch size due to increase in product-mix. The analysis showed that for more than 10 products in the production, the batch size tends to be consistent value, about two. The manufacturing team of the industry partner has verified the simulation model results.

These results suggest several directions for more experiments to study the impact of changing the buffer size in front of the tools (FMCs) on the reticle changes and the batch size. It has been seen that for one product, the number of changes tends to be constant and therefore one product was not included.

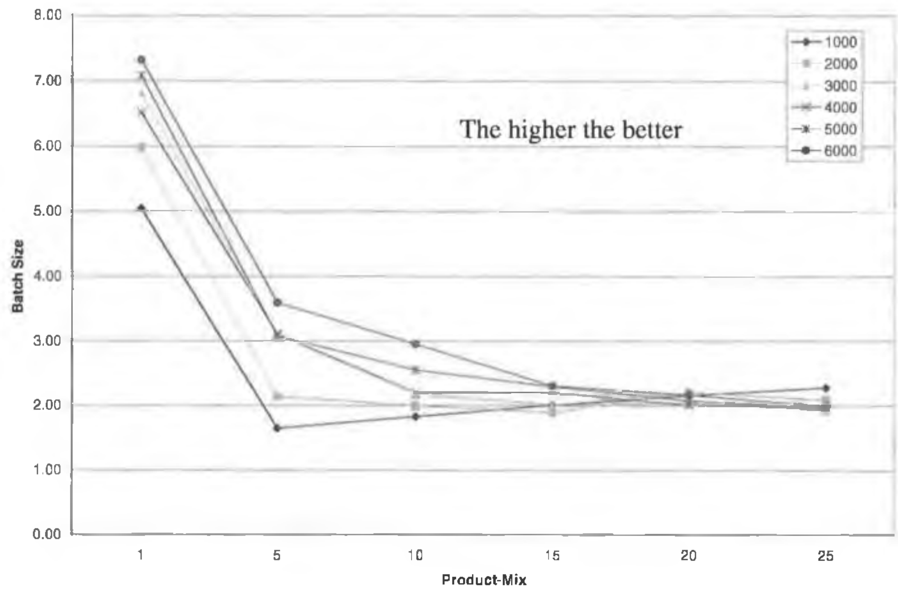


Figure 5.24: the impact of changing product-mix on toolset batch size

b) <u>Change Tool Buffer size on reticles changes</u>

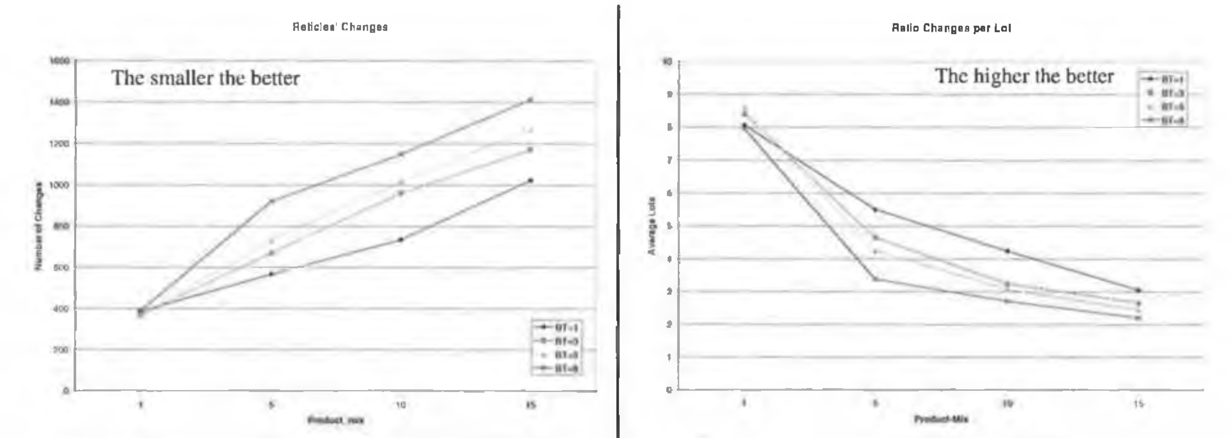
Product-mix considers the main independent variables for this analysis, as buffer size has been examined on four different levels (1, 3, 5, 8). The performance criteria were RtC and the batch size per lot or change ratio. A set of experiments (16 experiments) was performed as shown in Table 5.11.

Table 5.11: The experiment matrix

Exp	PM	BT	Total Reticle Changes	Ratio (Lots:Changes)	Exp	PM	BT	Total Reticle Changes	Ratio (Lots:Changes)
1	1	1	387	8	9	1	5	363	9
2	5	1	567	6	10	5	5	734	4
3	10	1	735	4	11	10	5	1015	3
4	15	1	1023	3	12	15	5	1270	2
5	1	3	371	8	13	1	8	392	8
6	5	3	671	5	14	5	8	920	3
7	10	3	960	3	15	10	8	1149	3
8	15	3	1170	3	16	15	8	1413	2

The experiments have assumed all the lots need to process 13 layers and the wafer starts 6000 wafers. The results of the experiments are summarized in Figure 5.25.

The most prominent results is that this analysis showed consistently the ratio to average lots per changes turned to between two and three all the cases when the product-mix increases. The buffer size has no significant effect on RtC, the trend is almost same, that gives the manufacturing team better understanding to the WIP issue and consider its impact on toolset performance for more comprehensive floor configurations.



A) Reticles' changes plot B) Average lots to change Ratio

Figure 5.25: Summary of tool buffer size sensitivity analysis

c) <u>Analysis of means for changing tools' reticle</u>

A set of experiments was designed to study the RtC based on analysis of means as primary analysis approach and analysis of variance as a secondary approach. The ANOM considers the smaller the better type in the quality characteristic while investigating RtC. The experiments have been performed based on orthogonal array L-32 modified with four factors to study as number of layers (LN) has been excluded because its significance was already established. Table 5.12 shows the experiment matrix and the results.

Based on the ANOM, the best level for each factor resulting from matrix experiment is shown italicized in the throughput column of Table 5.13. The tool buffer size has been set to one lot, which contradicting with the ANOM in the earlier section but buffer tool size has been distinguished as a non significant factor and hence it was acceptable to have some variations in the results.

The ANOM plots shown in Figure 5.26 reveal the relative magnitude of effects by factors on the average number of changes of the mask. The product-mix is seen to affect the objective function the most, followed by applying intelligent scheduling of lots.

Table 5.12: The experiment matrix and results

Exp.	SA	PM	WS	BT	Avg. RtC	Exp.	SA	PM	WS	BT	Avg. RtC
1	0	1	2250	1	71	17	1	1	2250	8	431
2	0	1	3750	3	319	18	1	1	3750	5	666
3	0	1	4500	5	856	19	1	1	4500	3	168
4	0	1	6000	8	664	20	1	1	6000	1	207
5	0	5	2250	1	246	21	1	5	2250	8	240
6	0	5	3750	3	502	22	1	5	3750	5	942
7	0	5	4500	5	387	23	1	5	4500	3	845
8	0	5	6000	8	393	24	1	5	6000	1	601
9	0	10	2250	3	469	25	1	10	2250	5	49
10	0	10	3750	1	687	26	1	10	3750	8	283
11	0	10	4500	8	198	27	1	10	4500	1	796
12	0	10	6000	5	363	28	1	10	6000	3	600
13	0	15	2250	3	209	29	1	15	2250	5	460
14	0	15	3750	1	568	30	1	15	3750	8	614
15	0	15	4500	8	589	31	1	15	4500	1	355
16	0	15	6000	5	460	32	1	15	6000	3	930

Table 5.13: Factor main effects for matrix experiment simulation study results

Factor-level main effects	***Applicable formula***	***Main effect value Avg. CtR***
$\boldsymbol{m_{PM1}}$	$(\eta_1+\eta_2+\eta_3+\eta_4+\eta_{17}+\eta_{18}+\eta_{19}+\eta_{20})/8$	***210***
m_{PM2}	$(\eta_5+\eta_6+\eta_7+\eta_8+\eta_{21}+\eta_{22}+\eta_{23}+\eta_{24})/8$	501
m_{PM3}	$(\eta_9+\eta_{10}+\eta_{11}+\eta_{12}+\eta_{25}+\eta_{26}+\eta_{27}+\eta_{28})/8$	587
m_{PM4}	$(\eta_{13}+\eta_{14}+\eta_{15}+\eta_{16}+\eta_{29}+\eta_{30}+\eta_{31}+\eta_{32})/8$	599
$\boldsymbol{m_{SW1}}$	$(\eta_1+\eta_5+\eta_9+\eta_{13}+\eta_{17}+\eta_{21}+\eta_{25}+\eta_{29})/8$	***372***
m_{SW2}	$(\eta_2+\eta_6+\eta_{10}+\eta_{14}+\eta_{18}+\eta_{22}+\eta_{26}+\eta_{30})/8$	482
m_{SW3}	$(\eta_3+\eta_7+\eta_{11}+\eta_{15}+\eta_{19}+\eta_{23}+\eta_{27}+\eta_{31})/8$	473
m_{SW4}	$(\eta_4+\eta_8+\eta_{12}+\eta_{16}+\eta_{20}+\eta_{24}+\eta_{28}+\eta_{32})/8$	570
$\boldsymbol{M_{BT1}}$	$(\eta_1+\eta_5+\eta_{10}+\eta_{14}+\eta_{20}+\eta_{24}+\eta_{27}+\eta_{31})/8$	***430***
M_{BT2}	$(\eta_2+\eta_6+\eta_9+\eta_{13}+\eta_{19}+\eta_{23}+\eta_{28}+\eta_{32})/8$	443
M_{BT3}	$(\eta_3+\eta_7+\eta_{12}+\eta_{16}+\eta_{18}+\eta_{22}+\eta_{25}+\eta_{29})/8$	513
M_{BT5}	$(\eta_4+\eta_8+\eta_{11}+\eta_{15}+\eta_{17}+\eta_{21}+\eta_{26}+\eta_{30})/8$	511
M_{SA1}	$(\eta_1+\eta_2+\eta_3+\eta_4+\eta_5+\eta_6+\eta_7+\eta_8+\eta_9+\eta_{10}+\eta_{11}+\eta_{12}+\eta_{13}+\eta_{14}+\eta_{15}+\eta_{16})/16$	538
$\boldsymbol{M_{SA2}}$	$(\eta_{17}+\eta_{18}+\eta_{19}+\eta_{20}+\eta_{21}+\eta_{22}+\eta_{23}+\eta_{24}+\eta_{25}+\eta_{26}+\eta_{27}+\eta_{28}+\eta_{29}+\eta_{30}+\eta_{31}+\eta_{32})/16$	***410***

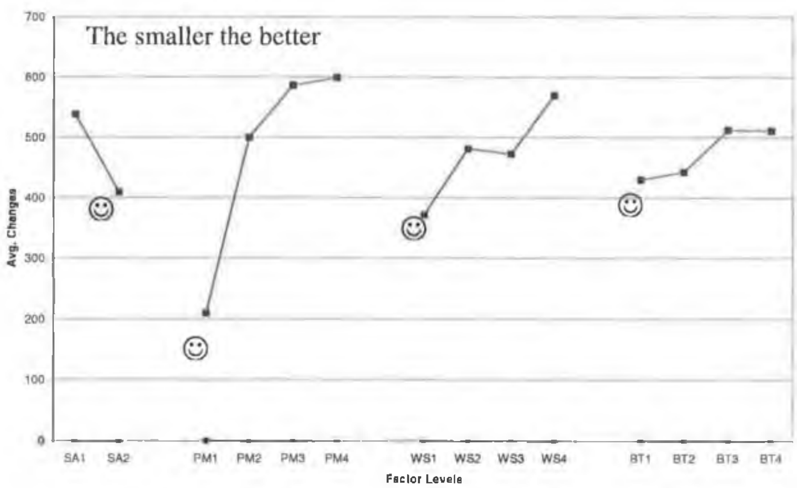


Figure 5.26: Analysis of means (ANOM) plot of main effects

It is worth showing the ANOVA table of this analysis in Table 5.14, indicating the sum of squares added together to estimate the pooled error. The ANOVA reinforces the trend from ANOM plot and highlight the relative magnitude of the effect of the first two factors on the objective function.

Table 5.14: ANOVA using the simulation results

Factor	DOF	SSB	SSB/DOF	$F_{cal.}$	$F_{tabl.}$
SA	1	132612	132613	*3.78*	*3.1599*
PM	3	791575	263858	*7.52*	*4.41139*
WS	3	157629	52543*		
BT	3	46151	15384*		
Error	21	743509	35405		
Total	31	1871476			

*The pooled factors have been taking into consideration

5.8 Advantages of Intelligent Scheduling approach

Additional experiments were conducted to study the impact of intelligent-agent scheduling on the toolset performance. The performance measures considered were tool utilization and RtC according to the manufacturing team priority. Three different levels of wafer starts were used 2250, 4500, and 6000, while product-mix was to 5, 10, and 15 products. Number of layers to be processed assumed to be 13 for all experiments and tool buffer size is five lots. The combinations of these scenarios (18 experiments) are summarized, Table 5.15. A sample of the results is shown in Figures 5.27, 5.28, and 5.29. In the first two scenarios, wafer input level 2250 wafer starts for five different products are the settings for experiments (1 and 10). The output of the model presents all the data required for analysis, for example, Tables A and B in Figure 5.27.

Table 5.15: Summarized simulation experiments

Exp.	*WS*	*PM*	*SA*	*Exp.*	*WS*	*PM*	*SA*
1	2250	5	On	10	2250	5	Off
2	2250	10	On	11	2250	10	Off
3	2250	15	On	12	2250	15	Off
4	4500	5	On	13	4500	5	Off
5	4500	10	On	14	4500	10	Off
6	4500	15	On	15	4500	15	Off
7	6000	5	On	16	6000	5	Off
8	6000	10	On	17	6000	10	Off
9	6000	15	On	18	6000	15	Off

The first row of each table explains the experiment number and the scenario conditions, while the second is the total completion time in hours. The columns have the data about the tool number and its utilization over the running time. Column four has the total number of changes (due to layer, product, or both) for every tool, while last column shows the number of lots which went through the tool in order to reflect the usage of the tool. and was used to verify the model as this number is directly related to the total number of layers to be processed (2250 x 13/25) in this case. Figure 5.27, illustrates two experiment outputs while different scheduling approach has been applied but still the total number of lots same.

The model using the intelligent approach of scheduling lots (SA1) has run same conditions for same experiment matrix to compare the output. The results explore the advantages of adding the intelligent lot scheduling for tool utilization as well as RtC.

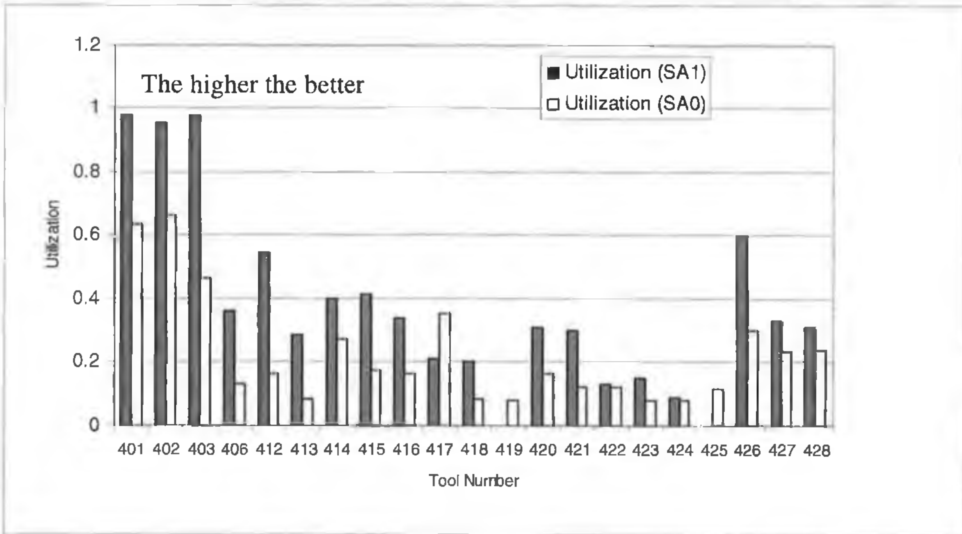
Examining the following Figures 5.27 - 5.29, it becomes apparent that the benefits of applying the intelligent approach for scheduling the lots in the photolithography toolset. This is particularly clear in the reduction of TPT and RtC at each tool as well as the increase of tool utilization.

Table A1

Exp.1	WS 2250	PM 5	SA 1	
TPT=	121.5294	Hours		
Number	Tool Number	Utilization	Number of Changes	Number of Lots
1	401	0.980428	13	60
2	402	0.957596	14	67
3	403	0.979028	13	60
6	406	0.360035	6	49
12	412	0.544403	37	143
13	413	0.289349	35	86
14	414	0.39983	24	48
15	415	0.414981	34	57
16	416	0.338889	29	48
17	417	0.209764	24	48
18	418	0.201712	19	58
19	419	0	0	0
20	420	0.312092	30	47
21	421	0.302607	17	67
22	422	0.132665	12	30
23	423	0.152021	8	44
24	424	0.090106	6	27
25	425	0	0	0
26	426	0.597291	48	128
27	427	0.331613	27	63
28	428	0.312593	18	40
		Total Number =	414	1170
		Average =	19.71429	55.71429

Table B1

Exp.10	WS 2250	PM 5	SA 0	
TPT=	162.4638	Hours		
Number	Tool Number	Utilization	Number of Changes	Number of Lots
1	401	0.63537	33	60
2	402	0.662361	31	90
3	403	0.465273	27	60
6	406	0.131065	7	30
12	412	0.164707	22	66
13	413	0.082436	30	41
14	414	0.271201	28	45
15	415	0.172203	18	45
16	416	0.164427	22	45
17	417	0.351192	85	148
18	418	0.082529	40	41
19	419	0.078566	31	40
20	420	0.166024	18	45
21	421	0.124609	17	45
22	422	0.121581	14	45
23	423	0.079264	13	36
24	424	0.080123	11	36
25	425	0.117369	38	54
26	426	0.302808	39	90
27	427	0.237011	33	63
28	428	0.241945	28	45
		Total Number =	585	1170
		Average =	27.85714	55.71429



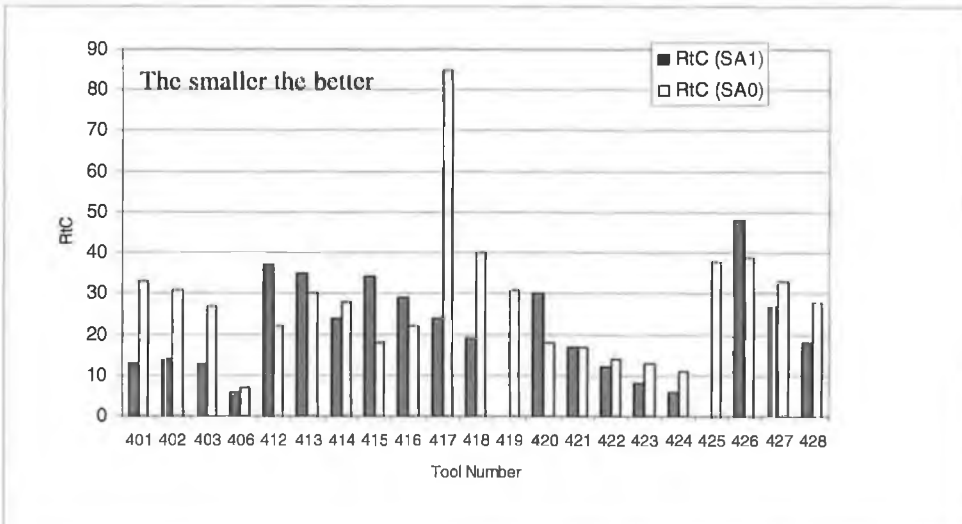


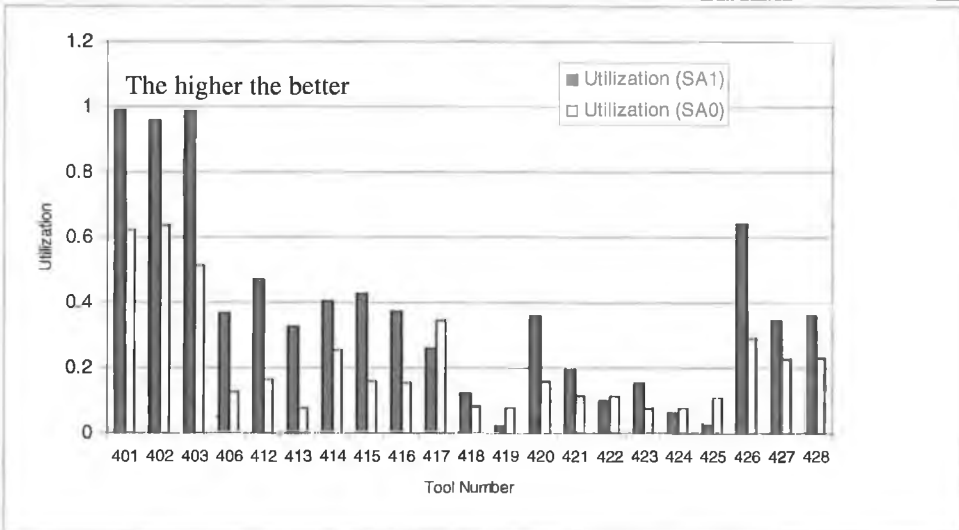
Figure 5.27: Summary of experiments (1 and 10) output

Table A2

Exp.5 TPT Number	WS 4500 256.7341 Tool Number	PM 10 Hours Utilization	SA 1 Number of Changes	Number of Lots
1	401	0.989276	23	120
2	402	0.957207	28	148
3	403	0.98857	23	120
6	406	0.366595	10	86
12	412	0.472874	56	270
13	413	0.327493	37	217
14	414	0.405767	46	94
15	415	0.426778	58	105
16	416	0.373158	50	91
17	417	0.25729	41	94
18	418	0.124987	48	90
19	419	0.022241	9	15
20	420	0.360021	46	90
21	421	0.195453	22	104
22	422	0.100376	16	54
23	423	0.155249	39	101
24	424	0.063053	7	39
25	425	0.02819	6	18
26	426	0.638956	59	290
27	427	0.346007	64	107
28	428	0.358679	43	87
		Total Number =	731	2340
		Average =	34.80952	111.4286

Table B2

Exp.14 TPT Number	WS 4500 314.066 Tool Number	PM 10 Hours Utilization	SA 0 Number of Changes	Number of Lots
1	401	0.620718	64	120
2	402	0.635536	73	180
3	403	0.514856	61	120
6	406	0.125712	13	60
12	412	0.163207	63	132
13	413	0.077673	36	81
14	414	0.253569	44	90
15	415	0.156966	52	90
16	416	0.155959	39	90
17	417	0.343275	120	297
18	418	0.079804	29	81
19	419	0.077062	42	81
20	420	0.157627	35	90
21	421	0.113605	34	90
22	422	0.114569	31	90
23	423	0.077726	35	72
24	424	0.077911	17	72
25	425	0.110044	67	108
26	426	0.29269	98	180
27	427	0.226645	52	126
28	428	0.231301	53	90
		Total Number =	1058	2340
		Average =	50.38095	111.4286



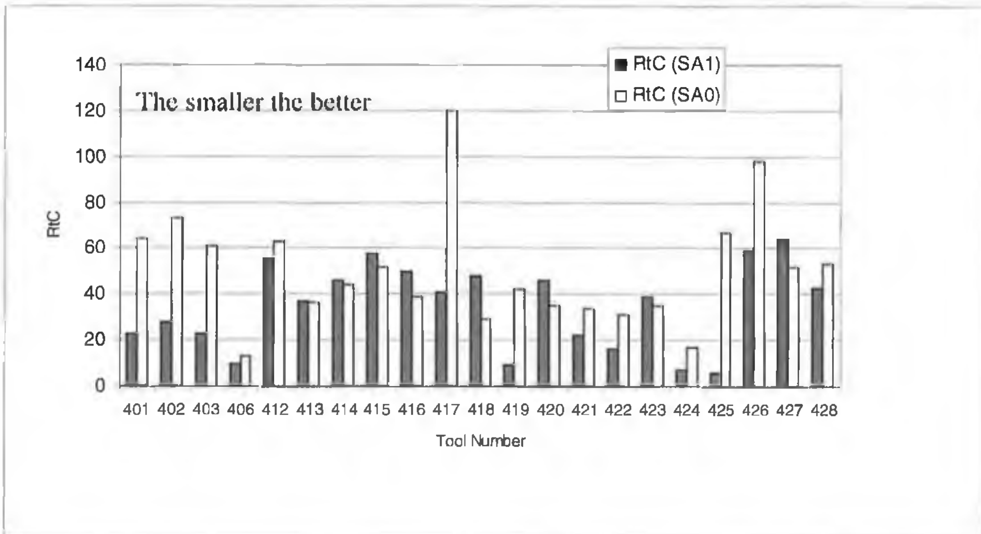


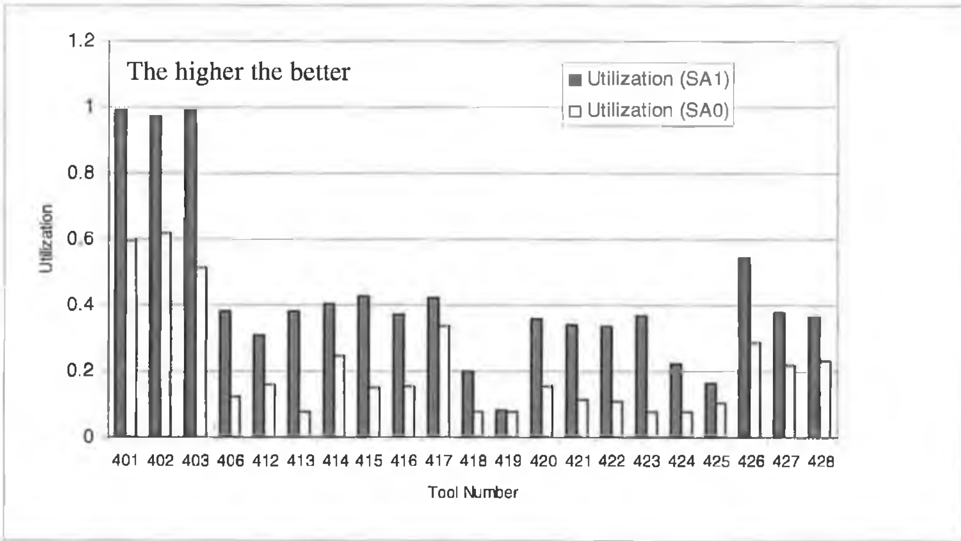
Figure 5.28: A summary of experiments (5 and 14) output

Table A3

Exp.9	WS 6000	PM 15	SA 1	
TPT	220.6646	Hours		
Number	Tool Number	Utilization	Number of Changes	Number of Lots
1	401	0.991696	33	160
2	402	0.974358	39	179
3	403	0.99196	33	161
6	406	0.383972	18	124
12	412	0.310922	32	139
13	413	0.381755	76	209
14	414	0.403615	80	126
15	415	0.427811	73	134
16	416	0.374655	66	123
17	417	0.422689	73	202
18	418	0.199764	62	130
19	419	0.083974	23	63
20	420	0.361192	58	119
21	421	0.340455	45	159
22	422	0.335641	52	168
23	423	0.366777	66	181
24	424	0.221461	75	139
25	425	0.161704	56	108
26	426	0.544183	75	263
27	427	0.376464	65	118
28	428	0.365956	56	115
		Total Number =	1156	3120
		Average =	55.04762	148.5714

Table B3

Exp. 18	WS 6000	PM 15	SA 0	
TPT	407.1965	Hours		
Number	Tool Number	Utilization	Number of Changes	Number of Lots
1	401	0.595282	89	160
2	402	0.620416	140	240
3	403	0.513876	95	160
6	406	0.124392	43	80
12	412	0.160133	50	176
13	413	0.077577	45	108
14	414	0.246164	73	120
15	415	0.151951	45	120
16	416	0.153741	54	120
17	417	0.33502	166	396
18	418	0.07788	47	108
19	419	0.075829	56	108
20	420	0.153234	38	120
21	421	0.114895	34	120
22	422	0.109216	29	120
23	423	0.07832	21	96
24	424	0.076352	21	96
25	425	0.10649	79	144
26	426	0.285686	120	240
27	427	0.220148	80	168
28	428	0.23382	67	120
		Total Number =	1392	3120
		Average =	66.28571	148.5714



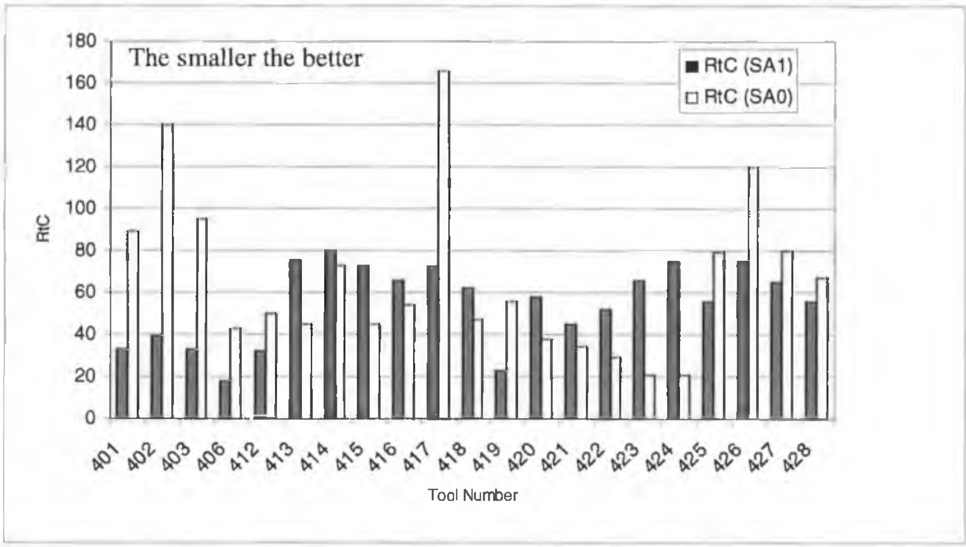


Figure 5.29: A Summary of experiments (9 and 18) output

5.9 Model Implementation

The manufacturing team has successfully employed the IPS model for scheduling the lots to the photolithography toolset showing the solution to be robust.

The average reduction of TPT was 14.78% with standard deviation 4.388%. Nine different production orders were compared with similar scenarios with IPS model to verify the model outputs, and the results were close to the actual production (see Table 5.16). The overall average deviation was 5.29% with standard deviation 1.36%.

Table 5.16: Comparison between actual data before and after IPS

Scenarios	WS	PM	Layers	Actual TPT (Hrs)	Simulated TPT(Hrs)	Dev %
				Using intelligent scheduling		
1	2600	5	1 (test)	63.11	60.58	4.17
2	2600	5	13	149.62	143.41	4.32
3	2600	10	13	170.99	162.85	5.02
4	3375	5	13	200.61	189.97	5.62
5	3375	8	13	206.45	199.47	3.51
6	3375	10	13	224.29	210.21	6.73
7	6000	8	13	235.01	225.32	4.32
8	6000	10	13	254.49	237.18	7.31
9	6000	15	13	275.33	257.80	6.8

The results of the model have caused a significant reduction in the TPT per lots within the photolithography area resulting in a clear improvement of average tool utilizations (Figure 5.30). Comparing the photolithography toolset performance before using IPS model with after using the model shows that the average TPT is higher (see Table 5.17).

Table 5.17: Simulation versus actual production

Scenarios	Actual TPT (Hrs)		Dev %
	Before IPS	After IPS	
1	70.12	63.11	11.13
2	170.55	149.62	14.1
3	196.64	170.99	15.05
4	220.22	200.61	9.78
5	227.09	206.45	10.07
6	252.96	224.29	12.78
7	282.77	235.01	20.32
8	308.35	254.49	21.16
9	327.31	275.33	18.88

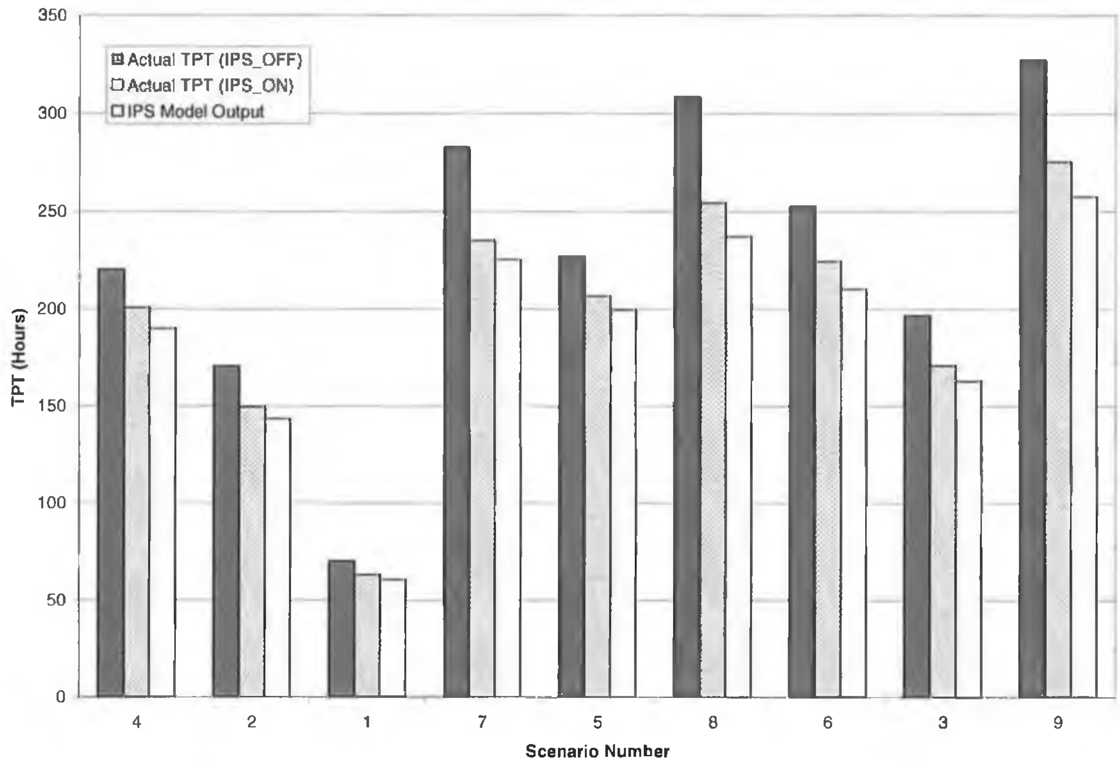


Figure 5.30: A Summary of model implementation experiments output

The scheduler has now become a reliable tool in the manufacturing area. Table 5.17 shows a minimum of 10% improvement in TPT in the photolithography toolset. In addition, Fab statistics indicate that after implementation of the scheduler, there has been a reduction in WIP and mean cycle time.

5.10 Concluding Remarks

There was an urge to develop intelligent scheduling in the photolithography area. Since, at present, the way to schedule is the expertise on the floor. The scheduling turns to be random in some stages due to the complexity of the production.

Theory of Constraints is a relatively recent development in the practical aspect of making organizational decisions in situations in which constraints exist. TOC has been used in our approach to guide the identification of the factors hindering the development, and proposing solutions to the global deployment of resources. One of the requirements for exploitation of the constraint is that the constraint equipment be utilized as much of the time as possible.

One of the main goals of the intelligent-agent based approach was to try to increase the tools' utilization and reduce TPT. In general, the more the qualified tools available in the factory, the easier scheduling process. In simpler industries, scheduling of lots goes into straight forward approach that directs the lot to the idle tool, while many criteria (e.g. qualifying matrix, reticle changes,..etc.) must be considered .

The approach used has considered the photolithography toolset in detail. The model has 21 tools. The results indicate that the model is viable, and highlight the importance of having such robust methodology to examine the production plans as well as the manufacturing equipment performance during the early stages of ramping up.

The results provide a number of interesting insights into the performance benefits. As one would expect the greatest benefit is obtained from improvements at the total throughput time and tools' utilization. Figure 5.31 and Figure 5.32 show a comprehensive comparison between some scenarios that clearly indicate the performance prior to the approach and after implementation. Applying Intelligent Scheduling has a significant effect on improving the lot distribution across the tools taking into consideration that having uniform lots distribution across the tools is impossible due to the high variability in the system (e.g. qualifying matrix, product-mix, and unscheduled maintenance). Nevertheless, the number of lots that will arrive at the toolset over a certain time interval, say the time required to complete 6000 wafers currently in process, will be better distributed compared to a situation where experts can only assign lots manually in the shop floor. In a simpler job shop environment with random or almost random scheduling, the workload tends to be fairly uniformly distributed across all the machines.

The semiconductor manufacturing system has different products with different configurations within complex production procedure and limited resources. The results obtained from IPS model support our conjecture that the benefits of applying the intelligent-agent based scheduling of lots using simulation global information are greater in shop configurations with high competition for capacity at key resources.

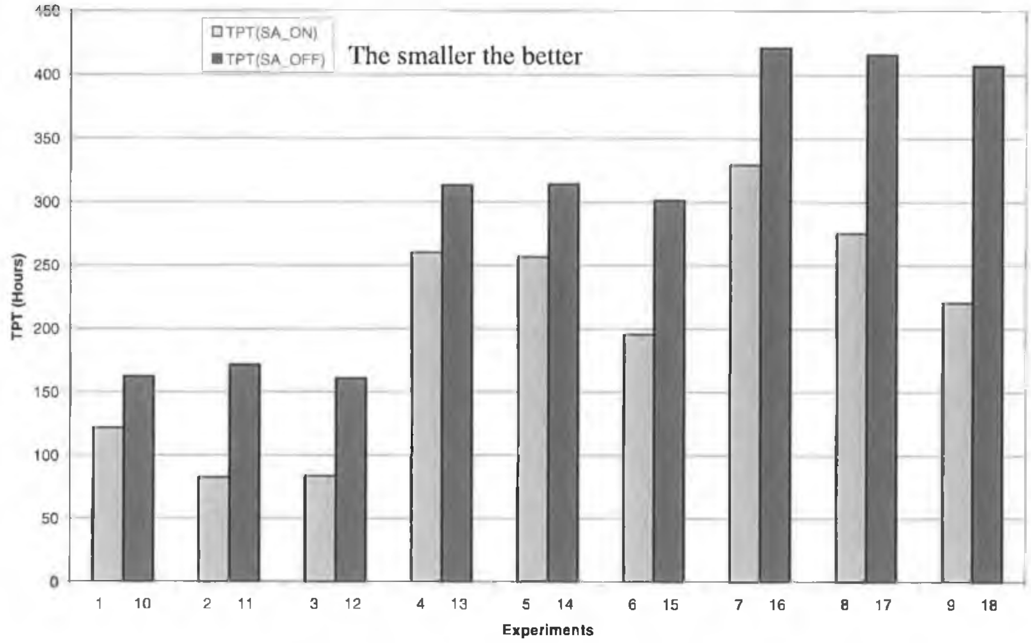


Figure 5.30: Effect of Intelligent Scheduling Approach on TPT

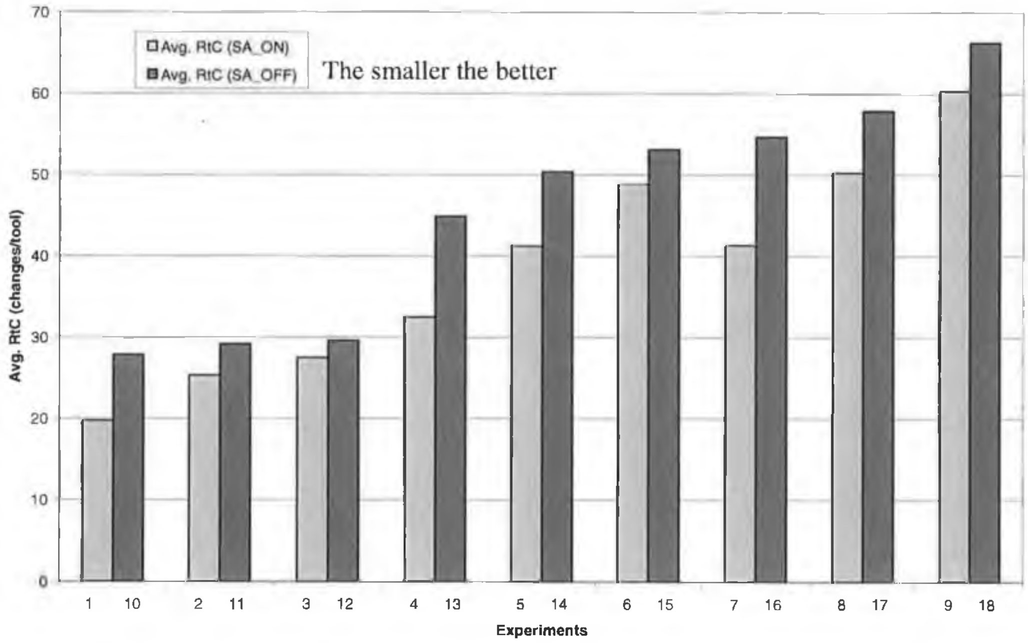


Figure 5.31: Effect of Intelligent Scheduling Approach on Average RtC per tool

The ANN for optimization demonstrates very promising properties for solving real-world scheduling problems. The proposed model follows three simple stages in order to capture a generalized pattern for the problem. Even though

the initial population was not as diverse in factor levels as experimentation, the ANN model efficiently searches for near optimum solution, without converging prematurely to a local minimum.
The obvious advantages of intelligent agent-base scheduling methodology were:

- speed,
- easy of application,
- flexibility to add more rules or constraints,
- effectivness for complicated toolset, and

The modeling phase is a novel approach to developing a simulation model. The modeling approach which has used SASM to show the flow of product as well as information through the system. The main benefit of this approach comes from the possibility to code this schematic model using any commercial simulation software package which has the requisite capability.

The benefits of having the modified simulation model with short execution time showed the efficacy of the modeling approach. The motivation for this approach stemmed from the fact that quick effective results of simulation are reported to be highly recommended by manufacturing team. More time can dedicated to analyze results and perform more sensitivity analyses. In addition, the error variations were within an acceptable range for the manufacturing team. Therefore, the extra effort involved in developing the modified simulation model seems worthwhile.
The main obstacles faced the simulation projects under investigation have been concluded in the following factors.

1) Absence of input data for the models.
2) Data inaccuracy.
3) Inability to address real sized problems.
4) Lack of sub-problem integration.
5) Software selection problem (software capabilities).
6) Understanding results.
7) Setting determined objectives.

Chapter 6

Conclusions and Recommendations for Future Research Work

6.1 Summary of the Thesis

The main theme of this thesis is the application of solving scheduling problems in semiconductor manufacturing systems. From a description of the manufacturing systems and scheduling problems in chapter 2, it can be seen that the pressures on manufacturers due to cost considerations, rapid growth of process technology, quality constraints, feature size reduction, and increasingly complex products are requiring ever higher efficiency systems to meet increased global competition. Scheduling is well known as the most important task in manufacturing planning activities. Solving scheduling problems has been recognized as an important key in improving manufacturing performance. Solution techniques of scheduling problems were classified into traditional and advanced techniques. Each of these techniques has advantages and pitfalls that feature at which type of problem is best used. The integration of number of these techniques provides a framework for developing tools that easily updated and adapted to dynamic situations, thus increasing the quality and accuracy of the solutions.

Planning problems in semiconductor manufacturing systems have attracted the researchers last decades. There are a number of techniques have been developed to support scheduling activity such as simulation and artificial intelligence. Chapter 3 has reported a review on related work and proposed the new methodology to develop intelligent scheduling model for photolithography area in semiconductor manufacturing facility. The steps into developing simulation models has been presented. These photolithography models were evaluated using existing models and actual data from the shop floor. The results obtained have proved that the new models are powerful tools and have been employed by the industry partner.

6.2 Conclusions

The conclusions of this thesis are successively presented as follows:

- Scheduling activity is one of the most complex tasks in the manufacturing systems. From the manufacturers and researchers observations, the complexity of this activity will increase in the future due to
    - global competition,
    - variations in customer demands,
    - decreasing in product life cycle,
    - rapid changes in technologies, and
    - quality constraints.

- This thesis has provided an extensive literature review of the solution techniques of scheduling problems. There are two main groups, analytical and advanced techniques. An overview has been given of the inherent strengths and weaknesses of each of these techniques.

- The dynamic environment of semiconductor manufacturing makes factory scheduling and dispatching in a semiconductor facility a complicated and challenging task. The research discusses the use of simulation in scheduling of these complex processes in such a flexible manufacturing environment.

- This thesis supports the conjecture that using simulation, specifically discrete event simulation, in semiconductor manufacturing provides an alternative to analytical and deterministic models. In turn simulation overcomes most of the shortcomings in these models.

- From the literature review, It has been seen that a good body of research has been published concerning the application of simulation in semiconductor manufacturing. However, there is a significant shortage of research in the photolithography manufacturing area, the most complex part of the production process. Most of the semiconductor factories that employ simulation often model the entire installation and typically treat each manufacturing area as an aggregate (and simplified) unit. As a consequence, potentially important events (e.g. maintenance downtime, repair time, priority of products, ..etc) may not be observed or considered in the simulation outputs.

- A detailed simulation model was built to characterize the flexible manufacturing cell in the photolithography process. It also examined the effect of scheduling, planning and control parameters on the cell performance. The FMC model demonstrates the benefits of using a framework of a simulation-based approach combined with Taguchi methodology to provide management with an effective decision support system (DSS).

- The model gives effective quick insights to the impact of changing scheduling policies and parameters on the performance of the FMC in one of the most critical manufacturing areas. It has also been applied successfully, verified, and provides the manufacturing team with a robust approach for better understanding of the behavior of the cells.

- Developing effective models incorporating all the process details, operating details, and manufacturing procedure details for scheduling becomes extremely complex. Well-thought out hybrid models based on the photolithography toolset can be effectively used to mimic the production flow and resources needed to support capacity and strategic planning. This model can also be used to predict and examine the performance of the photolithography toolset as well as the impact of various production parameters on that performance. These understandings are crucial to a facility prior to making investment decisions to improve production flow without the loss of throughput. To achieve this goal, the hybrid scheduling model was developed using '*EXTEND*' simulator software along with an intelligent-agent based approach and Taguchi methodology.

- Photolithography scheduling in semiconductor manufacturing is a very complex process because it can be considered as a multifaceted problem, e.g., the problems for lot release control, lot scheduling at serial processing tools and batch scheduling with the objective of minimizing mean TPT. The IPS model was developed to provide a solution to these problems using the information from the model which mimics that available on the factory floor. The model has been compared to actual production and other models used in industry. The results of a series of simulation scenarios showed that the newly developed model outperformed existing solutions for scheduling of photolithography.

- The IPS model was developed by interconnecting several modules of a single toolset. The ANN module was a simple one, trained with a limited set of data, yet its accuracy was satisfactory. The aim of the ANN is setting the near optimum weights of selection criteria for scheduling the lots in photolithography toolset under different production scenarios.

- The main observations made from the experiments in ANN are summarized as follows:
  1. The ANN model incrementally improves the solution quality over time as it is provided with more training exemplars;
  2. The neural-net approach achieves consistently better solution quality in significantly reasonable computational time.

  The results from ANN support the efficacy of using the neural-nets for real time scheduling applications.

- The benefits of using Taguchi experimental design methodology has to gain better understanding of the impact of the production scheduling parameters (e.g. product-mix, wafers start) on tool performance have been shown. The methodology can be used to provide quick insights into the significance factors in the system and their behavior.

- One of the competitive modeling techniques that was developed in this work is the SASM approach. It has been demonstrated with its use in coding the simulation model. The approach was effective in the example application of photolithography manufacturing area. It also enables new possibilities for simulation using any simulation software package that has the capabilities to code the model. All the blocks are simple and quick to model with the well-defined blocks. SASM has been used in modeling phase by the industrial partner and found to be easy to learn and use by personnel new to simulation.

- One of the major issues is how much time one needs to make a scheduling decision. The models developed in this thesis show an effective time to answer the required questions. Using appropriate modeling techniques to address the problem in an efficient way with clear predefined objectives is the key to success in building robust models. It is worth mentioning that it takes few minutes to run production order of 6000 wafer starts. The simulation run time with the quality of the results obtained has encouraged the industrial partner to employ the models as part of their day-to-day decision making system.

- The simulation running time was reduced from 8 hours to less than three minutes with only 4% average predicted error. The modified simulation model of photolithography toolset has enabled more knowledge for evaluating the impact of policy decisions on the real manufacturing system to be captured in a shorter time. Further time for sensitivity analysis has resulted from reducing the execution time of the model with more possibilities to provide more training data to neural networks module. Therefore, the extra effort involved in developing the modified simulation model was worthwhile.

- In terms of effectiveness of the decision-supporting capabilities, the model are reliable robust and useful. In addition, it has a satisfactory level of quality and integrity within the specific FMS problem domain knowledge. It is well known that the main limitation of most of simulation-based model is the data validity and accuracy. The proposed approach has been evaluated on a range of production plans by manufacturing team, and the model results also compared with previous deterministic techniques used in the facility. The model results were found to be better in terms of both speed and quality.

6.3 Recommendations for Future Research Work

The recommendations for future research come into two main areas, extending the application of IPS model, and scheduling research in general. The following are the recommendations presented.

- The intelligent scheduling approach described in this research use global information in a limited way, making local scheduling decisions at individual toolset. The model is generic and can be used for any photolithography toolset, although it is suggested to develop a global model that considers all the toolsets in the manufacturing facility. Instead of getting schedules of each toolset individually, a general view will be considered by linking toolset models with the rest of the manufacturing facility.

- The ANN module outputs used to update intelligent-agent based module and simulation were fed manually. In order to allow quicker feedback to the model, it is recommended to develop an automatic connection between the three modules, ANN, intelligent-based and simulation using spreadsheet.

- Artificial Intelligence has become one of the effective tools in solving many optimization problems. Designing a user-friendly expert system for photolithography to aid the non-specialist user in getting the appropriate schedule is a worthy objective for future research in this area.

- The new modeling approach using schematic diagrams has been briefly described. It is suggested to address the idea to establish a web simulation database that might successfully standardize the terminology while facilitating the addition of new symbols to the standard list via expert committee. The approach will help simulation software companies to adapt their software on the market needs as well as developing simulation entities (blocks) that can ease building the simulation model. In addition to that, it helps non-specialists in simulation to convert the schematic model to simulation codes without the need to understand the real world application.

- An interesting area of research is to explore the relationship between the simulation model complexity in terms of number of blocks, layers, and interrelationship and the simulation run time. This research will revolve around how level of detail the model could use to give feedback in an economic and reasonable time.

- Virtual manufacturing is one of the promising research areas for providing better understanding to the complex manufacturing systems. Virtual reality tools would greatly enhance the scheduling presentation and improve the understanding of the system behavior. Reference in this area are Kim *et al.* [216] and Chawla *et al.* [217].

- This research agrees with Rodd *et al.* [193] and Kopacek [194] who state that integration will be the main task facing manufacturing systems in the future. Morad *et al.* [195] have investigated the integration of process planning and scheduling using genetics algorithms. The model developed can handle a set of identical parallel machines that perform the same processes. Much of the work on intelligent system to integrate various manufacturing modules in order to provide full-integrated systems is still needed. Intelligent agents are seen as one of the keys for integration.

References

[1] Heavey, C., "A Model Management Systems Approach To Manufacturing Systems Design", PhD thesis, University College Galway, Ireland, (1993).

[2] King, J. R., "The Theory-Practice Gap in Job Shop Scheduling", The Production Engineer, March 1976.

[3] Reisman, A., Kumar, A., and Motwani, J., " Flow shop scheduling/sequencing: a statistical review of the literature, 1952 – 1994", IEEE Transactions on Engineering Management, Vol. 44, No. 3, pp 316 – 329, 1997.

[4] Groover, Mikell P. "Fundamentals of Modern Manufacturing: Materials, Processes, and Systems", John Wiley & Sons, Inc., 1996, ISBN: 0133121828.

[5] Meziane, Vadera, Kobbacy and Proudlove, "Intelligent Systems in Manufacturing: Current Developments and Future Prospects", Integrated Manufacturing System, Vol. 11, No. 4, pp 218-238, 2000.

[6] Hopp, W. J., and Spearman, M. L., "Factory Physics", Mc-Graw-Hill, 2001. ISBN: 0-256-24795-1.

[7] Shingo, S., and Dillon, A., "The sayings of Shigeo Shingo: Key Strategies for Plant Improvement (Japanese Management)", Productivity Press, 1988. ISBN: 0915299151.

[8] Zarembra, M. B., and Prasad, B., "Modern Manufacturing; Information Control and Technology", Sprinder-Verlag, NY, 1994. ISBN: 0-387-1989.

[9] Bedworth, D., and Bailey, J., "Integrated Production, Control Systems: Management, Analysis, and Design", John Wiley & Sons, 1988. ISBN: 0471821799.

[10] Weatherall, Alan, "Computer Integrated Manufacturing: From fundamental to implementation", Butterworth & Co. Ltd, 1988. ISBN: 0-408-00733-8.

[11] Waldner, J., "CIM: Principles of Computer Integrated Manufacturing", John Wiley & Sons Ltd., 1992. ISBN: 047193450 x

[12] Rehg, J.A and Kraebber, H.W., "Computer-Integrated Manufacturing", Prentice Hall, 2nd Edition, 2001. ISBN: 0130875538.

[13] Hannam, R., "Computer Integrated Manufacturing from concepts to realization", Addison-Wesley, 1997. ISBN: 0201175460.

[14] Arisha, A., "http://webpages.dcu.ie/~arishaa/CIMweb_files/CIMpage.htm", Computer Integrated Manufacturing Course, Dublin City University, 2002.

[15] International Standard Industrial Classification, http://www.europa.eu.int/

[16] Atherton, L. F., and Atherton, R. W., "Wafer Fabrication: Factory Performance and Analysis", Kluwer Academic, Boston, 1995.

[17] Avram, F., and Wein, L. M., "A Product Design Problem in Semiconductor Manufacturing", Operations Research, Vol. 8, 1989.

[18] Heath, D. C., Jackson P. L., Levesque, K., Fret, M., Tlakula, S., Akkan, C., "Chip Assignment Algorithms for Dynamic Wafer Design in

Semiconductor Manufacturing", Technical Report No. 935, Department of Operations Research and Industrial Engineering, Cornell University, 1990.

[19] Phdake, M. S., Kacker, R. N., Speeney, D. V., and Grieco, M. J., "Off-Line Quality Control in Integrated Circuit Fabrication Using Experimental Design", The Bell System Technical Journal, Vol. 62, No. 5, pp 1273 – 1309, 1983.

[20] Martin-Vega, L. A., Pippin, M., Gerdon, E., and Burcham, R., "Applying Just-In-Time in a Wafer Fab", IEEE Transaction on Semiconductor Manufacturing, Vol. 2, No. 1, 16 – 22, 1989.

[21] Hughes, R. A., and Shott, J. D., "The Future of Automation for High-Volume Wafer Fabrication and ASIC Manufacturing", Proceeding of IEEE, Vol. 74, No. 12, pp 1775 – 1793, 1986.

[22] Gupta, D., and Buzacott, J. A., "A Framework for Understanding Flexibility of Manufacturing Systems", Journal of Manufacturing Systems, Vol. 2, No. 2, p. 89, 1989.

[23] Ranky, P., "The Design and Operation of FMS: Flexible Manufacturing Systems", IFS Publications Ltd, Bedford, 1983.

[24] Kaighobadi, M., and Venkatesh, K., "Flexible Manufacturing Systems: An Overview", International Journal of Operations & Production Management, Vol. 14, No. 4, pp 26 – 49, 1994.

[25] Liu, J. and MacCarthy, B. L., "The classification of FMS scheduling problems", Int. J. Prod. Res., Vol. 34, No. 3, pp 647 – 656, 1996.

[26] Rachamadugu, R. and Stecke, K. E., "Classification and review of FMS scheduling procedures", Prod. Planning Control, Vol. 5, pp 2 – 20, 1994.

[27] Browne, J., Dubois, D., Rathmill, K., Seth, S. P., and Stecke, K. E., "Classification of flexible manufacturing systems", FMS Magazine, April, pp 114 – 117, 1984.

[28] Blazewicz, J., Ecker, Klaus H., Schmidt, Guenter and Weglarz, Jan, "Scheduling in Computer and Manufacturing Systems", Springer Verlag, Berlin 1994. ISBN: 3540580492.

[29] Conway, R. W., Maxwell, W.L. and Miller, I. W., "Theory of Scheduling", Addison Wesley, Reading, Massachusetts, 1967.

[30] French, S., "Sequencing and Scheduling", An introduction to the mathematics of the job shop, Ellis Horwood series, 1982.

[31] Lenstra, J. K., "Sequencing by Enumerative Methods", Mathematical Center Tract 69, Amsterdam, 1977.

[32] Pinedo, Michael, "Scheduling, Theory, Algorithms and Systems", Prentice Hall , 1998. ISBN: 0-13-706757-7.

[33] Rinnooy Kan A. H. G., "Machine Scheduling Problems: Classification, Complexity and Computations", Nijhoff, The Hague, 1976.

[34] Tanaev, V. S, Gordon, V. S., and Shafransky, Y. M., "Scheduling Theory, Single-Stage Systems", Kluwer, Dordrecht, 1994.

[35] Tanaev, V. S., Sotskov, Y. N., Strusevich, V. A., "Scheduling Theory, Multi-Stage Systems", Kluwer, Dordrecht, 1994a.

[36] Herrmann, J. W. and Snowdon, Jane L., "A Classification of Static Scheduling Problems", Complexity in Numerical Optimization, pp 203-253, 1993.

[37] Morton, Thomas E. and Pentico, David W, "HEURISTIC SCHEDULING SYSTEMS with Applications to Production and Project Management", John Wiley & Sons Inc.1993, ISBN: 0471578193.

[38] Pinedo, M., "Operations scheduling with applications in manufacturing and services", Irwin/McGraw-Hill, 1998. ISBN: 0072897791.

[39] Brucher, P., "Scheduling Algorithms", Springer-Verlag, Heidelberg, 1998. ISBN: 3-540-64105-X.

[40] Arisha, A. "Shop Scheduling: An Overview", Technical Report, School of Mechanical and Manufacturing Engineering, Dublin City University, 2001.

[41] Panwalker, S.S., Iskander, Wafik, "A Survey of Scheduling Rules". Operations Research, Vol.25, No.1, January-February 1977.

[42] Demirkol, E., Mehta, S., and Uzsoy, R., "Theory and Methodology: Benchmarks for shop scheduling problems", European Journal of Operational Research, Vol. 109, pp 137 – 141, 1998.

[43] Pinedo, Michael and Bhaskaran, Kumar, "Dispatching", Handbook of Industrial Engineering, pp 2182-2198, 1992.

[44] Blackstone, J. H., Philips, D. T., and Hogg, G. L., "A State-of-the-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations", Int. J. Production Research, Vol. 20, No. 1, pp 27 – 45, 1982.

[45] Jain, A. S., and Meeran, S., "A State-of-the-Art Review of Job Shop Scheduling Techniques", European Journal of Operational Research, Vol. 113, pp 390 – 434, 1999.

[46] Blazewich, J., Domschke, W., and Pesch, E., "The Job Shop Scheduling Problem: Conventional and New Solution Techniques", European Journal of Operational Research, Vol. 93, pp 1 – 33, 1996.

[47] Arisha, A., Young, P., EL Baradie, M., "Job Shop Scheduling Problem: An Overview" Proceeding of Flexible Automation and Intelligent Manufacturing "FAIM'01" Conference, 16 – 18 July, Dublin, Ireland, pp 682 – 693, 2001.

[48] Johnson, S. M., "Optimal Two and Three-Stage Production Schedules with Setup Times Included", Naval Research Logistics Quarterly, Vol. 1, No. 1, 1954.

[49] Jackson, J. R., "Simulation Research on Job Shop Production", Naval Research Logistics Quarterly, Vol. 4, pp 287-295, 1957.

[50] Lawrence M. Wein and Chevalier, Philippe B., "A Broader view of the Job Shop Scheduling Problem". Management Science, Vol. 38, No.7, July 1992.

[51] Gonzalez, T., and Sahni, S., "Flow Shop and Job Shop Schedulers: Complexity and Approximation", Operations Research, Vol. 26, pp 36 – 52, 1978.

[52] Garey, M. R., Johnson, D. S., and Sethi, R., "Complexity of Flow Shop and Job Shop Scheduling", Math. Opns. Res., Vol. 1, 117 – 129, 1976.

[53] Giffler, B. and Thompson, G. L., "Algorithms for Solving Production Scheduling Problems", Operations Research, Vol. 8, No. 4, 1960.

[54] Pinson, E., "The Job Shop Scheduling Problem: Concise survey and some recent Developments", Scheduling Theory and its applications Edited by P. Chretienne, E.G. Coffman Jr., J.K. Lenstra and Z. Lim, 1995 John Wiley & Sons Ltd.

[55] Hoitomt Debra J., Luh, Peter B., and Pattipati, Krishna R., "A Practical Approach to Job Shop Scheduling Problems", IEEE Transactions on Robotics and Automation, Vol. 9, No.1, Feb. 1993.

[56] Fisher, M. L., "Optimal Solution of Scheduling Problems using Lagrange multiplier: Part II", Symposium on the Theory of Scheduling and its Applications, Springer, Berlin, 1973.

[57] Adams, J., Balas E. and Zawack, D., "The Shifting Bottleneck Procedure for Job Shop Scheduling", Management Science, Vol.34, 1988, pp 391-401.

[58] Pinedo, Michael, Singer, Marcos, "A Shifting Bottleneck Heuristic for Minimizing the Total Weighted Tardiness in a Job Shop", Naval Research Logistics, Vol. 46, 1999, pp 1-17.

[59] Baker, Kenneth R., "Sequencing and Scheduling", John Wiley & Sons Inc.1998, ISBN: 0-9639746-1-0.

[60] Arisha, A., Young, P., and El Baradie, M., "Flow Shop Scheduling Problem: A Computational Study" Proceedings of the Sixth International Conference on Production Engineering and Design for Development (PEDD6), Cairo, Egypt, pp 543 – 557, 2002.

[61] White, K. P., and Rogers, R. V., "Job Shop Scheduling: Limits of the binary disjunctive formulation", International Journal of Production Research, Vol. 28, No. 12, pp 2187 – 2200, 1990.

[62] Brown, A. P. G. and Lomnicki, Z.A., "Some applications of the Branch and Bound Algorithm to the Machine Scheduling Problem". Operational Research Quarterly, Vol.17, 1966.

[63] Grabowski J., Nowicki E. and Zdrzalka S., "A Block Approach for Single Machine Scheduling with Release dates and Due Dates", European Journal of Operations Research, Vol. 26, pp 278-285, 1986.

[64] Wagner, H. M., "An Integer Linear Programming Model for Machine Scheduling", Naval Logistics Quarterly, Vol. 6, No. 2, 1959.

[65] Van Laarhoven P.J.M., Aarts E. H. L. and Lenstra J. K., "Job Shop Scheduling by Simulated Annealing", Operations Research, Vol. 40, pp 113-125, 1992.

[66] Elmaghraby, S. W. and Cole, R. T., "On the Control of Production in Small Job Shops". Industrial Engineering, Vol. 14, No. 4, 1963.

[67] Dzielinski, B. P. and Baker, C. T., "Simulation of Simplified Job Shop", Management Science, Vol. 6, No. 3, Apr. 1960.

[68] Ezat, A. M. and ELBaradie, M., "Flow-Shop Scheduling: Effect of Various Priority Rules on Minimizing Multiple Criteria", 30th International MATADOR Conference (UMIST), 1993.

[69] Gere, W. S., "Heuristics in Job Shop Scheduling", Management Science, Vol. 13, No. 3, 1966.

[70] Jones, C. H., "An Economic Evaluation of Job Shop Dispatching Rules", Management Science, Vol. 20, No. 3, 1973.

[71] Sculli, D., "Priority Dispatching Rules in Job Shops with Assembly Operations and Random Delays", OMEGA 8, 227-234, 1980.

[72] Sculli, D. and Tsang, K. K., "Priority Dispatching Rules In A Fabrication/Assembly Shop", Mathl Comput. Modeling, Vol. 13, No. 3, 1990, pp. 73-79.

[73] Ezat, A. M. and El Baradie, M., "A Computer Aided-Simulation for the Optimization of the Flow-Shop Scheduling Vs. Different Priority Rules", Proceedings of the Ninth Conference of the Irish Manufacturing Committee (IMC-9), pp 139-150, 1992.

[74] Zweben, M., Deale M. and Gargan, R., "Anytime Rescheduling", Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control, Morgan Kaufmann, San Francisco, Calif., 1990.

[75] Biefeld, E., and Cooper, L., "Bottleneck Identification Using Process Chronologies", Proceedings IJCAI-91, AAAI press, Menlo Park, Claif., 1991.

[76] Fox, Mark S. and Zweben, Monte, "Intelligent Scheduling", Morgan Kaufmann Publishers INC., 1994, ISBN: 415-392-2665.

[77] Stone, H.S., and Stone, J.M., "Efficient Search Techniques – An Empirical Study of The N-Queens Problem", IBM Journal of Research and Development, Vol. 31, pp 464 – 474, 1987.

[78] Keng, N., and Yun, D.Y.Y., "A Planning/Scheduling Methodology for the Constrainted Resource Problem", Proceeding IJCAI-89, AAAI Press, Menlo Park, Calif. 1990.

[79] Minton, S., Johnston, M., Philips, A.B., and Laird, P., "Minimizing Conflicts: A Heuristic Repair Method for Constraints Satisfaction and Scheduling Problems", Artificial Intelligence, Vol. 58, pp 161 – 205, 1992.

[80] Fox, M.S., Sadeh, N., and Baykan, C., "Constrainted Heuristic Search", Proceedings IJCAI-89, AAAI Press, Menlo Park, Calif., 1989.

[81] Srivastava, B., Kambhampati, S., and B. Do, Minh, "Planning the project management way: Efficient planning by effective integration of causal and resource reasoning in RealPlan", Artificial Intelligence, Vol. 131, No. 1-2 , pp 73 – 134, 2001.

[82] Jönsson, H., and Söderberg, Bo, "An information-based neural approach to generic constraint satisfaction", Artificial Intelligence, Vol. 142, No. 1, pp 1 – 17, 2002.

[83] Shang, Y., and Fromherz, M. P. J., "Experimental complexity analysis of continuous constraint satisfaction problems", Information Sciences, Vol. 153, July 2003, pp 1 – 36, 2003.

[84] Rosenthal, Don and Johnston, Mark, "A Look under the Hood", Interval Logic Corporation, 1997, http://www.interval-logic.com.

[85] Kouvelis, Panagiotis, "Design and planning problems in flexible manufacturing systems: a critical review", Journal of Intelligent Manufacturing, Vol. 3, pp 75 – 99, 1992.

[86] Sabuncuoglu, I., "A study of scheduling rules of flexible manufacturing systems: a simulation approach", International Journal of Production Research, Vol. 30, No.2, pp 527 – 546, 1998.

[87] Basnet, C., and Mize, J. H., "Scheduling and Control of flexible manufacturing systems: a critical review", Int. J. Compter Integrated Mfg, Vol. 7, No. 5, pp 340 – 355, 1994.

[88] Gupta, Y. P., Evans, G. W., and Gupta, M. C., " A Review of multi-criteria approaches to FMS scheduling problems", International Journal of Production Economics, Vol. 22, pp 13 – 31, 1991.

[89] Rodammer, F. A., and Preston White Jr., K., "A recent survey of production scheduling", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 18, No. 6, pp 841 – 851, 1998.

[90] Stecke, K. E. and Solberg, J. J. "Loading and control policies for a flexible manufacturing system", International Journal of Production Research, Vol. 19, No. 5, pp 481 - 490, 1981.

[91] Arbib, C., Lucertini, M. and Nicolo, F. "Workload balance and part-transfer minimization in flexible manufacturing systems", International Journal Flexible Manufacturing Systems, Vol. 3, No. 1, pp 5 – 25, 1990.

[92] Stoop, P. P. M. and Wiers, V. C. S., "The complexity of scheduling in practice", Int. J. Ops Prod. Managmt, Vol. 16, No. 10, pp 37 – 53, 1996.

[93] Wilhelm, W. E. and Shin, H. M., "Effectiveness of alternate operations in a flexible manufacturing system", Int. J. Prod. Res., Vol. 23, No. 1, pp 65 – 79, 1985.

[94] Ro, I. K. and Kim, J. I., "Multi-criteria operational control rules in flexible manufacturing systems (FMSs)", Int. J. Prod. Res., Vol. 28, No. 1, pp 47 – 63, 1990.

[95] Chen, I. J. and Chung, C. H. "Effects of loading and routings decisions on performance of flexible manufacturing systems", Int. J. Prod. Res. , Vol. 29, No. 11, pp 2209 – 2225, 1991.

[96] Kazerooni, A., Chan, F. T. S., and Abhary, K., " Real-time operation selection in an FMS using simulation – a fuzzy approach", Prod. Planning Control, Vol. 8, pp 771 – 779, 1997.

[97] Chan, F T S, "Evaluation of combined dispatching and routing strategies for a flexible manufacturing system", Proc Instn Mech Engs, Vol. 216 Part B: J Engineering Manufacture, pp 1033 – 1046, 2002.

[98] Stecke, K. E., "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems, Management Science, Vol. 29, pp 273 – 287, 1983.

[99] Raman, N., Talbot, F. B., and Rachamadugu, R., "Due date based scheduling in a general flexible manufacturing system", Journal of Operations Management, Vol. 8, pp 115 – 132, 1989.

[100] Uluzoy, G., and Bilge, U., "Simultaneous scheduling of machines and automated guided vehicles", International Journal of Production Research, Vol. 31, pp 257 – 287, 1993.

[101] Denziler, D. R., and Boe, W. J., "Experimental investigation of flexible manufacturing system scheduling rules", International Journal of Production Research, Vol. 25, No. 7, pp 979 – 994, 1987.

[102] Sabuncuoglu, I., and Hommertzheim, D. L., "Experimental investigation of FMS due-date scheduling problem: evaluation of machine and AGV scheduling rules", International Journal of Flexible Manufacturing Systems, Vol. 5, pp 301 – 324, 1993.

[103] Sabuncuoglu, I., and Hommertzheim, D. L., "Experimental investigation of an FMS due-date scheduling problem: an evaluation of due-date assignment rules", International Journal of Computer Integrated Manufacturing, Vol. 8, pp 133 – 144. 1995.

[104] Kusiak, A., and Chen, M., "Expert systems for planning and scheduling manufacturing systems", European Journal of Operations Research, Vol. 34, No. 3, pp 113 – 130, 1988.

[105] Udo, G. J., " Neural Networks Applications in Manufacturing Processes", Computer and Industrial Engineering, Vol. 23, No. 1-4, pp 94 – 100, 1992.

[106] Chambers, M., and Mount-Campbell, C.A., "Process Optimization via Neural Networks Metamodeling", Int. J. Production Economics, Vol. 79, pp 93 – 100, 2002.

[107] Chan, F.T.S., Jiang, B., and Tang, N.K.H., "The development of Intelligent Decision Support tools to aid the Design of Flexible Manufacturing System", Int. J. Production Economics, Vol. 65, pp 73 – 84, 2000.

[108] Hannam, Roger, " Computer Integrated Manufacturing: from concepts to realization" Addison-Wesley, 1997, ISBN: 0 201 17546 0.

[109] Ross, D. T., " Structure analysis (SA): a language for communicating ideas" IEEE Trans. on Software Engineering, SE-3, No. 1, pp 16-34, 1977.

[110] United States Air Force, " Integrated Computer Aided Manufacturing (ICAM)" Architecture Pt II, Vol. IV, Function Modeling Manual (IDEF0). Air Force Materials Laboratory, Wright-Paterson AFB, Ohio, 45433, AFWAL-TR-81-4023, June 1981.

[111] Banks, J., Carson, J., Nelson, B. L., Nicol, D. M. "Discrete-Event System Simulation", Prentice-Hall, Inc. , NJ, 2001.

[112] Hollier, R. H., " A Simulation Study of Sequencing in Batch Production", Operation Research Quarterly, Vol. 19, No. 4, 1986.

[113] Kochhar, A.K., "Computer simulation of manufacturing system – 3 decades of progress", Proceedings of the third European Simulation Congress, Edinburgh, UK, pp 3-9, 1989.

[114] Shannon, R.E, "Systems Simulation-The Art and Science", Prentice-Hall, 1995.

[115] Arisha, A., and El Baradie, M., "Simulation Based Rescheduling of Manufacturing Resources: A Real-World Application", ESDA 2002, 6th Biennial Conference on Engineering Systems Design and Analysis, Istanbul, Turkey, July 8-11, 2002, Ref. No. ESDA2002/MAN-063.

[116] Carrie, Allan, "Simulation of Manufacturing Systems", John WILEY & Sons, 1998. ISBN: 0471 915 74 2.

[117] Uzsoy, R., Lee, Chung-Yee, and Martin-Vega, L. "A review of production planning and scheduling models in the semiconductor manufacturing, Part I: system characteristics performance evaluation and production planning", IIE transaction on scheduling and logistics, Vol. 26, pp 44 – 55, 1994.

[118] Moench, Lars, Prause, Matthias, M. and Schmalfuss, V., "Simulation-based solution of load-balancing problems in the photolithography area of a semiconductor wafer fabrication facility", Proceedings of the 2001 Winter Simulation Conference, pp 1170- 1177, 2001.

[119] Hunter, J., Delp, D., Collins, D., and Si, J., "Understanding a Semiconductor Process Using a Full-Scale Model", IEEE Transactions on Semiconductor Manufacturing, Vol. 15, No. 2, 2002.

[120] Ignizio, J. P., "A generic toolset simulation template for semiconductor Fab", In press, 2003.

[121] Sloan, T. W., "Shop-Floor Scheduling of Semiconductor Wafer Fabs: Exploring the Influence of Technology, Market, and Performance", IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 2, 2003.

[122] Mackulak, G. T., Savory, P., "A Simulation-Based Experiment for Comparing AMHS Performance in a Semiconductor Fabrication Facility", IEEE Transaction on Semiconductor Manufacturing, Vol. 14, No. 3, pp 273 – 280, 2001.

[123] Lin, J. T., Wang, F-K., and Yen, P-Y., "Simulation analysis of dispatching rules for an automated interbay material handling system in wafer fab", Int. J. Prod. Res., Vol. 39, No. 6, pp 1221 – 1238, 2001.

[124] Sivakumar, A. I., and Chong, C. S., "A simulation based analysis of cycle time distribution, and throughput in semiconductor backend manufacturing", Computers in Industry, Vol. 45, pp 59 – 78, 2001.

[125] Vargas-Villamil, F. D., and Rivera, D., "Multilayer optimization and scheduling using model predictive control: application to reentrant semiconductor manufacturing lines", Computers and Chemical Engineering, Vol. 24, pp 2009 – 2021, 2000.

[126] Dabbas, R. M., Chen, H-N., Fowler, J. W., and Shunk, D., "A combined dispatching criteria approach to scheduling semiconductor manufacturing systems", Computers & Industrial Engineering, Vol. 39, pp 307 – 324, 2001.

[127] Dayhoff, J. E., and Atherton, R. W., "Signature Analysis of Dispatch Schemes in Wafer Fabrication", IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 9, No. 4, 1986.

[128] Antherton, R. W., and Dayhoff, J. E., "Signature Analysis: Simulation of Inventory, Cycle Time, and Throughput Tade-offs in Wafer Fabrication", IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 9, No. 4, 1986.

[129] Uzsoy, R., Martin-Vega, L.A., Brown, S.M., and Leonard, P.A., "Production Scheduling Algorithms for a Semiconductor Test Facility", Proceedings of International Semiconductor Manufacturing Science Symposium, San Mateo, CA, May 1989.

[130] Uzsoy, R., Lee, C.Y., Martin-Vega, L.A., and Hinchman, J., "Scheduling Semiconductor Test Operations: Optimization and Approximation", in New Directions for Operations Research in Manufacturing, G. Fandel, T. Gulledge, A. Jones (eds.), Springer Verlag Lecture Notes in Economics and Mathematical Systems, pp 179 – 199, 1992.

[131] Uzsoy, R., and Demirkol, E., "Decomposition Methods for Scheduling Reentrant Flow Shops with Sequence Dependent Setup Times", Journal of Scheduling, Vol. 3, 155 – 177, 2000.

[132] Ovacik, I.M., and Uzsoy, R., "Decomposition Methods for Complex Factory Scheduling Problems", Kluwer Academic Publishing, 1997.

[133] Kempf, K.G., Uzsoy, R., Smith, S.F., "Evaluation and Comparison of Production Schedules", Computers in Industry, Vol. 42, pp 203 – 220, 2000.

[134] Akcali, E., Nemoto, K., and Uzsoy, R., "Quantifying the Benefits of Cycle-Time Reduction in Semiconductor Wafer Fabrication", IEEE Transactions on Electronics Packaging Manufacturing, Vol. 23, pp 39 – 47, 2000.

[135] Lee, Y. , Kim, S., Yea, S., Kim, B., "Production Planning in Semiconductor Wafer Fab Considering Variable Cycle Times", Computer Industrial Engineering, Vol. 33, No. 3 – 4, pp 713 – 716, 1997.

[136] Williams, D., and Favero, D., "Dynamic Deployment Modeling Tool for Photolithography WIP Management", IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp 55 – 58, 2002.

[137] Lachman-Shalem, S., Grosman, B., and Lewin, D. R., "Nonlinear Modeling and Multivariable Control of Photolithography", IEEE Transactions on Semiconductor Manufacturing, Vol. 15, No. 13, 2002.

[138] Yoon, H. J., and Lee, D. Y., "Deadlock-Free Scheduling Method for Track Systems in Semiconductor Fabrication", IEEE Transactions on Semiconductor Manufacturing, Vol. 13, pp 1787 – 1792, 2000.

[139] Karafylidis, I., "Design of a dedicated parallel processor for the simulation of photolithography process using a genetic algorithm", Engineering Applications of Artificial Intelligence, Vol. 12, pp 411 – 427, 1999.

[140] Gurney, K., " An introduction to Neural Networks", UCL Press, 1997.

[141] Zhang, H. C., and Huang, S. H., " Application of neural networks in manufacturing: a state of the art", International Journal of Production Research, Vol. 33, No. 3, pp 705 – 728, 1995.

[142] Grabot, B., "Objective Satisfaction Assessment using Neural Network Approach", International Journal of Production Research, Vol. 36, No. 9, pp 2377-95, 1998.

[143] Hopfield, J. J., and Tank, D. W., "Neural computation of decisions in optimization problem", Biological Cybernetics, Vol. 52, No. 3, pp 141 – 152, 1985.

[144] Maa, C. Y., Chin, C., and Shanblatt, M. A., "Improved linear programming neural networks", Proceedings of the 31st Midwest Symposium, on Circuits and Systems, Urbana, Illinois, pp 748 – 751, 1989.

[145] Vaithyanathan, S., and Ignizio, J. P., "A stochastic neural network for resource constrained scheduling", Computer and Operations Research, Vol. 19, No. 3, pp 241 – 254, 1992.

[146] Arizono, I., Yamamoto, A., and Ohta, H., "Scheduling for minimizing total actual flow time by neural networks", International Journal of Production Research, Vol. 30, No. 3, pp 503 – 511, 1992.

[147] NeuroShell® Predictor , Ward System Group, 1997.

[148] Shen, W., Maturana, F., and Norrie, D. H., "MetaMorph II: An Agent-Based Architecture for Distributed Intelligent Design and Manufacturing", Journal of Intelligent Manufacturing, Vol. 11, No. 3, pp 237 – 251, 2000.

[149] Shen, W., Norrie, D.H., and Barthes, J.P., "Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing", Taylor and Francis, London, 2000.

[150] Kopacek, P., "Intelligent manufacturing: present states and future trends", Journal of Intelligent and Robotic Systems, Vol. 26, No. 3 – 4, pp 217 – 229, 1999.

[151] Lee, W. B., and Lau, H. C. W., "Multi-agent modeling of dispersed manufacturing networks", Expert Systems with applications, Vol. 16, No. 3, pp 297 – 306, 1999.

[152] Sycara, K. "Resource Allocation in Distribution Factory Scheduling", IEEE Expert, Vol. 6, No. 1, pp 29 – 40, 1991.

[153] Burke, P., and Prosser, P., "The Distributed Asynchronous Scheduler", Intelligent Scheduling, M. Zweben and M.S.Fox, eds., Morgan Kaufmann, San Francisco, pp 309 – 339, 1994.

[154] Smith, R. G., "The Contract Net Protocol: High-Level Communication and Control in Distributed Problem Solver", IEEE Trans. Computers, Vol. 29-C, No. 12, pp 1104 – 1113, Dec. 1980.

[155] Charalambous, O., and Hinidi, K. S., " A review of artificial intelligence-based job shop scheduling", Information and Decision Technologies, Vol. 17, No. 1, pp 189 – 201, 1991.

[156] Sereco, A. C., Roth, S. P., Williams, B. D., and Gilmore, J., " The GENSHED project: general scheduling knowledge base system", Georgia Institute of Technology, Feb. 1986.

[157] Dagli, C., and Huggahalli, R., " A neural network architecture for faster dynamic scheduling in manufacturing systems", Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 3, IEEE Computer Society Press, Washington, DC, pp 2408 – 2413, 1991.

[158] Rabelo, L. C., and Alptekin, S., " A hybrid neural and symbolic processing approach to flexible manufacturing systems scheduling", in Kandel, A. and Langholz, G. (Eds), Hybrid Architecture for Intelligent System, CRC Press, Boca Raton, FL, pp 381 – 405, 1992.

[159] Rabelo, L. C., Yih, Y., Jones, A., and Tsai, J. S., " Intelligent scheduling for flexible manufacturing systems", Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 3, pp 810 – 815, 1993.

[160] Yih, Y., Rabelo, L. C., and Jones, A., " A hybrid AI/Simulation framework for FMS scheduling", Proceedings of Conference on New Directions in

Simulation for Manufacturing and Communications, Operations Research Society of Japan, Tokyo, pp 204 – 210, 1994.

[161] Yih, Y., Liang, T. P., and Moskowitz, H., " Robot scheduling in a circuit board production line: a hybrid OR/ANN approach", IEE Transactions Operations Research Society of Japan, Tokyo, Vol. 25, No. 2, pp 26 – 33, 1993.

[162] MacCarthy, B. L. and Jou, P., "Set-up reduction through production scheduling", Proceedings of the 2nd International Conference Computer Integrated Manufacturing – ICCIM '93, Vol. 1., World Scientific & Global Publications, Singapore, pp 668 – 674, 1993.

[163] Sim, S. K., Yeo, K. T., and Lee, W. H., " An expert neural network system for dynamic job shop scheduling", International Journal of Production Research, Vol. 32, No. 8, pp 1759 – 1773, 1994.

[164] Szelke, E., and Markus, G. A., " Learning reactive scheduler using CBR/L", Computers in Industry, Vol. 33, No. 1, pp 31 – 46, 1997.

[165] Kim, C. O., Min, H. S., and Yih, Y., " Integration of inductive learning and neural networks for multi-objectives FMS scheduling", International Journal of Production Research, Vol. 36, No. 9, pp 2497 – 2509, 1998.

[166] Lee. C.Y., Piramuthu, and Tsai, Y.K." job Shop Scheduling with Genetic Algorithm and Machine Learning", International Journal of Production Research, Vol. 35, No. 4, pp 1171 - 1191, 1998.

[167] Grant, H., and Clapp, C., " Making production scheduling more efficient helps control manufacturing costs and improve productivity", International Conference on Industrial Engineering, Institute of Industrial Engineers, Singapore, 1988.

[168] M. G. Gnoni, M.G., Iavagnilio, R., Mossa, G., Mummolo, G., and Di Leva,A. "Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry", International Journal of Production Economics, Vol. 85, No. 2, pp 251 – 262, 2003.

[169] Yan, Hong-Sen, Wang, Z. and Jiao, X. "Modeling, scheduling and simulation of product development process by extended stochastic high-level evaluation Petri nets", Robotics and Computer-Integrated Manufacturing, Vol. 19, No. 4, pp 329 – 342, 2003.

[170] Jensen, J. B., "The impact of resource flexibility and staffing decisions on cellular and departmental shop performance", European Journal of Operational Research, Vol. 127, No. 2, Pages 279 – 296, 2000.

[171] Arisha, A., Young, P., and El Baradie, M., "Flow Shop Scheduling Problem: A Computational Study", Proceedings of the Sixth International Conference on Production Engineering and Design for Development (PEDD6), Cairo, Egypt, pp 543 – 557, 2002.

[172] Hilton, C., "Manufacturing Operations System Design and Analysis", Intel Technology Journal, 4th Quarter '98, 1998.

[173] Doniavi, A., Mileham, A. R., and Newnes, L. B. "A systems approach to photolithography process optimization in an electronics manufacturing

environment" International Journal of Production Research, Vol. 38, pp 2515- 2528, 2000.

[174] Prasad, Kowdle, "A Generic Computer Simulation Model to Characterize A Manufacturing Area in an IC FAB Facility", IEEE transactions on Components, Hybrids, and Manufacturing Technology, Vol. 14, No. 3, Sept. 1991.

[175] Sawik, Tadeuez, "Production Planning and Scheduling in Flexible Assembly Systems" Springer – Verlag Berlin Heidelberg, 1999. ISBN: 3 540 64998 0.

[176] Jacobs, J. H., Etman, L. F. P., Van Campen, E. J. J., and Rooda, J. E., "Charaterization of Operational Time Variability Using Effective Process Times", IEEE transactions on Semiconductor Manufacturing, Vol. 16, No. 3, August 2003.

[177] Imagine That Inc., Extend 5.0, San Jose, CA, USA, http://www.imaginethatinc.com, 2002.

[178] Nayani, N. and Mollaghasemi, M. "Validation and verification of the simulation model of a photolithography process in semiconductor manufacturing", Proceedings of the 1998 Winter Simulation Conference, pp 1017- 1022, 1998.

[179] Roy, Ranjit, Design of Experiment Using The Taguchi Approach: 16 Steps To Product and Process Improvement, A John Wiley & Sons, INC., 2001. ISBN: 0471361011.

[180] Phadke, M. S., "Quality Engineering Using Robust Design", Prentice-Hall International, Englewood Cliffs, NJ, 1989. ISBN: 0137451679.

[181] Hicks, C., "Fundamental Concepts in the Design of Experiments", Saunders College Publishing, 1993. ISBN: 003097710-X

[182] Caprihan, R. and Wadhwa, S., "Impact of Routing Flexibility on the Performance of an FMS – A Simulation Study", International Journal of Flexible Manufacturing Systems, Vol. 9, pp 273 – 298, 1997.

[183] Peace, G., "Taguchi Methods, A hands-on Approach", Addison-Wesley Publishing Company Inc., 1993. ISBN: 0201563118

[184] Nemoto, K., Akcali, E., and Uzsoy, R. M., "Quantifying the Benefits of Cycle Time Reduction in Semiconductor Wafer Fabrication", IEEE Transactions on Electronic Packaging Manufacturing, Vol. 23, No. 1, 2000.

[185] Rodd, M. G., Verbruggen, H. B., and Krijgsman, A. J., "Artificial Intelligence in real-time control", Engineering Applications of Artificial Intelligence, Vol. 5, No. 5, pp 385 – 399, 1992.

[186] O'Keefe, RM., Wildman, LE., Loparo, K. A., and Nielsen, N. R., "The role of artificial intelligence in discrete-event simulation", Artificial Intelligence, Simulation, and Modeling, John Wiley & Sons, NY, pp 359 – 379, 1990.

[187] Rovithakis, G.A., Gaganis, V. I., Perrakis, S. E., and Christodoulou, M. A., " Neuro-schedulers for flexible manufacturing systems", Computers in Industry, Vol. 39, No. 3, pp 209 – 217, 1999.

[188] Goldratt, Eliyahu M. and Cox, Jeff, "The Goal", Second Revised Edition, Corton-on-Hudson, N.Y.: North River Press, 1992.

[189] Cunningham, J. A, “The Use and Evaluation of Yield Models in Integrated Circuit Manufacturing”, IEEE Transactions on Semiconductor Manufacturing, Vol. 3, No. 2, pp 60 – 72, 1990.

[190] Wang, J., “Neural Network models and their role for optimization: an Overview”, ORSA CSTS Newsletter, Vol. 15, No. 1, pp 4 – 8, 1994.

[191] Lee, In, and Shaw, M. J., “A neural-net approach to real time flow-shop sequencing”, Computer & Industrial Engineering, Vol. 38, pp 125 – 147, 2000.

[192] Neutek, inc., Qualitek 4.0, Michigan, USA, http://www.rkroy.com, 2001.

[193] Rodd, M. G., Verbruggen, H. B., and Krijgsman, A. J., “Artificial Intelligence in real-time control”, Engineering Applications of Artificial Intelligence, Vol. 5, No. 5, pp 385 – 399, 1992.

[194] Kopacek, P., “Intelligent manufacturing: present states and future trends”, Journal of Intelligent and Robotic Systems, Vol. 26, No. 3 – 4, pp 217 – 229, 1999.

[195] Morad, N. and Zalzala, A. M. S., “Genetic Algorithms in Integrated Process Planning and Scheduling”, Journal of Intelligent Manufacturing, Vol. 10, No. 2, pp 169-79, 1999.

[196] Arisha, A., Young, P., and El Baradie, M., “A Simulation Model to Charaterize Photolithography Process Of A Semiconductor Wafer Fabrication”, AMPT’03, Dublin, pp 539 – 542, 2003.

[197] Shen, Y., and Leachman, R. C., “Stochastic Wafer Fabrication Scheduling”, IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 1, Feb. 2003.

[198] Chern, Ching-Chin, and Liu, Yu-Lien, “Family-Based Scheduling Rules of a Sequence-Dependant Wafer Fabrication System”, IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 1, 2003.

[199] Kim, Yeong-Dae, Shim, Sang-Oh, Choi, B., and Hwang, H., “Simplification Methods for Accelerating Simulation-Based Real-Time Scheduling in a Semiconductor Wafer Fabrication Facility”, IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 2, 2003.

[200] Dabbas, R. M., and Fowler, John W., “A New Scheduling Approach Using Combined Criteria in Wafer Fabs”, IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 3, 2003.

[201] Chen, Jyh-Horng, Fu, L., Lin, M-H, Huang, A., “Petri-Net and GA-Based Approach to Modeling, Scheduling, and Performance Evaluation for Wafer Fabrication”, IEEE Transactions on Semiconductor Manufacturing, Vol. 17, No. 5, 2001.

[202] Kim, Yeong-Dae, Shim, Sang-Oh, Choi, Bum, and Hwang, Hark, “Simplification Methods for Accelerating Simulation-Based Real-Time Scheduling in a Semiconductor Wafer Fabrication Facility”, IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 2, 2003.

[203] Taguchi, G., and Konishi, S., “Orthogonal Arrays and Linear Graphs”, Dearborn, MI: American Supplier Institute, 1987.

[204] Antony, J., and Kaye, M., “Experimental Quality”, IEE Manufacturing Magazine, Vol. 74, No. 4, pp 178 – 181, 1999.

[205] Krishna, BV, Venugopal, P., and Rao, KP, "Optimization of process parameters in cold solid state joining of sintered stell and copper P/M predorms", Transaction of the Indian Institute of Metals, Vol. 54, No. 4, pp 363 – 373, 2003.

[206] Namose, I., "Optimization of gas utilization in plasma processes", IEEE Transaction on Semiconductor Manufacturing, Vol. 16, No. 3, pp 429 – 435, 2003.

[207] Chen, L-H, and Chen, Y-H, "A Design Procedure for a Robust Job Shop Manufacturing System under a Constraint Using Experiments", Computer Industrial Engineering, Vol. 30, No. 1, pp 1 – 12, 1996.

[208] Mezgar, I., Egresits, Cs., and Monostori, L., "Design and real-time reconfiguration of robust systems by using design of experiements", Computer in Industry, Vol. 33, pp 61 – 70, 1997.

[209] Khaw, J. F.C., Lim, B. S., and Lim, Lennie, E. N., "Optimal design of neural networks using the Taguvhi Method", Neurocomputing, Vol. 7, pp 225 – 245, 1996.

[210] D'Anglo A., Gastaldi, M., and Levialdi, N., "Performance Analysis of a FMS: A statistical Approach", Int. J. of Production Economics, Vol 56 – 57, pp 47 – 59, 1998.

[211] Peace, GS., "Taguchi methods, a hands-on approach", Reading, MA: Addison-Wesley, 1993.

[212] Clausing, DP., "Total quality development", NY: ASME, 1995.

[213] Arisha, A., Young, P., and El Baradie, M., "Evaluation of Scheduling Strategies on the Performance of a Flexible Manufacturing Cell - A Simulation Study", ICC&IE'32, Limerick, pp 700 – 706, 2003.

[214] Law, A. M., and Kelton, W. D., "Simulation Modeling and Analysis", New York: McGraw-Hill, ISBN: 0070592926, 1999.

[215] Parks JM., and Li, C., "Mathematical techniques and software for stochastic design optimization", Proceedings of the Seventh ASCE Speciality Conference on Probablistic Mechanics and Structural Reliability, ASCE, NY, pp 118 – 121, 1999.

[216] Kim, H., Zhou, C., and Du, Hua X., "Virtual Machines for Message Based, Real-Time, And Interactive Simulation", Proceeding of the 2000 Winter Simulation Conference, pp 1529 – 1532, 2000.

[217] Chawla, R., and Banerjee, A., "A Virtual Environment For Simulation Manufacturing Operations in 3D", Proceeding of the 2001 Winter Simulation Conference, pp 991 – 997, 2001.

Appendices

Appendix A

Review on Semiconductor Manufacturing Basic Processes

Basic Processes in Semiconductor Manufacturing

The process by which integrated circuits are manufactured can be divided into four basic steps: wafer fabrication, wafer probe, assembly or packaging, and final testing as shown in Figure 1.

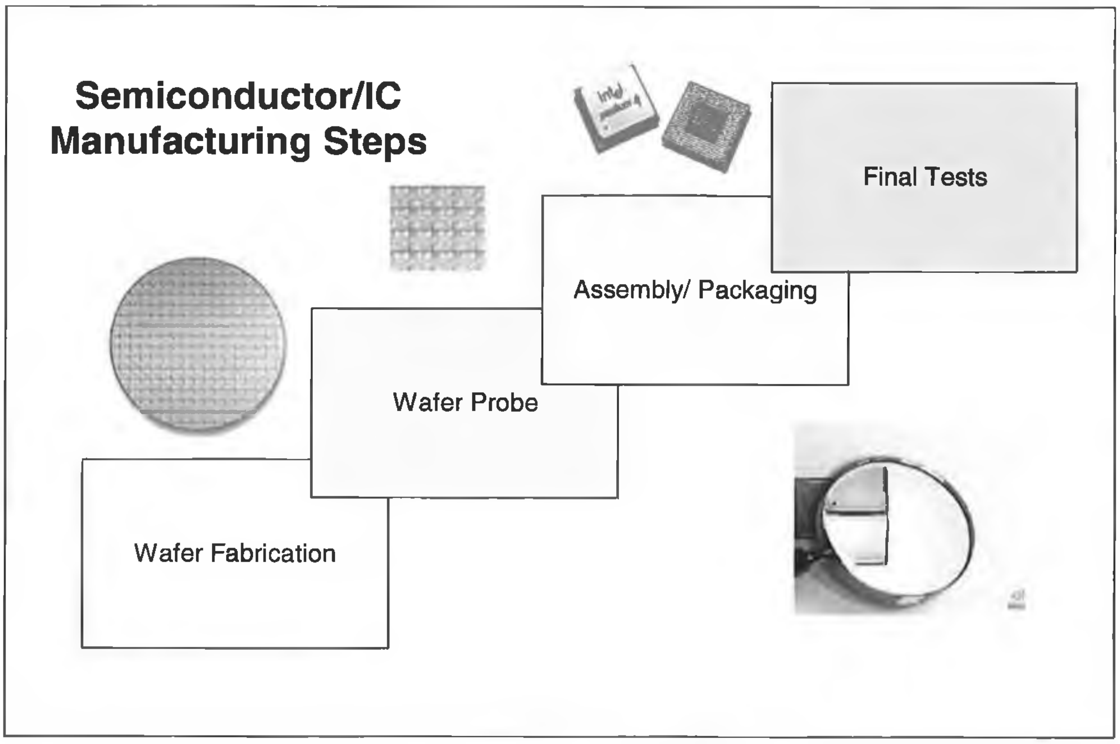


Figure 1: Basic Steps in very large-scale integrated circuits manufacturing

What follows is a brief description of the four basic steps in semiconductor manufacturing, more detailed description may be found in the literatures if required (e.g. Groover [1], Uzsoy *et al.* [2], Runyan *et al.* [3], and Atherton *et al.* [4]).

- **Wafer Fabrication**

Wafer fabrication is the most technologically complex and capital intensive of all four phases. It involves the processing of wafers of silicon or gallium arsenide in order to create the semiconductor devices in the wafer and build up the layers of conductors and dielectric on top that provide the complex interconnection between devices. The total number of operations can be in the hundreds to build a complex component such as a microprocessor. Many of these operations have to be performed in a clean-room environment to prevent particulate contamination of the wafers. The facility in which wafer fabrication takes places is referred as a wafer Fab. While the specific operations may vary widely depending on the product and the technology in use, an idea of the processes in wafer fabrication can be seen in the Figure 2.

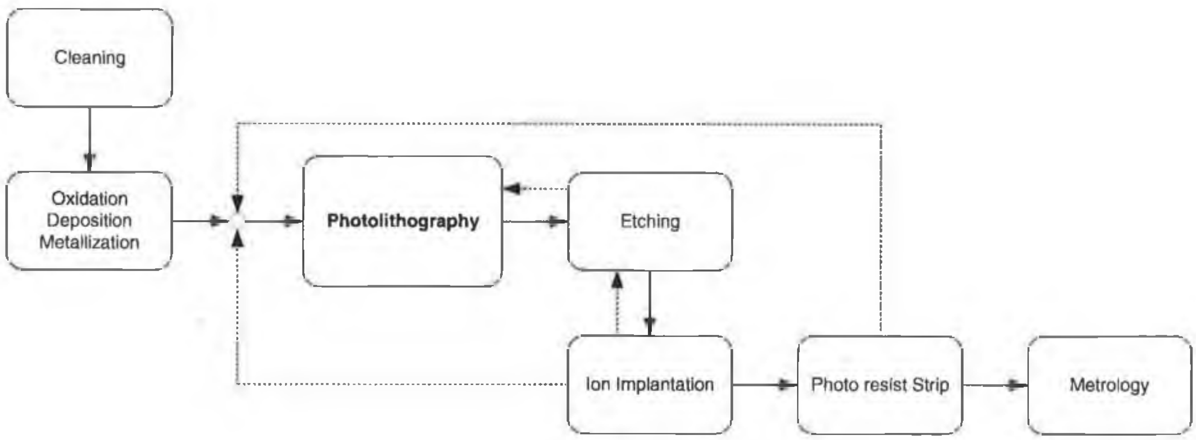


Figure 2: Main processes in wafer fabrication

Brief description of basic manufacturing process steps for wafer fabrication includes:

Cleaning: The object of this operation is the removal of particulate matter before a layer of circuitry is produced.

Oxidation, deposition, metallization: A layer of material is grown or deposited on the surface of the cleaned wafer. Extensive setup times are involved, resulting in machines being dedicated to a limited number of operations.

Photolithography: This is the most complex operation, as well as the one requiring greatest precision. A photo-resistant liquid is deposited onto the wafer and the circuitry defined using photography. The photo-resist is first deposited and baked. It is then exposed to ultraviolet light through a mask which contains the pattern of the circuit. Finally the exposed wafer is developed and baked.

Etching: In order to define the circuits, the exposed material is etched away.

Ion Implantation: Selected impurities are introduced in a controlled fashion to change the electrical properties of the exposed portion of the layer. Setup times may range from minutes to hours.

Photo-resist Strip: The photo-resist remaining on the wafer is removed by a process similar to etching.

Metrology: The layer inspected and measured to identify defects and guide future operations.

One of the features of semiconductor manufacturing is that the sequence of operations is repeated for each layer of circuitry on the wafer. Detailed descriptions of the technologies used in wafer fabrication can be found in texts on this subject such as Sze [5] and Runyan *et al.* [3].

- **Wafer Probe**

In wafer probe, the individual circuits, of which there may be hundreds on each wafer, are tested electrically by means of thin probes. Circuits that fail to meet specifications are marked with an ink dot. The wafers are then cut up into individual circuits and the defective circuits discarded.

Wafer fabrication and probe are generally referred to as "front-end" operations. The following stages, assembly and final test, are referred to as the "back-end". In back-end operations, lots may vary in size from several individual circuits to several thousands. The actual sequence of operations a lot will go through depends on the product and on customer specification. These characteristics are due to the fact that a lot is generally more closely associated with a particular order and customer than is the case in wafer Fab or probe.

- **Assembly**

In assembly the circuits are placed in plastic or ceramic packages that protect them from the environment. There are many different types of packages, such as plastic or ceramic dual in-line packages, leadless chip carries, and pin-grid arrays. Since it is possible for a given circuit to be packaged in many different ways, there is a great proliferation of product types at this stage. Once the leads have been attached and the package sealed and tested for leaks and other defects, the product is sent to final test.

- **Final Tests**

The goal of the testing process is to ensure that customers receive a defect free product by using automated testing equipment to interrogate each integrated circuit and determine whether it is operating at the required specifications. An important characteristic of the testing process from a production planning standpoint is the downgrading or binning that takes place here. A circuit, when tested, may not meet the specification it was originally built for, but may meet another less rigorous one [2]. For example, a microprocessor intended to operate at 2.4 GHz may fail at that frequency but may pass test at 2.0 GHz. Thus when a lot is tested a number of different grades of product may emerge, resulting in not enough of the desired product being available and unwanted inventory of the lower grade product.

References

[1] Groover, Mikell P., "Fundamentals of Modern Manufacturing: Materials, Processes, and Systems", John Wiley & Sons, Inc., 1996, ISBN: 0133121828.

[2] Uzsoy, R., Lee, Chung-Yee, and Martin-Vega, L., "A review of production planning and scheduling models in the semiconductor manufacturing, Part I: system characteristics performance evaluation and production planning", IIE transaction on scheduling and logistics, Vol. 26, pp 44 – 55, 1994.

[3] Runyan, W. R., and Bean, K. E., "Semiconductor Integrated Circuits Processing Technology", Addison-Wesley, 1990.

[4] Atherton, L. F., and Atherton, R. W., "Wafer Fabrication: Factory Performance and Analysis", Kluwer Academic, Boston, 1995.

[5] Sze, S. M., VLSI Technology, McGraw-Hill, New York, 1983.

Appendix B

General Flow Shop Scheduling

1. Introduction

The flow shop scheduling is perhaps the most common form of industrial scheduling. Numerous combinatorial optimization procedures have been proposed for solving the general flow shop problem with the maximum flow time criterion. Many researchers have been successful in developing solution algorithms for flow shop scheduling and sequencing [1]. The beginning was Johnson [8] trying to optimize two and three machines, followed by many researchers trying to solve larger problems (e.g. Campbell [13], and Baker[9]). Many resarechers developed algorithms tryingo to solve problems up to 10 machines (e.g. [4] and [14]). Dannenbring [15] found that for small size shop problems his heuristic outperformed others in minimizing the make-span for 1280 flow shop scheduling problems. Ezat and El Baradie carried a simulation study for pure flow shop scheduling with make-span minimization as a major criterion for number of jobs $n \leq 90$ to be processed on number of machines $m \leq 90$ [12].

In this appendix, a computational study has been developed to obtain optimal / near optimal solution for general flow shop scheduling problem with make-span minimization as the primary criterion and the minimization of either the mean completion time, total waiting time or total idle time as the secondary criterion. The objective is to determine a sequence of operations in which to process 'n' jobs on 'm' machines in same order (flow shop environment) where skipping is allowed. The Simulation approach for deterministic and stochastic flow shop scheduling has been developed. It reads and manipulates data for 500 jobs on 500 machines. Phase 2 of the simulation model presents heuristic technique (Dispatching rules) with different factorial experiments in a comparative study on the performance of different dispatching rules, such as FCFS, SPT, LPT, SRPT and LRPT with respect to the objectives of minimizing make-span, mean flow time, waiting time of jobs, and idle time of machines. Moreover, comparison between the enumerative technique and the dispatching rules based on computational time and the optimum make-span has been issued.

The proposed model is evaluated and found to be relatively more effective in finding optimal/ near optimal solutions in many cases. The influence of the problem size in computational time for this model is discussed.

The purpose of this study is twofold:

4) To provide a simulation model able to find the optimum / near optimum sequence for general flow shop scheduling problem with make-span minimization as main criteria;
5) To compare computational time to obtain feasible solution in two different solving approaches.
6) To examine different dispatching rules on minimizing multiple criteria.

2. General Flow Shop Scheduling Problem

The general flow shop problem consists of two major elements: (1) a production system of '*m*' machines; and (2) a set of '*n*' jobs to be processed on these machines. All '*n*' jobs are so similar that they have essential the same order of processing on the '*M*' machines, Figure B.1. The focus of this problem is to sequence or order the '*n*' jobs through the '*m*' machine(s) production system so that some measure of production cost is minimized [16].

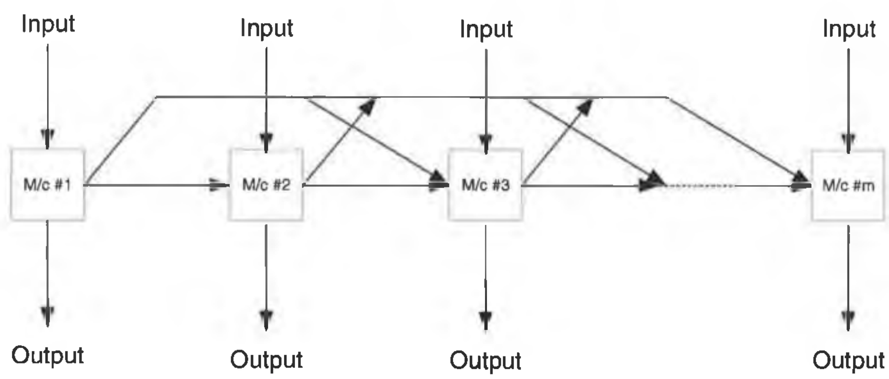


Figure B.1: Work flow in General Flow Shop Scheduling Model

2.1 Problem Assumptions

The assumptions of the flow shop problem are well documented in the production research literature (e.g. Baker [9] [59], French [4]). In summary:

1) All '*n*' jobs are available for processing, beginning on machine number 1, at time zero.
2) Once started into the process, one job may not pass another, but must remain in the same sequence position for its entire processing through the '*m*' machines.
3) Each job may be processed on only a single machine at one time, so that job splitting is not permitted.
4) There is only one of each type of machine available.
5) At most, only one job at a time can be processed on an individual machine.
6) The processing times of all '*n*' jobs on each of the '*m*' machines are predetermined.
7) The set-up times for the jobs are sequence independent so that set-up times can be considered a part of the processing times.
8) In-process inventory is allowed between consecutive machines in the production system.
9) Non-preemption; whereas operations cannot be interrupted and each machine can handle only one job at a time.
10) Skipping is allowed in this model.

2.2 Performance Criteria

The performance criteria discussed in chapter two, and Appendix D provides a classification of main performance criteria. The performance criteria used in this study are those most commonly used as reported by Stafford [17], for optimizing the general flow shop model in particular in phase two of the model.

1. Make-span

Throughout the half century of flow shop scheduling research, the predominant objective function has been to minimize make-span [17]. The formula used to calculate the make-span is shown in Table B.1. Minimizing make-span has been taken as the primary criterion for the simulation model. The expression used is as follows: *Minimize:* C_{max}

2. Mean Completion Time

Many researchers (e.g. Conway *et al.* [3], Panwalker *et al.* [6], Pinedo[7]) have discussed mean job completion time or mean flow time as an appropriate measure of the quality of a flow shop scheduling problem solution. Mean job completion time can be expressed as follows:

$$\overline{C} = \sum_{i=1}^{n} \textit{Job completion times} / n$$

3. Total Waiting Time

Minimizing total job idle time, while the jobs wait for the next machine in the processing sequence to be ready to process them, may be expressed as follows:

$$W_{(nxm)} = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij}$$

4. Total Idle Time

Overall all machine idle time will be considered in this model (the time that machines *2,...,* M spend waiting for the first job in the sequence to arrive will be counted). Overall machine idle time may be minimized according to the following expression:

$$\textit{Minimize:} \sum_{1=1}^{m} \sum_{j=1}^{n} X_{ij}$$

3. General Flow shop Simulation Model

Simulation study for general flow shop scheduling problem with make-span minimization as primary criteria for $n \leq 250$ and $m \leq 250$ with different ranges of random numbers generated (0-99) for processing times matrix in stochastic models has been conducted into two phases:

- *Phase (1)* to find the optimum/near optimum solution for general flow shop problem to minimize the make-span;
- *Phase (2)* to measure the effectiveness of heuristic approach (dispatching rule) for flow shop scheduling and compare the performance of different rules.

Flow shop scheduling problem considers as an optimization problem imposed to sequencing constraints. The proposed model has been developed of computer program using C language (MS Visual C ++) and the full coded program is shown in Appendix E and F. The model runs on Pentium III PC (300MHz) and 128 MB RAM.

The simulation model solves large sequencing problems using exact enumerative techniques. The main objective in the first phase is to find the optimum/ near optimum as it sweeps through all possible feasible schedules and select the optimum sequence of operations. The second phase tends to examine heuristic approach (dispatching rules) using a comparative study with phase one results. For better quality, a number of reptitions of each simulation run to be set.

A summary of model parameters, decision variables, and throughput time (TPT) formula used to calculate the make-span, performance criteria, and dispatching rules under study in phase two has been shown in Table B.1.

3.1 Phase 1: Simulation Model

This phase has multi-objectives for $n/m/F/C_{max}$ problem. It can provide the followings:

1) All the job sequences and their correspondent make-span for each sequence.
2) The optimal job sequence and its make-span value.
3) Frequencies for all job sequences.
4) CPU time for the solution.

Table B.1: Summary of Terminology associated with Simulation Model

Model parameters

Model $n/m/F/C_{max}$

n number of jobs to be processed,

m number of machines (processing steps),

F general flow shop scheduling problem,

C_{max} the criterion is Make-Span,

P_{ij} processing time off job 'j' on machine 'i',

$\overline{C}$ the criterion is Mean Completion Time,

w_{ij} waiting time before start the job 'j',

X_{ij} idle time of machine 'i' before start job in position j in the sequence,

$W_{(nxm)}$ total waiting time.

Dispatching Rules

FCFS : First Come First Served
SPT : Shortest Processing Time
LPT : Longest Processing Time
SRPT : Shortest Remaining Processing Time
LRPT : Longest Remaining Processing Time

Decision Variables

$n \geq 0$ i = 1, ..., n,
$n \leq 250$ (recommended)
(the model can read up to 500 jobs),
$m \geq 0$ j = 1, ..., m,
$m \leq 250$ (recommended)
(the model can read up to 500 machines),
Number of runs,
Number of seed.

Objective functions

1) Make-span

Minimize: C_{max}

2) Mean Completion Time

Minimize: $\overline{C} = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{m}(w_{ij} + P_{ij})$

3) Total Waiting Time

Minimize: $W_{(nxm)} = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}$

4) Total Idle Time

Minimize: $\sum_{1=1}^{m}\sum_{j=1}^{n} X_{ij}$

The Make-span formula used in the model is given as:

$$C_{\max} = \max[f\{q(n-1,m),m\}, f\{q(n,m),m-1\}] + t\{q(n,m),m\}$$

The first step in the simulation model is the determination of the problem type that will be solved. The model can handle both deterministic and stochastic problems. The input processing times are generated from different seed random numbers (0-99) for each run for stochastic models or may be read directly from an input file (in.dat) for deterministic models.

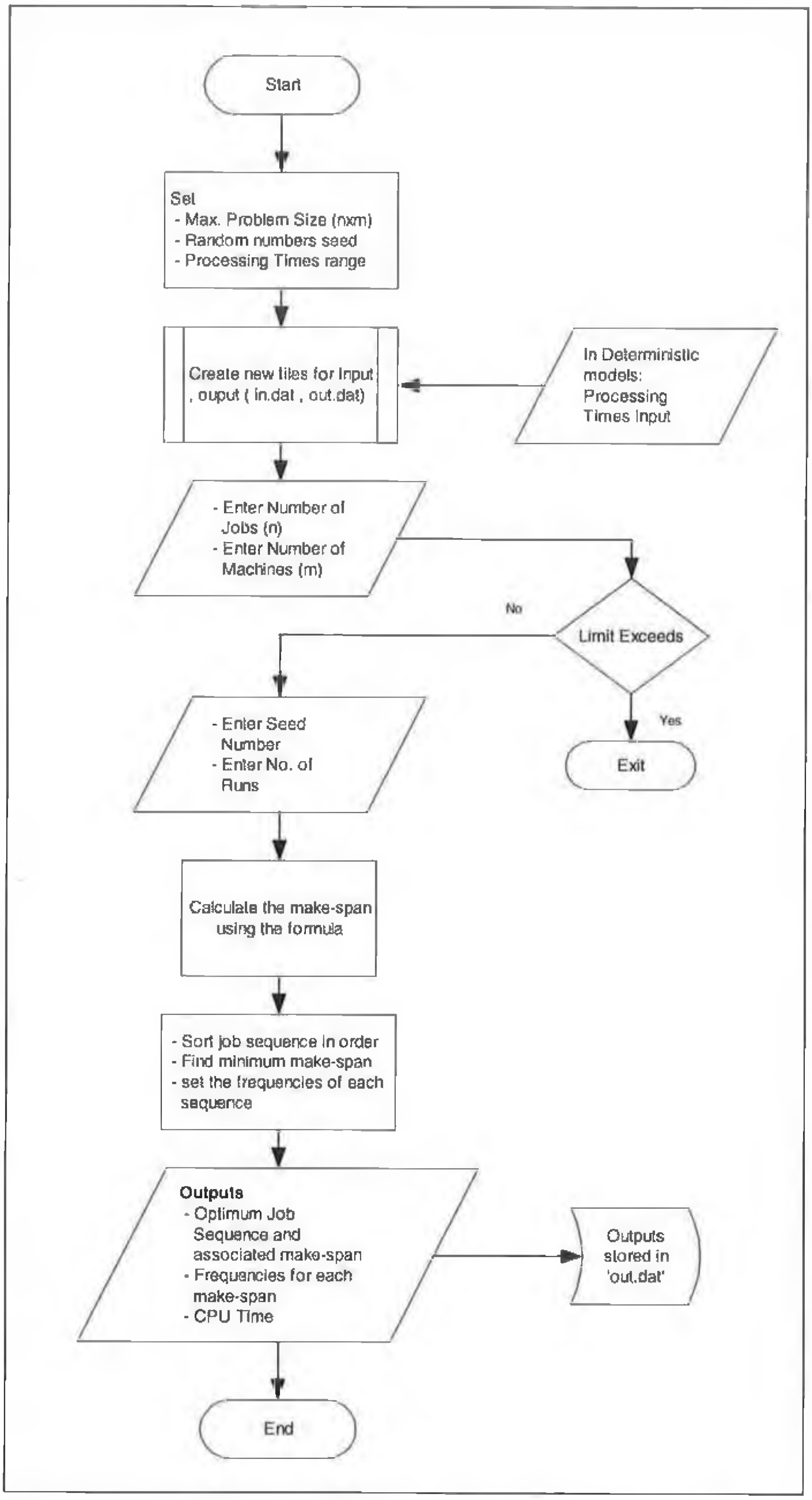


Figure B.2: Flowchart of simulation model (Phase 1)

The processing times range from 0-10 time units as a default but can be set to any different range based on the particular problem. Once the type of model to be evaluated has been determined, the size of the problem (number of jobs 'n' and number of machines 'm') should be given. It is recommended to use a range of $n \leq 250$, and $m \leq 250$.

Following the determination of the model type and size, the model starts to set the main variables (number of machines, number of jobs) and then the number of runs (replications) in case of stochastic models. The model uses TPT formula in Table B.1 to calculate all the possible feasible sequences of jobs. Meanwhile, the frequency of every make-span and the optimum sequence and its corresponding make-span are provided as model outputs. The output of the model comes into two forms; (1) displays on the screen and (2) saves in text file (out.dat). The flowchart (Figure B.2) shows the model steps from the beginning to the output.

3.1.1 Experimentation

Different factorial sets of experiments were conducted to verify that the program would provide optimal solutions to general flow shop problems, a sample of the output of the program and optimal make-span are shown in Figure B.4. Four sets of problems were presented as examples. These sets are characterized by {D(4 x 4), D(8 x 8), S(10 x 75), S(11 x 35)}, where the first terms in the braces represent the number of jobs and the second is the number of machines, while D stands for deterministic and S denotes a stochastic.The results are shown in Figures B.5 to B.8 respectively. The processing times in stochastic models generated from a random seed set of numbers (45, 25). The number of runs for each case is set to be 300 where the results turn to steady state started in phase two as shown in Figure B.3.

Although, many researchers have been working on the flow shop scheduling problem for many years, very few results has been found about the distribution of the objective function. In effect, such presentation gives an intuitive idea about the problem and is important to allow the reader to judge the quality of the solution [18]. The distributions of all the possible make-spans obtained by complete enumeration of two different problems are given in Figures B.5 to

B.8. The processing times were randomly generated (integers between 1 and 10). The distribution seems to be almost symmetric and its range is contained in an interval of 20% around the mean. A χ^2 test does neither confirm nor refute that this distribution is Gaussian; therefore, the use of the mean make-span given by a heuristic seems to be meaningful.

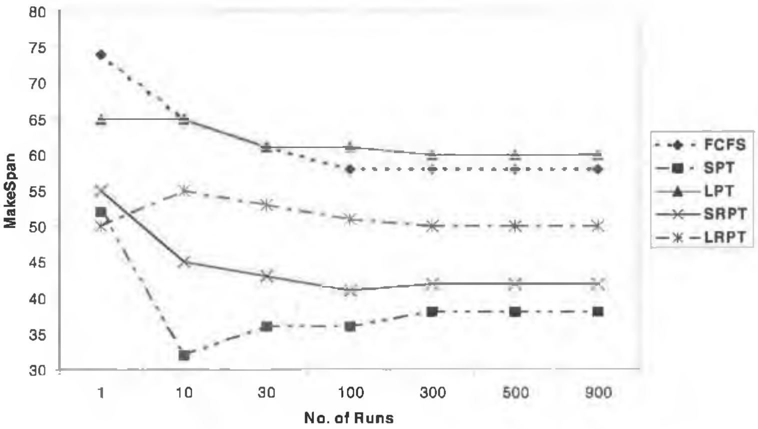


Figure B.3: A Steady-State Analysis of the Model

```
5 jobs on 7 machines

   n\ m   M1   M2   M3   M4   M5   M6   M7
   J1     8    9    7    8    6    2    8
   J2     3    5    9    7    5    2    4
   J3     9    1    8    0    10   8    2
   J4     3    8    10   2    3    0    0
   J5     4    2    6    8    10   1    0
```

Input matrix of processing times

```
5 Jobs, 7 Machines, Random seed=48
    Number of runs           = 1
```

Job sequences

```
12345 Make-Span=75
12354 Make-Span=75
12435 Make-Span=75
24153 Make-Span=71
24513 Make-Span=72
24531 Make-Span=67
31524 Make-Span=66
34125 Make-Span=66
34152 Make-Span=64
41532 Make-Span=66
45312 Make-Span=61
45321 Make-Span=63
54132 Make-Span=66
51342 Make-Span=64
51432 Make-Span=66
. . . .
. . . . .
Number of Job Sequences = 120

53421  = Optimum Job Sequence, Optimal Make-span = 60
CPU time in seconds =0.211000

Make-Span's Frequencies:
```

The entire possible job sequences

```
For the Make-Span =60: the Frequency =3
For the Make-Span =61: the Frequency =3
For the Make-Span =62: the Frequency =1
For the Make-Span =63: the Frequency =3
For the Make-Span =64: the Frequency =15
For the Make-Span =66: the Frequency =15
For the Make-Span =67: the Frequency =20
For the Make-Span =68: the Frequency =6
For the Make-Span =70: the Frequency =1
For the Make-Span =71: the Frequency =39
For the Make-Span =72: the Frequency =1
For the Make-Span =73: the Frequency =3
For the Make-Span =75: the Frequency =10
53421= Optimum Job Sequence, Optimal Make-Span = 60
```

The frequency of each make-span

Optimal Job Sequence

Figure B.4: A sample of the program output

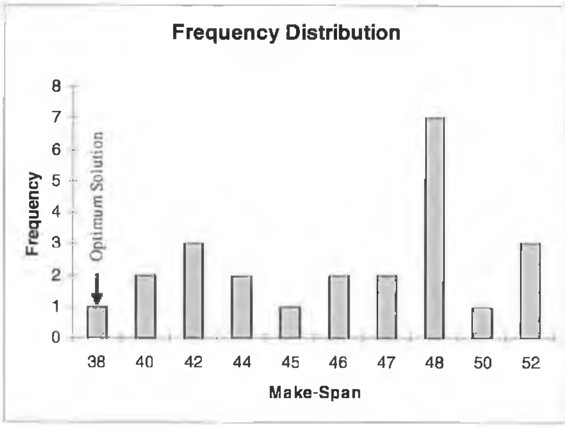
4 jobs on 4 machines

Processing times are represented in the following table:

	M1	M2	M3	M4
J1	5	8	8	6
J2	5	5	5	5
J3	6	10	1	6
J4	9	6	0	4

4 Jobs, 4 Machines, Random seed=0
Number of runs = 1

1234Make-Span=42
1243Make-Span=38
1324Make-Span=46
1342Make-Span=48
1432Make-Span=48
1423Make-Span=40
2134Make-Span=42
2143Make-Span=42
2314Make-Span=45
2341Make-Span=48
2431Make-Span=47
2413Make-Span=44
3214Make-Span=50
3241Make-Span=52
3124Make-Span=46
3142Make-Span=48
3412Make-Span=48
3421Make-Span=52
4231Make-Span=47
4213Make-Span=44
4321Make-Span=52
4312Make-Span=48
4132Make-Span=48
4123Make-Span=40
Number of Job Sequences = 24



1,2,4,3 = Optimum Job Sequence, Optimal Make-span = 38

CPU time in seconds =0.020000

Make-Span's Frequencies:
For the Make-Span =38: the Frequency =1
For the Make-Span =40: the Frequency =2
For the Make-Span =42: the Frequency =3
For the Make-Span =44: the Frequency =2
For the Make-Span =45: the Frequency =1
For the Make-Span =46: the Frequency =2
For the Make-Span =47: the Frequency =2
For the Make-Span =48: the Frequency =7
For the Make-Span =50: the Frequency =1
For the Make-Span =52: the Frequency =3

1,2,4,3 = Optimum Job Sequence, Optimal Make-Span = 38

Figure B.5: Example '1' Deterministic model (4 jobs x 4 machines)

```
8 Jobs, 8 Machines,
Random seed=0
Number of runs            = 1
Number of Job Sequences = 40320

6,5,4,3,8,7,1,2  = Optimum Job
Sequence,

Optimal Make-span = 67
CPU time in seconds =61.519000
N.B:
The Best results from LEKIN was 77
unit time and the sequence
was 6 2 1 3 7 5 4 8
```

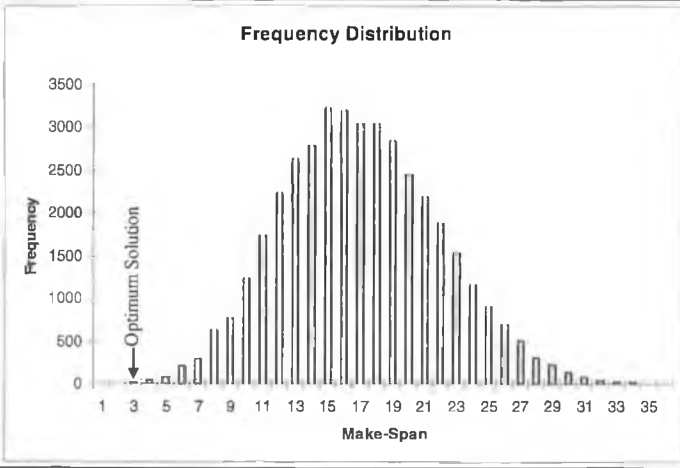


Figure B.6: Example '2' Deterministic model (8 jobs x 8 machines)

```
10 Jobs, 75 Machines,
Random seed=45
Number of runs            = 300
Number of Job Sequences = 3628800

3,5,7,8,2,4,9,10,1,6  = Optimum Job
Sequence, Optimal Make-span = 114

CPU time in seconds =1555.547000

N.B: Processing times are
represented generated from the
random seed 45 and their values
between 0-10
```

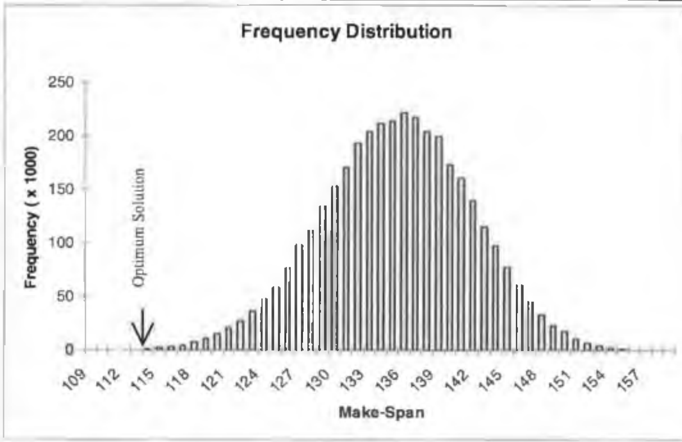


Figure B.7: Example '3' Stochastic model (10 jobs x 75 machines)

```
11 Jobs, 35 Machines,
Random seed =25
Number of runs            = 300
Number of Job Sequences = 39916800

5,10,3,11,1,4,8,7,9,2,6  = Optimum
Job Sequence,
Optimal Make-span = 125

CPU time in seconds =13179.451000

N.B: Processing times are
represented generated from the
random seed 25 and their values
between 0-10.
```

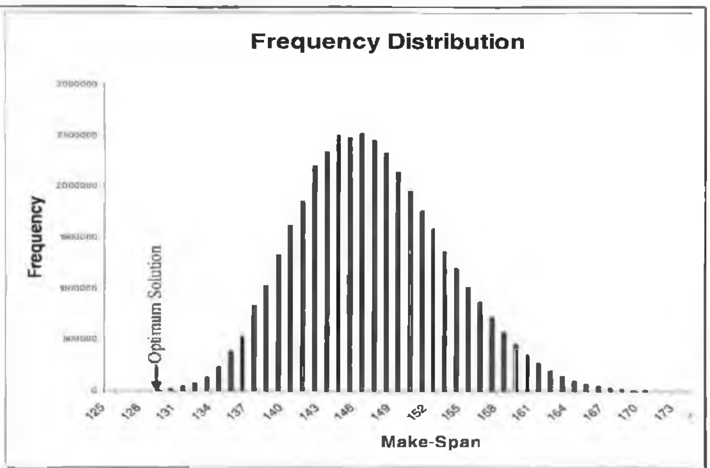


Figure B.8: Example '4' stochastic model (11 jobs x 35 machines)

3.1.2 Verification and Validation

The simulation model has been verified in three ways, first by comparing with LEKIN scheduling software package [38], although this is limited as LEKIN cannot handle shop problem higher than $n \geq 10$ and $m \geq 18$. Second, LINDO linear programming software has also been used to check a sample of the output. The third checks the output through a trace file that consists of detailed output representing the step-by-step progress of the simulation model over the simulated time. This allows detection of subtle errors.

Once the model was verified, the next step was validation using other proven techniques. In general, the quality of a technique's solutions is measured in at least two dimensions: (1) how close the solution comes to the optimal solution if it can be measured; and (2) how much computer time is required to solve problems of a given size.

Table B.2: Comparison between some different studies

Method	*Avg. % increase over Optimum*	*Limitations*
Palmer, 1965	10 % - 35 %	Optimum achieved in 30% of cases Small scale problems only
Campbell, 1970 [13]	5 % – 20 %	Optimum is not guaranteed Economical $n \leq 8$,
Dannenbring, 1977 [15]	5 % – 15 %	Optimum achieved in 35% of cases $n \leq 6$, $m \leq 10$ only
Gupta, 1971 [14]	10 % - 20 %	Optimum is not guaranteed
Al-Qattan, 1990 [19]	0 % - 15 %	Optimum is not guaranteed
Ezat – El Baradie, 1993 [12]	0 % - 10 %	Optimum is for $n \leq 12$ $m \leq 60$ Pure flow shop scheduling problems only Max. size $n \leq 90$, $m \leq 90$
Tsang – Stafford, 2001 [16]	0 % - 5 %	Optimum is guaranteed for $n \leq 7$, $m \leq 7$
LEKIN, 1998 [7]	0 % - 10 %	Optimum is not guaranteed Max. size $n \leq 10$, $m \leq 18$
Arisha – El Baradie, 2001 [1]	0 % 0 % - 10 %	Optimum is guaranteed for $n \leq 50$, $m \leq 250$ General flow shop scheduling problems only Max. size $n \leq 500$, $m \leq 500$ For $n \geq 50$, $m \geq 250$

Due to wide differences in software, platform, problem size, experimental design and reporting, it is very difficult to compare the performance of different techniques directly. To allow some comparison to be made, Table B.2

shows the average percentage increase over optimum make-span time as reported by each of the researchers for their algorithms. To enhance the comparison, the right column indicates the relative limitations of each model.

3.1.3 Results Analysis

Due to the advent of computer technology, the optimum solutions to small and even relatively medium-sized problems can be found in reasonable computational time. The particular attraction of this enumerative technique is that can highly efficient get the optimum sequence. However, this approach is uneconomical and partial enumeration and other algorithms can offer more applicable means to find optimum or near optimum solutions.

The computational times for the different factorial experiments are shown in Table B.3. The exponential growth in computational time with the number of jobs is central to the difficulty of using this approach for larger problems. Figure B.9 plots the computational times on logarithmeic scales to show the range.

Table B.3: CPU time (seconds) to find optimum make-span for different problem sizes

n x m	5	10	20	40	100	250
5	0.161	0.18	0.17	0.181	0.21	0.22
6	1.1	1.1	1.2	1.2	1.3	1.3
7	8.1	8.2	8.4	8.4	8.8	9.0
8	40.3	43.3	44.9	45.6	44.3	45.2
9	232.1	232.6	245.0	235.3	245.1	248.1
10	1158.0	1176.3	1256.4	1307.3	1354.2	1298.3
11	13115.3	13118.3	14234.7	13215.2	13684.3	13968.3
12	56025.3	56036.9	57231.3	56016.7	57863.3	58015.2
20	-	551369.4	-	543652.4	-	-
30	-	-	-	2605248.0	-	-

3.2 Phase 2:

The explosive growth in computational times needed to find the optimum solution for larger problems means that the enumerative approach is no longer economic. Heuristic approaches attempt to find feasible solution (near optimum) in less computational time. One of the common heuristic techniques is using dispatching rules (priority rules) in order to reduce the solution space and hence the computational time.

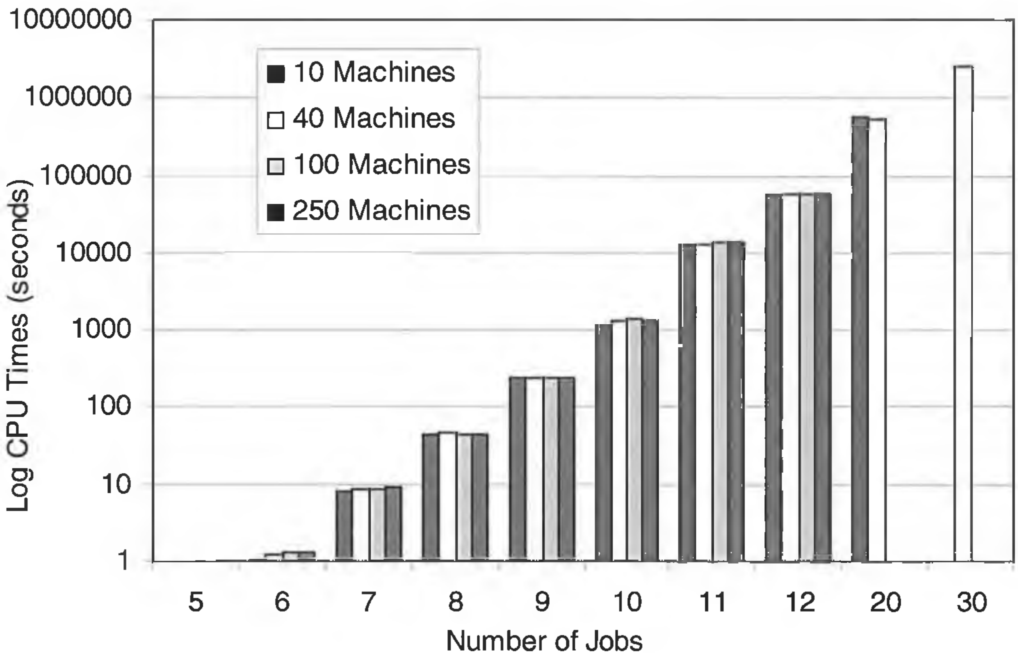


Figure B.9: CPU Times vs. different Number of Jobs

This phase two of the model shares the same main characteristics of phase one. It deals with problems up to $n \leq 250$, $m \leq 250$ and examines five different dispatching rules. The output of each is the expected value of the criterion function, which results when the rule is followed [11].

The second phase of the simulation model has multi-objectives '*n/m/F/X* problem'; it can also provide the followings:

1) Comparison of Output based on each criterion.
2) CPU time for the solution.

3.2.1 Dispatching Rules

A dispatching rule is used to select the next job to be processed from a set of jobs awaiting service at a facility that becomes free. The difficulty of the choice of a dispatching rule arises from the fact that there are (*n!*) ways of sequencing '*n*' jobs waiting in the queue at a particular facility and the shop floor conditions elsewhere in the shop may influence the optimal sequence of jobs at the present facility[20].

The dispatching rules are used mainly in two areas in the production line; inventory and shop floor. A vast body of literature discussed the dispatching rules extensively (e.g. Pinedo *et al.* [7][43], Blackstone *et al.* [21], Haupt [22]). The dispatching rules varied from very simple to extremely complex rules and they have normally many objectives, some of them are to:

1) Minimize make-span,
2) Minimize waiting time/cost,
3) Maximize machine utilization,
4) Improve the production flow, and
5) Decrease delay.

Despite of the fact that no dispatching rule has been demonstrated to be optimal for general shop scheduling, the use of dispatching rules can help to find a satisfying solution in many applications and optimal solution in other applications[2].

3.2.2 Selected Dispatching Rules

The need for studying dispatching rules arises and the comparison between them is essential to find the best rule for the application under study. Five popular basic dispatching rules have been selected to be investigated in this research as reported by Mattfeld [23].

- Rule (1) FCFS (First Come First Served): This rule dispatches jobs based on their arrival times or release dates. The job that has been waiting in queue the longest is selected. The FCFS rule is simple to implement and has a number of noteworthy properties. For example, if the processing times of the jobs are random variables from the same distribution, then the FCFS rule minimizes the variance of the average waiting time. This rule tends to construct schedules that exhibit a low variance in the average total time spent by the jobs in the shop.

- Rule (2) SPT (Shortest Processing Time): The SPT rule minimizes the sum of the completion times ΣC_j (usually referred as the flow time), the number of jobs in the system at any point in time, and the average number of jobs in the system over time for the following machine environments:
    - Set of unique machines in series.
    - Bank of identical machines in parallel.
    - Proportionate flow shop.

- Rule (3) LPT (Longest Processing Time): The LPT rule is particularly useful in the case of a bank of parallel machines where the make-span has to be minimized. This rule selects the job with the longest processing (from the queue of jobs) to go next when a machine becomes available. Inherently, the LPT rule has a load balancing property, as it tends to avoid the situation where one long job is in process for long time. Therefore, after using the LPT rule to distribute the jobs among the machines, it is possible to re-sequence the jobs for the individual machines to optimize another objective besides make-span. This rule is more effective when preemption is allowed.

- Rule (4) SRPT (Shortest Remaining Processing Time): The SRPT is a variation of SPT that is applicable when the jobs have different release dates. SRPT rule selects operations that belong to the job with the smallest total processing time remaining. It can be effective in minimizing the make-span when preemption is allowed.

- Rule (5) LRPT (longest Remaining Processing Time): The LRPT is a variation of LPT that selects the operations that belong to the job with the largest total processing time remaining. LRPT rule is of importance when preemption is allowed and especially in parallel identical machines. LRPT rule always minimizes the idle time of machines.

3.2.3 Experiments

The same steps in phase one were carried out in this phase. A different factorial experiment for the selected rules has been set to include a wide range of industrial shops beginning with a simple shop floor of 5 machines up to complex industrial shops (such as semiconductor manufacturing) with 200 machines or more. The model examines seven machine shops (5, 20, 50, 80, 130, 200, 250) and nine different loading conditions (number of jobs) equal to (5, 10, 30, 50, 80, 100, 150, 200, 250). A sample of phase two output is shown in Figure B.10.

Number of Runs

The number of runs for each case is set to be 300 for every stochastic run, where the results turn to steady state started as shown in Figure B.3.

3.2.4 Results

In order to obtain a wide range of shop scheduling problems, the full factorial experiments have been run. The detailed results are given in Appendix G, and only summaries of the results are shown in this section.

- **Average Make-Span Criterion**

For small and medium machine numbers, Figure B.11 and B.12, there is a clear spread across the different rules with the SPT rule providing the best results and the LPT rule performing worst.

For larger machine numbers, Figure B.13, the LPT rule is still clearly the worst, however the other rules show almost identical results. Nevertheless the SPT rule is the best performer overall.

- **Average Mean Completion Time Criterion**

For this criterion, Figures B.14 – B.16, the SRPT rule provides the best results. Again, the LPT rule performs worst, sometimes rivaled by the LRPT rule.

```
16 Jobs, 33 Machines, Random seed=33
    Number of runs           = 300

Average Make-span

Rule 1 FCFS     = 1029
Rule 2 SPT      = 909
Rule 3 LPT      = 1042
Rule 4 SRPT     = 993
Rule 5 LRPT     = 1014

Average Mean Completion Time

Rule 1 FCFS     = 729
Rule 2 SPT      = 733
Rule 3 LPT      = 784
Rule 4 SRPT     = 782
Rule 5 LRPT     = 751

Average Total Waiting Time

Rule 1 FCFS     = 3120
Rule 2 SPT      = 2708
Rule 3 LPT      = 2457
Rule 4 SRPT     = 2856
Rule 5 LRPT     = 2564

Average Total Idle Time

Rule 1 FCFS     = 6087
Rule 2 SPT      = 6274
Rule 3 LPT      = 8262
Rule 4 SRPT     = 4896
Rule 5 LRPT     = 7351


CPU time in seconds = 2.912088
```

The Output based on Make-span criteria

The Output based on Mean Completion Time criteria

The Output based on Total Waiting Time criteria

The Output based on Total Idle Time criteria

Total computational Time

Figure B.10: A sample of phase two output

- **Average Waiting Time Criterion**

This criterion changes the order of the rules with LPT performing best and SPT performing worst, Figures B.17 – B.19. In addition increasing the number of jobs increases the spread between best and worst results significantly.

- **Average Idle Time Criterion**

As with the waiting time criterion, Figures (B.20 – B.22) show that the spread of the performance increases with job number. Here the LRPT rule is clearly the best while the LPT rule performs worst.

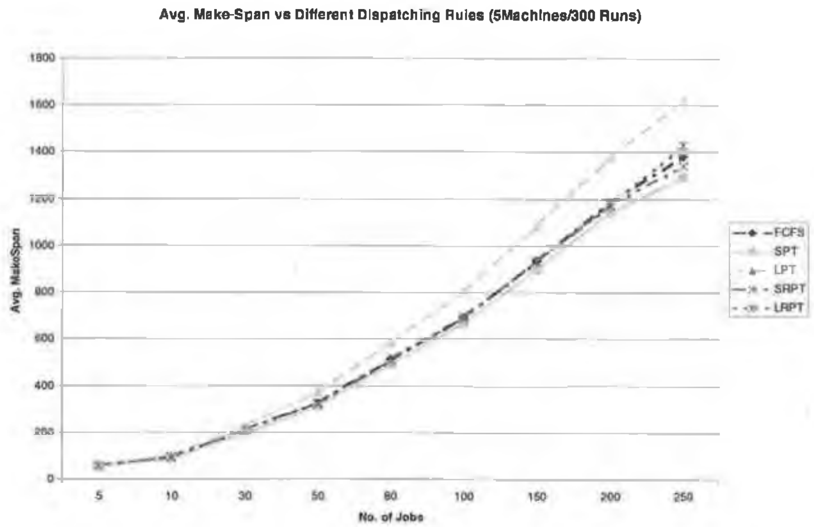


Figure B.11: Five machines shop (make-span criterion)

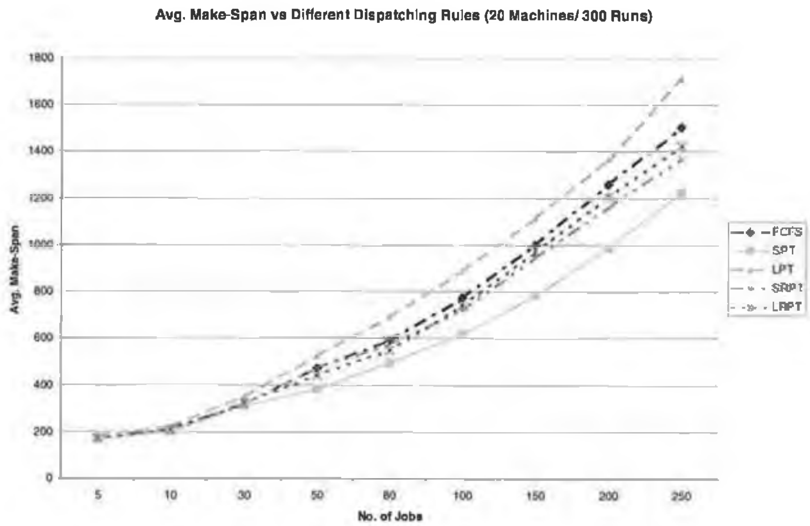


Figure B.12: 20 machines shop (make-span criterion)

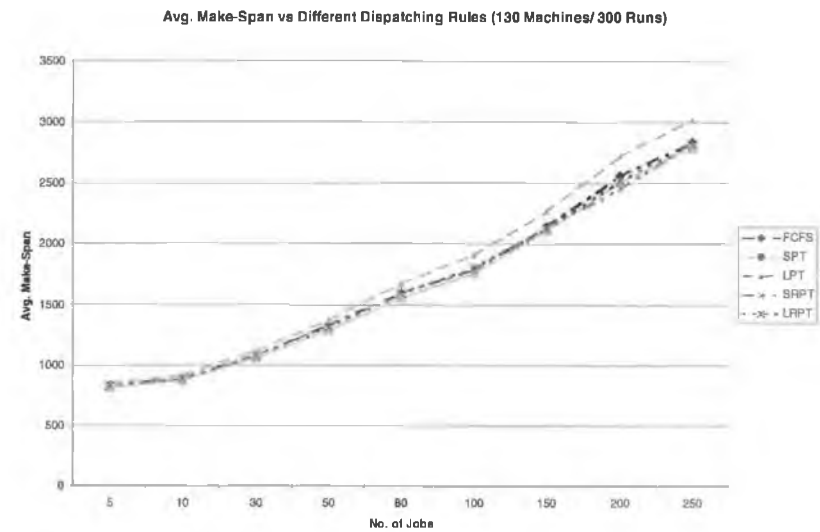


Figure B.13: 130 machines shop (make-span criterion)

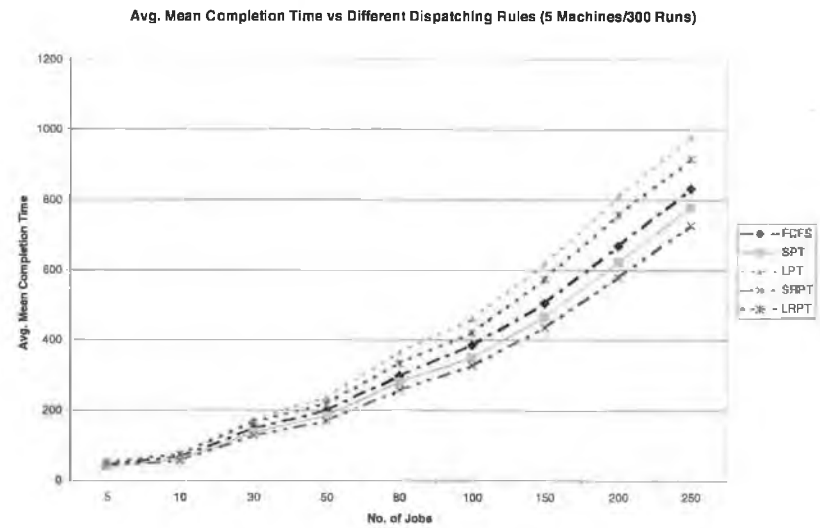


Figure B.14: Five machines shop (mean completion time criterion)

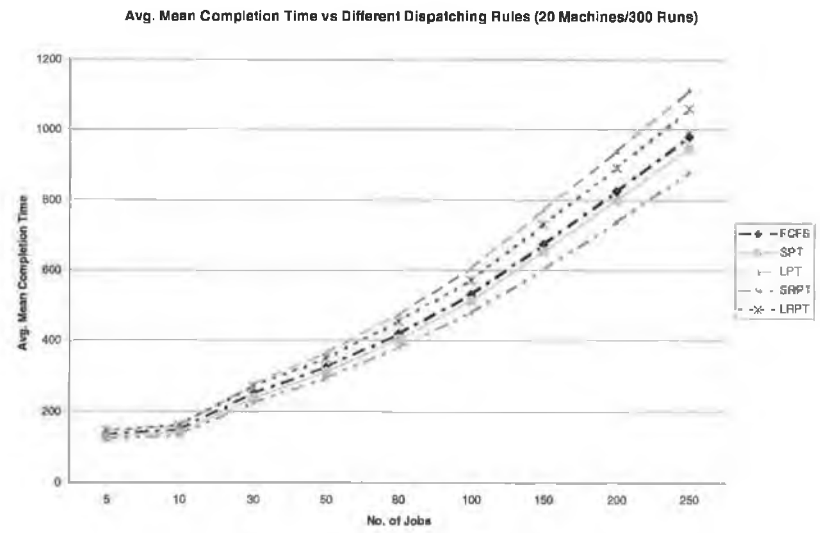


Figure B.15: 20 machines shop (mean completion time criterion)

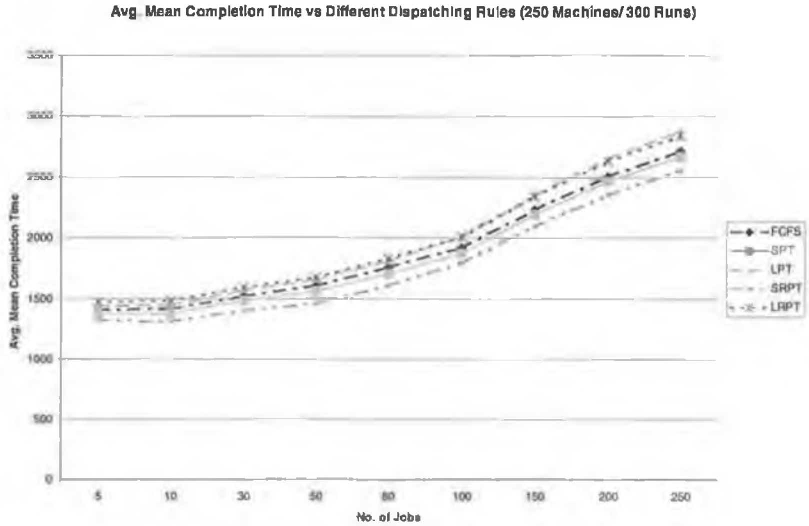


Figure B.16: 250 machines shop (mean completion time criterion)

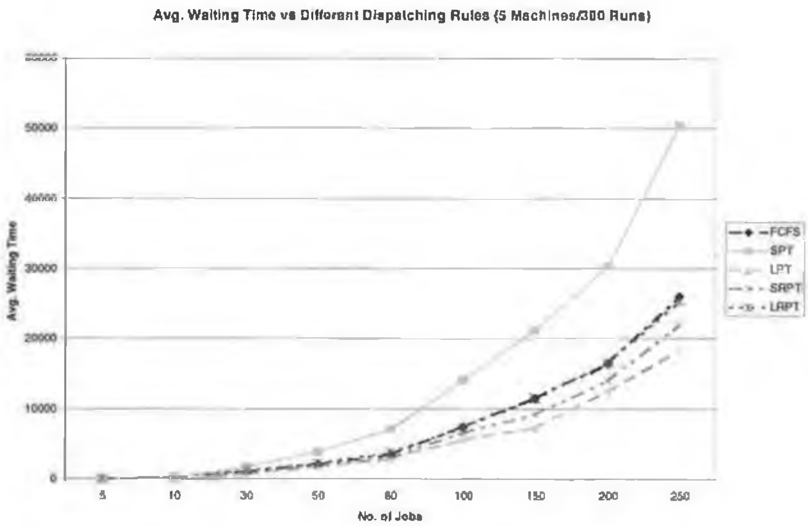


Figure B.17: Five machines shop (jobs waiting time criterion)

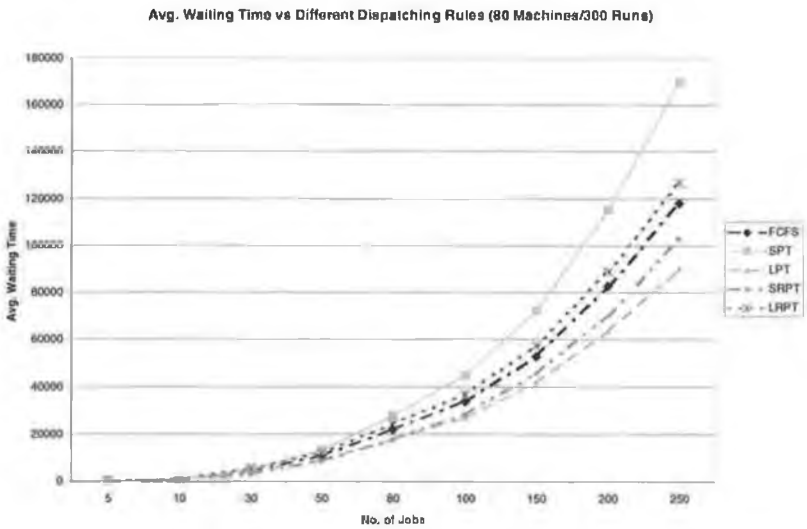


Figure B.18: 80 machines shop (jobs waiting time criterion)

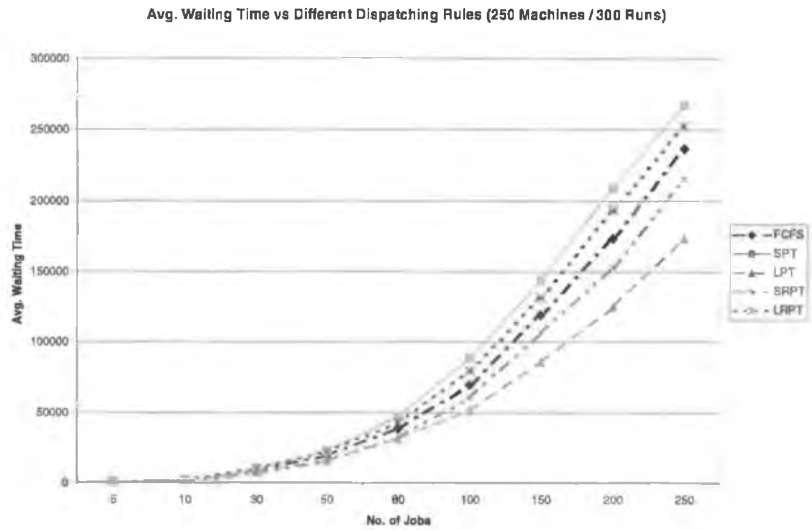


Figure B.19: 250 machines shop (jobs waiting time criterion)

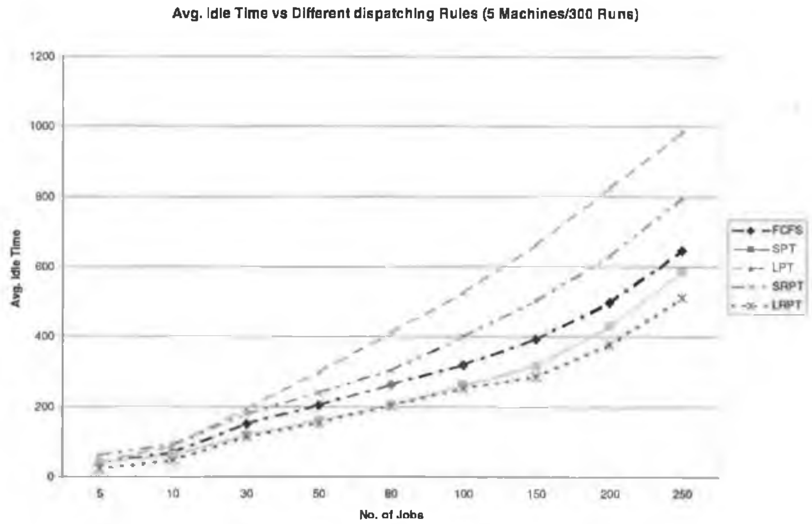


Figure B.20: Five machines shop (machines idle time criterion)

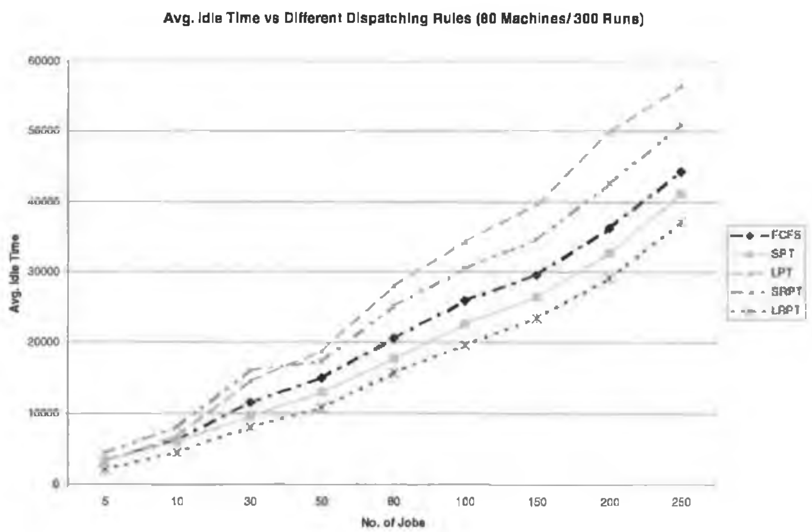


Figure B.21: 80 machines shop (machines idle time criterion)

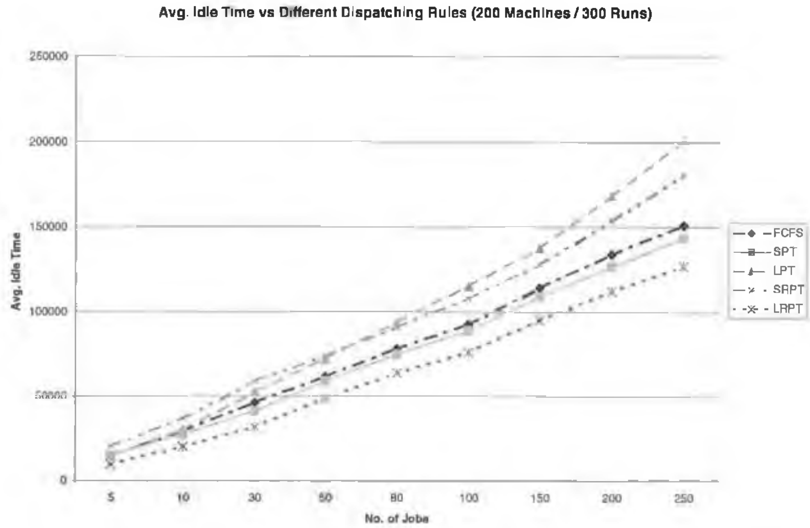


Figure B.22: 200 machines shop (machines idle time criterion)

3.2.5 Results Analysis

The comparative study on the performance of various dispatching rules has been carried out under different shop machine and utilization levels. As mentioned earlier, the model runs for 300 iterations to reach the steady state using the random generator for process times for the same specific shop conditions. It has been observed that no single rule performs well for all criteria related to completion time, waiting time and idle time. SPT has performed the best to minimize make-span under different conditions, as is clearly evident from Figure B.23.

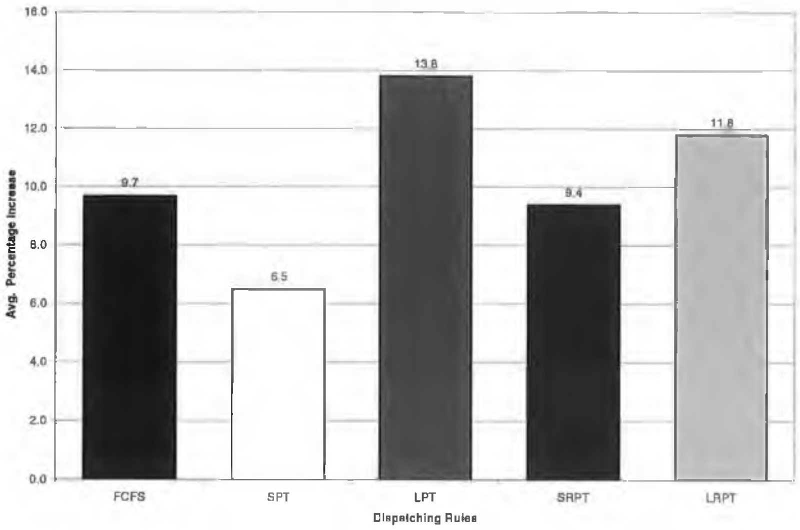


Figure B.23: Comparison of average normalized increase over optimum make-span for different dispatching rules

The SPT rule is quite often used as benchmark since it is found to be very effective in minimizing make-span and mean tardiness under highly loaded shop floor conditions [20]. SPT is the worst for minimizing the job-waiting criterion, while LPT shows the worst performance for make-span criterion; it tends to be the best rule to minimize the job waiting time especially for high utilization level ($n \geq 80$). For the average mean completion time (mean flow time) SRPT shows the best performance for different levels of utilization. LRPT and LPT performed worst for average mean completion time criterion. LRPT rule tends to dominate with respect to the machine idle time while LPT and SRPT showed the worst performance as job number increased.

While never being the best or worst performer for any criterion the FCFS rule is effective in minimizing the maximum flow time and the variance in flow time. Its consistent "mid-table" performance allows its use as a benchmark.

4. Concluding Remarks

- The exponential increase in solution time with number of jobs is shown in Table B.3 for the first phase (enumerative). The length of time for computation means that the full search cannot be economically used where the number of jobs exceeds 30.

- The parameters that affect the size of flow shop problems are 'n' and 'm'. The problem size complexity is based on these two parameters. The results of the proposed model indicated that 'n' has much stronger influence on computer solution time than 'm'. Based on these studies and the proven NP-completeness of the problem, it is clear that 'n' is a much more important determinant of computer solution time required for the flow shop problem.

- Although, researchers have been working on the flow shop sequencing problem for many years, no clear comparative study about the difference of computational time of exact enumerative approach and any heuristic technique has been reported. Table B.4 shows such a comparison between computational time to find the optimal solution in phase 1 (exact enumeration) and to solve the problem using heuristic technique to get near-optimum solution or satisfying solution.

Table B.4: Comparison of computational time

Problem Size (n x m)	***Computational time in seconds***	
	Phase 1	***Phase 2***
5 x 5	0.161	17.25
8 x 20	44.9	18.00
10 x 40	1307.3	19.12
11 x 40	13215.2	19.13
12 x 40	56016.7	19.32
20 x 10	551369.4	19.63
30 x 40	2605248.3	20.45
100 x 100	-	22.42
250 x 250	-	25.96

- Dispatching rule based scheduling is a computationally fast approach [23]. Dispatching rules are useful heuristic for finding a reasonably good schedule with regard to a single objective such as the make-span, mean completion time, mean waiting time, or mean machine idle time. As no signal dispatching rule showed high superiority for all the performance criteria, the combination of several basic rules may be more effective. Added to that, there is also the possibility to switch between different rules based on the shop dynamics.
- Dispatching rules cannot provide the optimal solution in most cases but the difference between the near-optimal solution and the optimal solution might be acceptable when computational time and problem size are taken into account. SPT provides a solution 6.5% from the optimal, while the worst rule (LPT) was 13.8% more than the optimal.

References

[1] Arisha, A., Young, P., and El Baradie, M., “Flow Shop Scheduling Problem: A Computational Study” Proceedings of the Sixth International Conference on Production Engineering and Design for Development (PEDD6), Cairo, Egypt, pp 543 – 557, 2002.

[2] King, J. R., “The Theory-Practice Gap in Job Shop Scheduling”, The Production Engineer, March 1976.

[3] Conway, R. W., Maxwell, W.L. and Miller, I. W., “Theory of Scheduling”, Addison Wesley, Reading, Massachusetts, 1967.

[4] French, S., “Sequencing and Scheduling”, An introduction to the mathematics of the job shop, Ellis Horwood series, 1982.

[5] Pinedo, M., “Operations scheduling with applications in manufacturing and services”, Irwin/McGraw-Hill, 1998. ISBN: 0072897791.

[6] Panwalker, S.S., Iskander, Wafik, “A Survey of Scheduling Rules”. Operations Research, Vol.25, No.1, January-February 1977.

[7] Pinedo, Michael and Bhaskaran, Kumar, “Dispatching”, Handbook of Industrial Engineering, pp 2182-2198, 1992.

[8] Johnson, S. M., “Optimal Two and Three-Stage Production Schedules with Setup Times Included”, Naval Research Logistics Quarterly, Vol. 1, No. 1, 1954.

[9] Baker, Kenneth R., “Sequencing and Scheduling”, John Wiley & Sons Inc.1998, ISBN: 0-9639746-1-0.

[10] Ezat, A. M. and ELBaradie, M., “Flow-Shop Scheduling: Effect of Various Priority Rules on Minimizing Multiple Criteria”, 30th International MATADOR Conference (UMIST), 1993.

[11] Gere, W. S., “Heuristics in Job Shop Scheduling”, Management Science, Vol. 13, No. 3, 1966.

[12] Ezat, A. M. and El Baradie, M., “A Computer Aided-Simulation for the Optimization of the Flow-Shop Scheduling Vs. Different Priority Rules”, Proceedings of the Ninth Conference of the Irish Manufacturing Committee (IMC-9), pp 139-150, 1992.

[13] Campbell, H. G., “A heuristic algorithm for n-job and m-machines sequencing problem”, Management Science, Vol. 16, pp. 630 – 673, 1970.

[14] Gupta, J. N. D., “A functional heuristics algorithm for the flow-shop scheduling problem”, Operation Research, Vol. 22, No. 1, pp. 39 – 48, 1971.

[15] Dannenbring, D. G., “An Evaluation of Flow Shop Sequencing Heuristics”, Mgmt Science, Vol. 23, pp 1174 – 1182, 1977.

[16] Tseng, Fan and Stafford, E., “Two MILP models for N x M SDST flow shop sequencing problem”, Int. J. Prod. Res., Vol. 39, No. 8, 1777 – 1809, 2001.

[17] Stafford, E. F., "On the development of a mixed-integer linear programming model for the standard flow shop". Journal of the Operational Research Society, Vol. 39, pp. 1163 – 1174, 1988.

[18] Taillard, Eric D., "Some efficient heuristic methods for the flow shop sequencing problem", European Journal of Operational Research, Vol.47, pp. 65 – 74, 1990.

[19] Al-Qattan, P. E., "Systematic Approach to Cellular Manufacturing System Design", Elsvier Science Publishers B. V., pp 415 – 424, 1989.

[20] Rajendran, C. and Holthaus, O., "A Comparative study of dispatching rules in dynamic flow shops and job shops", European Journal of Operational Research, Vol. 116, pp.156 – 170, 1999.

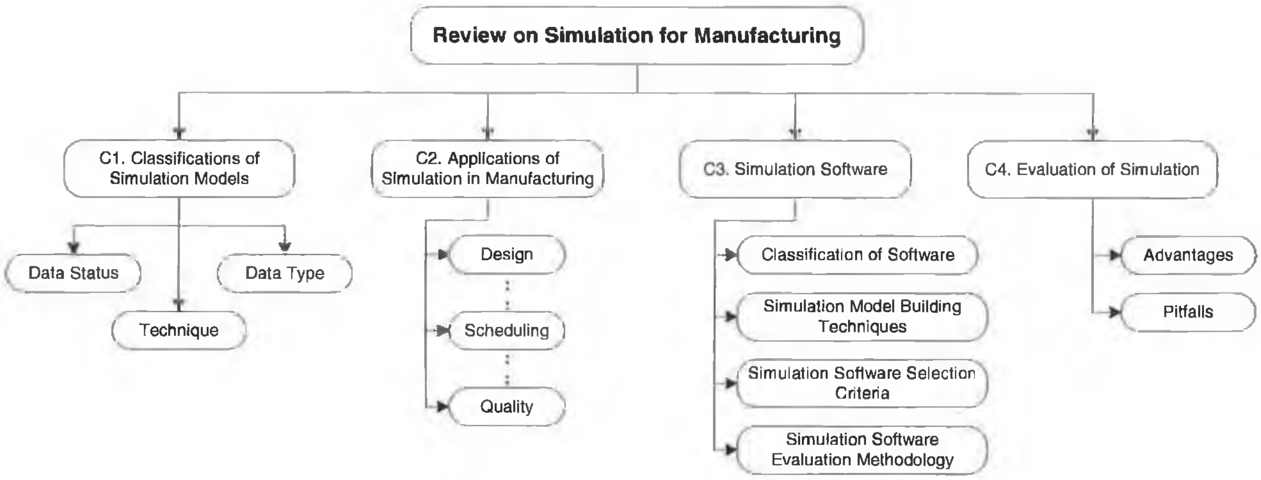
[21] Blackstone, J.H., Phillips, D. T. and Hogg, G.L., "A State-of- the-art survey of dispatching rules for manufacturing job shop operations", Int. Journal of Production Research, Vol. 20, No. 1, pp 27 – 45, 1982.

[22] Haupt, R., "A Survey of Priority Rule-Based Scheduling", OR Spektrum, Vol. 11, pp 3-16, 1989.

[23] Mattfeld, Dirk C., "Evolutionary Search and the Job Shop, investigation on Genetic Algorithms for production scheduling", Physica-Verlag Heidelberg, 1996. ISBN: 3-7908 0917 9

Appendix C

Review on Simulation for Manufacturing



Simulation is the imitation of the operation of a real-world process or system over time by means of analogous situation, either to gain information more conveniently in order to enhance the system performance. According to Shannon [1], Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system.

Simulation involves the generation of an artificial history of the system, and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented. Simulation is an indispensable problem-solving methodology for the solution of many real-world problems and is used to describe and analyze the behavior of a system, ask "what if" questions about the real system, and aid in the design of real systems. Both existing and conceptual systems can be modeled with simulation. A vast body of research efforts has reviewed simulation in manufacturing (e.g. Banks [2], Banks *et al.* [111], Law *et al.* [4], Pritsker [5], Shannon [1], and Kochhar [6]).

C1. Classification of Simulation Models

Simulation models have many ways to be classified (e.g. [1], [3], [4], [8], [9], and [10]). Simulation model can be classified into three groups, Figure C.1.

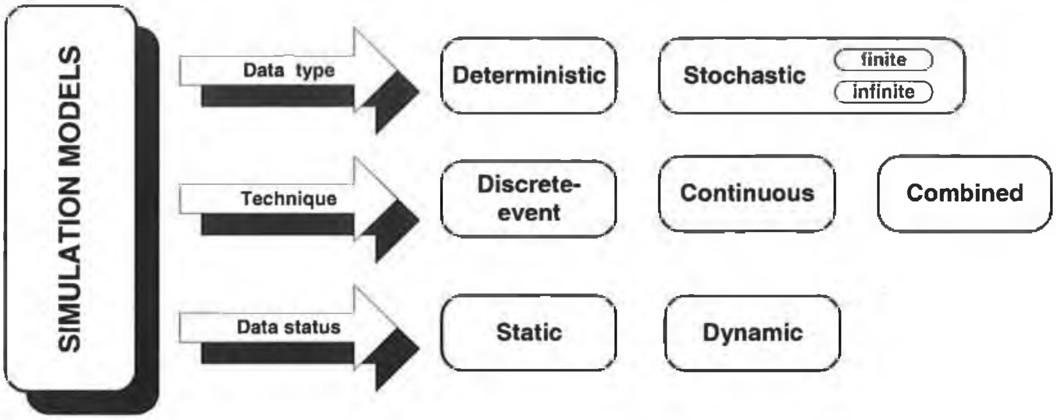


Figure C.1: Classification of Simulation Models

The discrete-event simulation (DES) model and continuous simulation models have been a focus of many comparisons due to their applications in industries. Table C.1 shows a brief comparison between the two types of simulation models.

Table C.1: A comparison between DES model and continuous simulation models

Comp.	*Continuous*	*Discrete-Event*
Time step	Infinite, Model recalculation are sequential and time dependent	Interval is dependent on when events occur.
Method	Differential equations	Logical relationships
Components	Aggregate	Individual entities
Variables	Levels	Queues, states, attributes
Changes	Rates	Events
Ordering	FIFO in most cases	Any described priority rule
Applications	Chemical industries, control systems, System dynamics	Manufacturing, Business process, Networks

C2. Applications of Simulation in Manufacturing

Simulation has turned to be in a real challenging task. Over last three decades, computer simulation as one of the advanced techniques has been applied to various activities in manufacturing systems such as process planning, maintenance and diagnosis, scheduling, and quality management.

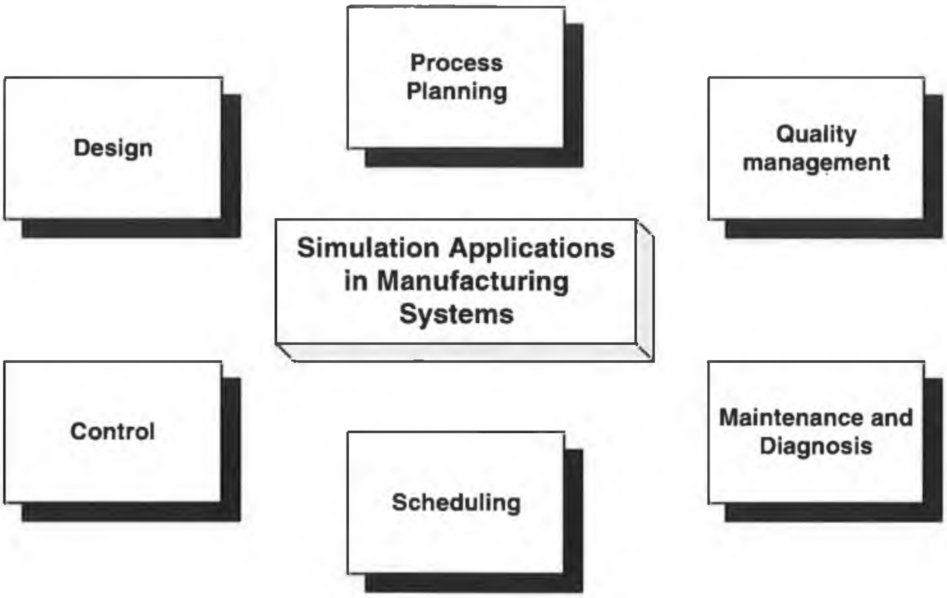


Figure C.2: Simulation Applications in Manufacturing Systems

Simulation models provide a picture that gives the appearance that it can think for itself. It has the capability of considering complex interrelated tasks and structurally projected outcomes by exercising the many alternative combinations in a reasonable time, while normally would take months to do in the real process. Simulation, by definition, allows for experimenting with a model of the system to better understand processes, with a goal of improving performance. Simulation modeling incorporates various inputs to a system and provides a means to evaluate, redesign, and measure or quantify customer satisfaction, resource utilization, process streamlining, and time spent.

Simulation of sophisticated automation systems in both the discrete parts and process environments has been relatively common since 1970's. Usually exercises were carried out to assist with the utilization of fixed resources and finite capacity under static conditions. Nevertheless, many Researchers made a significant contribution to the development of the simulation as a decision making tool [2]. New tools, new methods, and new software packages have been designed to build a powerful model that can deal with complex systems and to make a trade-off between the universality of the simulation and the complexity of the modeling.

A focus on simulation of FMS is shown in the coming section as the interest of this research in the use of simulation for FMSs.

C2.1 Simulation of FMS

Along with the rapid advancement in the power of the computer hardware and software, the possibility to simulate more sophisticated FMS with higher expectations and objectives [11]. Simulation modeling turns to be a powerful and flexible analysis approach in addressing many issues in FMS (Figure C.3). The role of simulation in FMS [1][4][8][12][23] started with the representation of real problem (real system) by a formal model to observe the dynamic behavior of the model and extended to assist in building knowledge-base of expert system to optimize the processes.

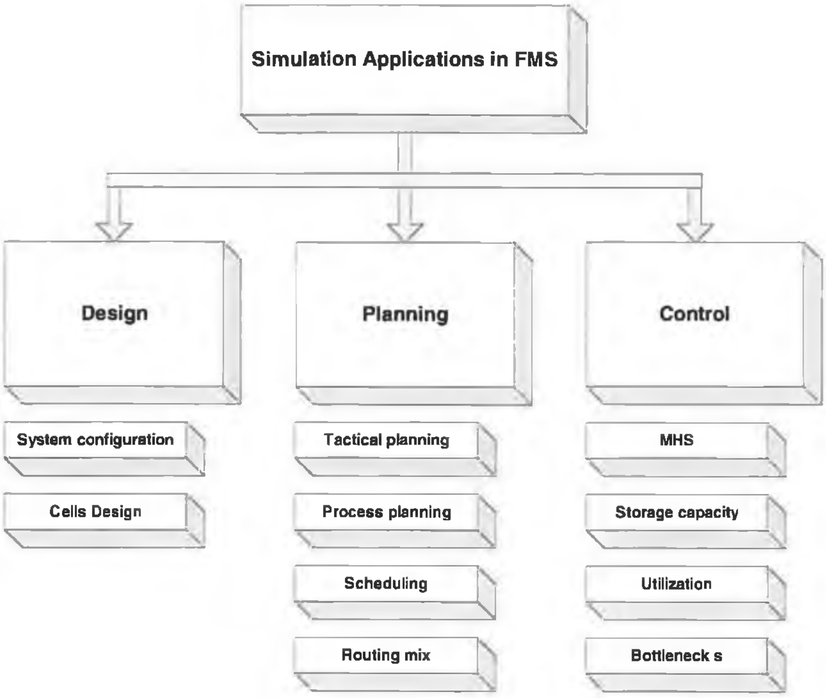


Figure C.3: Main applications of simulation in FMS

The main use of simulation in FMS issues comes into three main categories as follows:

1) FMS design problems
    a) Optimal system configuration (i.e. determination of number and types of machines, level of WIP in the system)
    b) Specification of the FMS layout and FMC
    c) Selection of a storage system (size of local buffers and/or central storage)
    d) Determination of other important system resources (i.e. number of pallets, number of fixtures, number and types of tools)

2) FMS planning problems
    a) Part type selection problem
    b) Machine grouping problem
    c) Loading problem
    d) Routing mix problem
    e) Other planning problems (i.e. tool storage capacity, scheduling, optimization of unit utilization)

3) FMS control and functions

   a) Specification of the type and capacity of the material handling system (MHS)
   b) Determination of optimal size of buffers
   c) Unit utilization
   d) Delays due to bottlenecks

Moreover, simulation of flexible manufacturing operations is one of the best tools for studying the impact of integration between different components upon overall system performance [13]. Each component in the production facility will have some individual performance capability, but when this is integrated to the manufacturing and imposed to different constraints due to production, the adjustments to this performance must be known. Many simulation software packages provide efficient means to accomplish replications of operation and examine different scenarios.

C2.2 Simulation of FMC

Simulation approach has been used in FMC. The simulation applications to an FMC can be classified into three main groups as shown in Figure C.4.

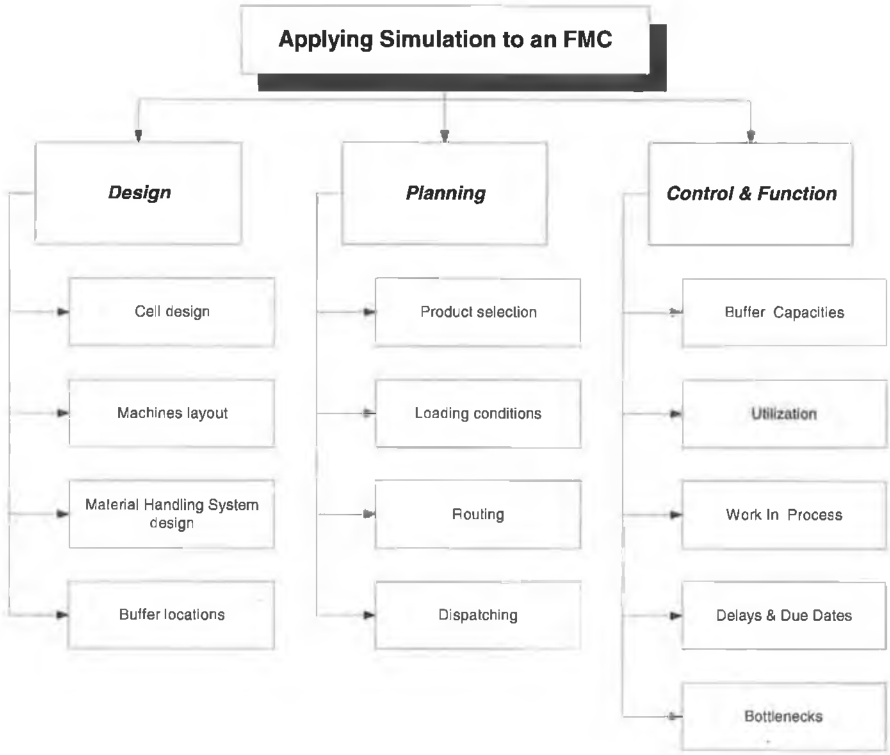


Figure C.4: Main applications of simulation in FMC

Simulation provides a simple yet flexible approach to generate a superb schedule for the flexible manufacturing cell scale. It has been influential in developing models to characterize the real systems and hence apply the required sensitivity analyses and optimization methods to the critical parameters (see chapter five).

C2.3 Advantages and Pitfalls of Simulation

The number of projects using simulation is steadily increasing. Many firms are realizing the benefits of utilizing simulation for more than just the one-time modeling of the facility. Rather, due to advances in software, managers are incorporating simulation in their daily operations on an increasingly regular basis. For most companies, the benefits of using simulation go beyond just providing a look into the future. These benefits are mentioned by many authors (e.g. Banks *et al.* [2] and [3], Pritsker [5], Law *et al.* [4][7] and [7], Carrie [8], Harrington *et al.* [16], and Arisha *et al.* [17] and [27]). In this research, a comprehensive classification of benefits and pitfalls in simulation projects have been presented in Table C.2.

C3. Simulation Software

Simulation software has been classified into different number of groups based upon many research studies (e.g. [1] [2] [5] [7][10], and [12]). In this research, the simulation software is classified in three main groups as shown in Figure. The fist group includes simulation language, and the second is general-purpose computer language whilst the third group refers to simulation packages which embrace different types of simulation software such as data-driven generic simulators and program generators.

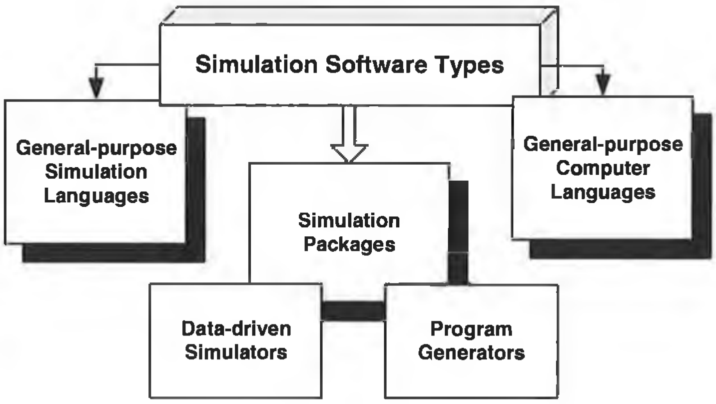


Figure C.4: Simulation Software types

Table C.2: Advantages and pitfalls of simulation

Factor	*Advantages*	*Pitfalls*
Financial	Simulation software package price is less than 1 % of total amount being expended for the implementation of a design and redesign or any real experimentation.	Some full simulation software packages are expensive for small enterprises to spend.
	Simulation proved to be wise investment as the benefits of having model can save many times than the cost of the package.	Simulation speed is factor of upgrading hardware and updating the package. This is attributable to the advances in hardware.
	Many simulation firms offer special deals for many users or network installation which makes the price of the user is low.	
Time	Time savings in simulation have many directions. For example, an entire shift in a matter of minutes or less can be examined.	Implementation time and project horizon depend on the data availability and staffs support.
	Time flexibility is an essential tool in simulation.	Training (if needed) is time-consuming in many cases.
	Simulation saves the time of experimenting the scenarios on the real facilities	
Training	Simulation software packages are getting easier to use and having friendly interface more and more.	Special training for the staff is needed in some projects.
	Some packages just need some simulation basics to be able to make a model	Assigning team for simulation training is inappropriate in many big firms.
	Training the simulation team can provide less disruptive and less mistakes in building then using the model	The preliminary training offered by the simulation software companies is not enough in some cases.
Model-ing	Modeling steps are well defined and easy to follow.	Modeling is an art that is learned over time and through experience.
	Identify the constraints and diagnoses of the problems provide better understanding of the system.	In many cases, the complexity of the manufacturing system gives modeler a hard time to do his job.
Alternatives	Simulation gives the possibility to test every aspect of a proposed change or addition without committing resources.	In some cases simulation provides hassles when the output interpretation is on a good level.
	Explore possibilities is one of the greatest advantages of simulation that one can have once a valid model has been developed.	Inappropriate use of the model might lead to non sense results.
	Bottlenecks detection as well as overview on the facility which provide the management to think better in the options to change.	
Simulation Tools	Animation feature is powerful tool of presenting the model and gives better understanding of the processes. Some packages have virtual reality modules which enrich the presentation capability.	Not all the packages have the same capabilities in presenting the models. Virtual Reality module does not have many features when it comes with the simulation package.
	Most of the packages have link to Microsoft Excel and Access to provide more practicality to the use simulation mode.	Evaluating and Selecting the appropriate software package is a crucial problem. (see [27])
	Statistical analysis techniques are helpful to the users and save lots of time.	
Risk	Simulation reduces risk of implementation on real facility as it provides a safe economic tool to provide answers.	Successful simulation projects percent are still not as high it gives confident to many administrations.
	Using simulation, build consensus to present design changes and approve modifications.	If the model is not well established the results can not be trusted and they can misconduct in some cases.

C3.1 General-purpose Simulation Language

A general-purpose simulation language is a simulation package that is used for modeling different types of systems with different characteristics. When a model is developed using a simulation language, the simulation analyst has to write a program using the modeling constructs of the language. This approach provides flexibility, but it is costly and time consuming. Some of the most popular simulation languages are SIMAN, SLAMII, SIMSCRIPT II.5, GPSS/H, SIMULA, PCModel, AutoMod II and ECSL. Some of these languages may have special features for manufacturing such as workstation and material handling module [19]. For example, AutoMod II is specifically directed towards material handling and manufacturing problems. Another example of such a package, SIMAN/CINEMA IV, has special material handling features, such as forklifts and conveyors.

C3.2 General-purpose Computer Language

Using the general-purpose language such as FORTRAN, C, or C++ involves writing routines for the basic facilities that would be included in any simulation package. This is time consuming and most unlikely to be cost-effective unless no available package can fulfill the requirements of the user.

C3.3 Simulation Packages

There are many different methods of classifying simulation packages into different types or groups. However, the main two groups are data-driven simulators and program generators.

A) Data-driven Simulators

Data-driven simulator has been defined as a computer package that allows the modeler to model systems with little or no programming [19]. Many data-driven simulators are domain-specific. They are used to model systems with specific features (e.g. cellular manufacturing systems) [20]. There are simulators currently available for certain types of manufacturing, computer, and communication systems. Examples of simulators that are dedicated to manufacturing simulation are SIMFACTORY II.5, WITNESS, ProModel for

Windows, and Xcell+. A graphical user interface is a fundamental part of simulators, which is used for modeling as well as for running the model. Most of these simulators employ a network in their underlying concept. Thus, entities are assumed to flow through a network from node to node. At these nodes they may be delayed as they engage in activities with entities and resources placed on the nodes. The resources placed on the nodes may also be engaged in endogenous activities. For example, in a manufacturing application such entities can be machines, which occasionally fail.

B) Program Generators

Program generators are used as another way of making simulation more accessible to non-computer specialists. A program generator is a computer program which itself generates another program. Unlike a compiler, which takes a source program written in a problem-oriented language and produces machine code, a program generator takes a system description and produces source code. This generated source code may then be compiled or interpreted to present a computable simulation model. Examples of program generators are CAPS/ECSL, VS7, and DRAFT. Program generators are usually interactive and accept a description of a conceptual model such as an activity cycle diagram (ACD). Most program generators require definition of the model entities, activities, queues, attributes, and priorities. Thus, the user starts modeling by drawing an 'ACD,' and then describes the components of the diagram to the program generator. Generally, features of program generators lie between those of simulation languages and data-driven simulators. A survey by Hlupic and Paul [21] showed that 10% of the users at universities and industry use only simulation language and this percent decrease with the advent of simulators. Table C.3 shows that the majority use both simulators and simulation languages or just simulators.

Table C.3:Percentage of different types of simulation software users at universities and industry (based on Ref. [26])

Type of software	*Universities*	*Industry*
Simulators	52 %	73 %
Simulation languages	4 %	9 %
Both	45 %	18 %

Another interesting survey by Christy and Watson [22] showed how new programming languages are selected for simulation applications (Table C.4). The fact that most of simulation software either simulator or language followed similar trends.

Table C.4: How new simulation software is selected for organization

Basis of selection	*Percent %*
Championed by someone in the organization	62
Available on time-sharing network	34
Checklist of features	29
Benchmark tests	24
Outside Recommendations	23
Other	12

C4. Simulation Model Building

Many research efforts directed to the simulation model building (e.g. [1], [3], [16], and [23] - [25]). The simulation model building has been discussed in this research into two phases; the model building approach and the simulation modeling process. For model building approaches, there are five basic options to establish the required simulation model. Table C.5 shows the different techniques including the advantages and disadvantages of each technique.

However, manufacturing applications include so many aspects and details and hence simulation modeling process can be arduous. Banks [2] showed a set of steps to guide a model builder in a thorough simulation modeling process. Similar figures and interpretation can be found in other sources such as Shannon *et al.* [1], Law *et al.* [4], and Centeno *et al.* [25]. Figure C.5 shows the main steps to implement for simulation modeling process. It is worth saying that the modeling process has been used in chapter five and six while building the simulation model relevant to the research.

Table C.5:Different techniques of building simulation model

Technique	***Examples***	***Advantages***	***Disadvantages***
Using a simulation Package	Use software packages such as: WITNESS, AUTOMOD, PROCESSMODEL, TaylorED, and EXTEND	- Powerful software tailored to the purpose of simulation - Permits user to concentrate on logic of system to be modeled, not on computer programming as such. - Makes available many man-years of experience and programming effort. - Permits models to be developed more quickly than in a general-purpose computer language. - Can be used by non-specialist in simulation.	- Purchase cost is relatively high. - Training needs. - May not be well designed for the special features of user's system. - Risk of invalid conclusions until substantial expertise is gained.
Using a simulation language	Use a software packages such as: GPSS/H, SIMULA, ECSL and PCModel	- The programmer can develop the model to achieve all the user requirements - Some packages provide special features that serve in manufacturing models - Most suitable for use by a simulation specialist.	- it needs programming , so it is time and money consuming - Hard to be used with non-specialist.
Writing the model in a general-purpose computer language	Use languages such as: FORTRAN, C , C++	- It fits more with the system requirements (tailored program). - Avoids the cost of purchasing a package. - Programmer knows all the details of his own program.	- Cost of good programmer is high. - Time needed to develop the program. - The entire program has to be debugged.
Using a generic model	Use generic models for particular types of systems, such as: FMS, AGV,....	- Avoids any programming. - Minimize the time to get results. - Usually acceptable cost. - Usually runs on any "standard" hardware. - Suitable for use by non-specialists.	- May not be capable of modeling specific features. - May include unacceptable simplifications or assumptions.
Using a consultant	Use simulation expertise or consultancy company	- Speed of obtaining results. - Professional expertise.	- The cost is normally high. - Difficulty in modifying model if plans change. - Expertise may not be transferred to company's own staff.

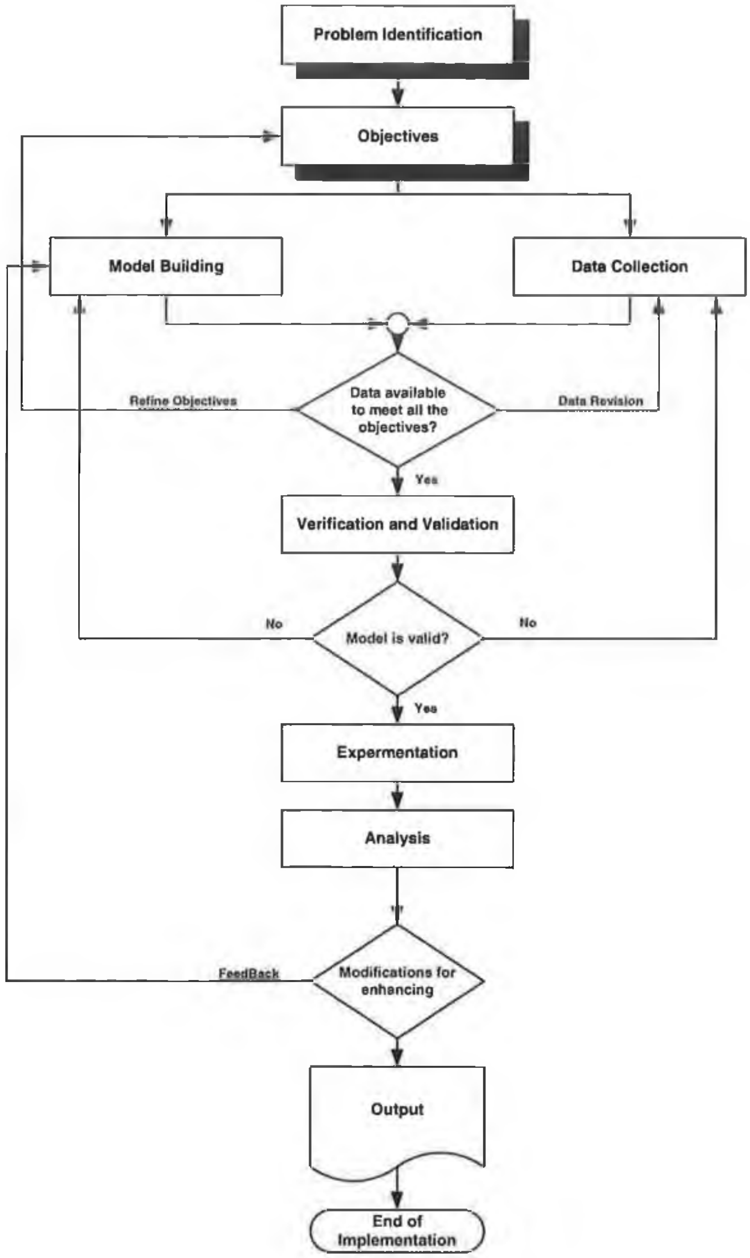


Figure C.5:The simulation modeling process steps

C5. Simulation Software Selection

The number and variety of simulation software packages on the market increased tremendously along with the increase of using simulation. Consequently, the varieties of these packages led to some bewilderment on the part of potential users when faced the selection process. The use of simulation as a tool to help these complex, dynamic and stochastic manufacturing systems involve large capital investments, as it is cheaper and easier to experiment with simulation models, rather than experimenting with the real systems. There is a variety of potential benefits of simulation in manufacturing environments including: greater understanding of systems, reduced operating costs, risk reduction, lead time reduction, reduction of capital costs, and faster configuration changes. As a result, managers and administrators have begun to look to simulation for an aid to day-to-day operational problems as well as tactical and strategic issues. The growing use of simulation for the analysis of manufacturing systems has resulted in a rise in the number of both general purpose and application oriented simulation software packages. Some researchers have contributed their own classifications of evaluation criteria. Table C.6 extends the literature proposed by Nikoukaran *et al.* [26] in order to cover major studies in this subject. More details about the classification and selection criteria can be found in Arisha *et al.* [27]. The simulation software selection process is considered as one of the most critical tasks in simulation projects. There are many considerations that should be taken into account while selecting the simulation package. The classification of the criteria into groups and sub-groups is an effective way to organize the list different features that should be considered in the evaluation process. The criteria can be classified twofold: technical criteria and business criteria as shown in Figure C.6. These two groups represent the highest levels of the proposed framework. The business criteria concern the vendor, the user, and their contract features, while the technical elements consider most of the features of the simulation software. An explanation of each criterion and sub-criterion is presented to describe the feature and its importance in evaluating simulation software.

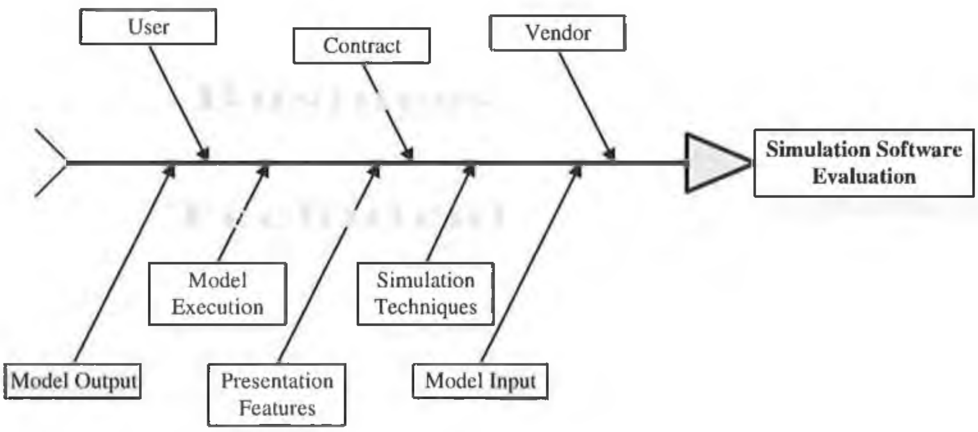


Figure C.6: Simulation software evaluation criteria (cause and effect diagram)[27]

Table C.6: Major literature in simulation software selection criteria

Holder **1990** **[31]**	**Banks** **1991** **[30]**	**Law** ***et. al*** **1991** **[7]**	**Banks** ***et. al*** **1991** **[14]**	**Mackulak** ***et. al*** **1994** **[32]**	**Davis** ***et. al*** **1994** **[33]**	**Banks** ***et. al*** **1996** **[34]**	**Kuljis** **1996** **[35]**	**Hlupic** **1997** **[36]**	**Nikoukaran** ***et. al*** **1999** **[37]**	**Arisha** **2002** **[27]**
1.Technical 2.System Development needs 3.End user needs 4.Future Development 5.Functionality 6.Commercial	1.Input 2.Processing 3. Output 4.Environ-ment 5.Cost	1.General features 2.Animation 3.Statistical capabilities 4.Customer support 5.Output results	1.Robust features 2.Qualitative considerations 3.Cost 4. Basic features 5. Special constructs	1.General 2.Data acquisition 3.Model development 4.Validation and Verification 5.Model execution 6.Documentation 7. Simulation project data 8.Methods of user interface	1.Cost 2.Comprehen siveness of the system 3. Integration with other systems 4.Documenta tion 5.Training 6.Ease of use 7.Hardware and installation 8.Confidence related issues	1.Input features 2.Processing features 3.Output features	1.Main system 2.Data input/Model specification 3. Simulation experiment 4.Simulation results 5.Printed manuals 6.On-line user assistance	1.General features 2.Visual aspects 3.Coding aspects 4.Efficiency 5.Testability 6.Statistical facilities 7.Input/Output 8.Modelling assistance 9.Software compatibility 10.Experimenta l facilities 11.Financial and technical features	1.Vendor 2.Model and input 3.Execution 4.Animation 5.Testing and efficiency 6.Output 7.User	1.Technical -Model input - Simulation techniques - Presentation features - Model execution - Model Output 2.Business -Vendor -Contract - User

C6. Simulation Software Evaluation and Selection Methodology

A suggested methodology of simulation software selection is shown in Figure C.7. As an analysis of the background research material and research experience, the methodology established. First step represents the beginning of the process of selection and considers being critical. The purpose of simulation software and the need for the package direct the selection process to more specific simulation software type that serves the application. For example, if the software is intended to use in education purpose, it should be determined whether teaching will be performed for undergraduate or postgraduate level [9]. As the level of education increases, the comprehensiveness and complexity of software might increase, beside the model development complexity. After the determination of the purpose of the simulation software, a preliminary list of simulation software packages can be used. The next step is to fill the assessment checklist (the user, expert, or simulation team). An empirical weighing approach (based on customer/application preferences) has to be employed to complete the selection methodology. Step 3 contains the main selection criteria that should be assigned to evaluate the software packages in the list. The main elements of evaluation performed in step 4 without going further in the details. Step 5 is to shorten the simulation software list based upon the evaluation performed in step 4. The assessment of the packages in Step 4, 5 achieved through a checklist that is shown in Figure C.8. Two actions should be established in parallel, first is to let the user with the aid of his simulation team or simulation expert and based on the results, weight each criteria in terms of its relative importance will be set. Beside, the simulation packages evaluation should carry on the same time as a contact with the short-list simulation software companies for contract details and prices discounts if possible. On the other hand, detailed evaluation criteria should be applied to the selected list. Following this stage, a final selection for one package to be purchased should be made based upon the scores collected from the checklist. There are many elements could be added to this stage but it is beyond the scope of this study. After purchasing, staff training is an important step towards ensuring efficient performance and effective implementation.

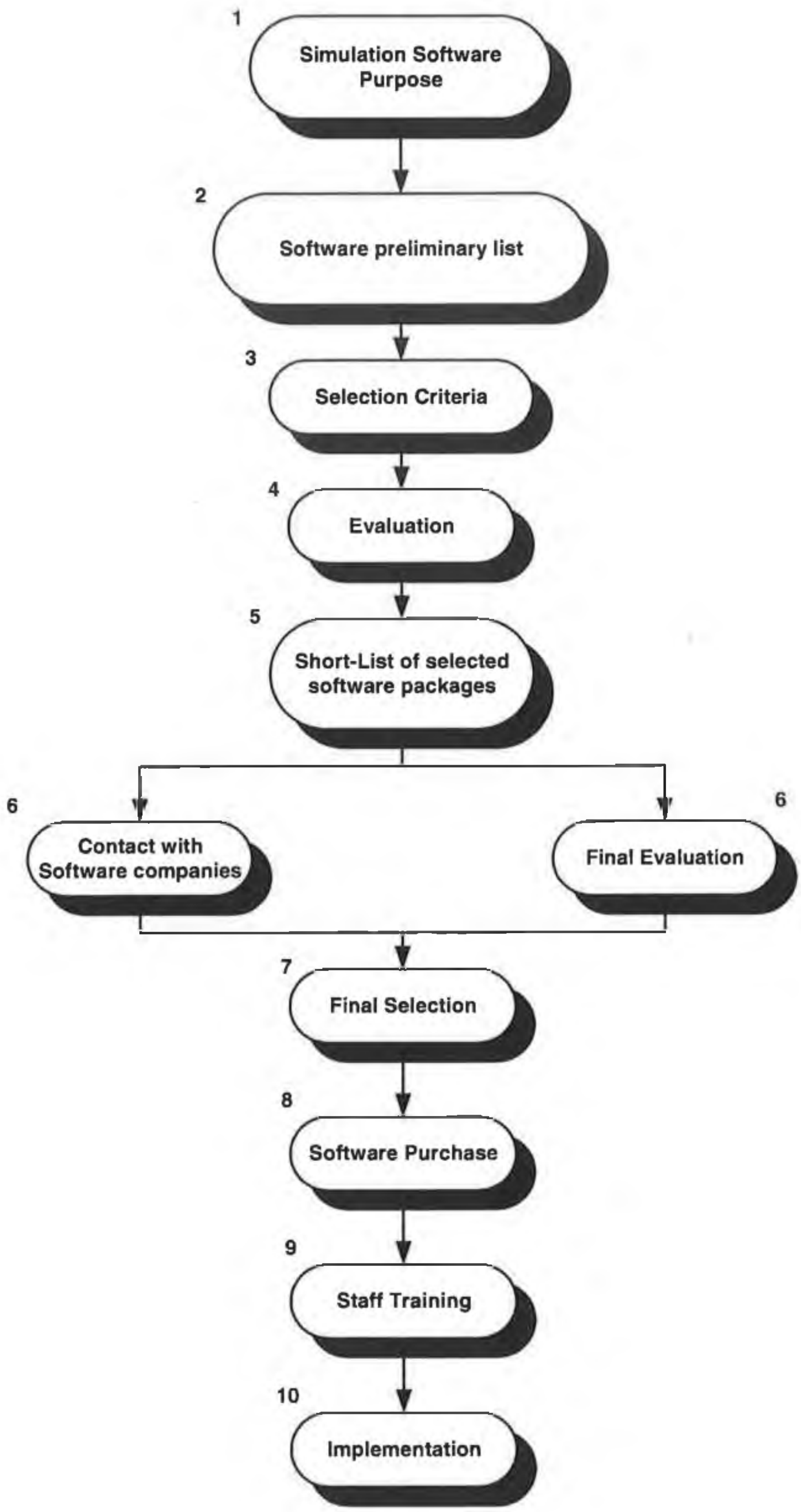


Figure C.7:Steps of suggested simulation software selection

Criterion	group	Sub-group	Feature	Indication Levels				
				A	B	C	D	E
Business	Vendor	Pedigree	Vendor history (Reputation)					
			Vendor experience					
			Contact facility					
			Company type (local, international...etc.)					
		Service	Service after sale					
			Trouble shooting					
			Documentation availability					
			Pre-Purchase services (CD demo, evaluation copy, ...etc.)					
		Support	Training on the software					
			Technical support					
			Consultancy session					
	Contract	Financial	Package price					
			Discount availability					
			Number of modules					
			Type of license (network, individual...etc)					
			Payment allowance					
			Updating cost					
		Technical	Maintenance					
			Group meeting					
			Security / Authority					
Technical	Model Input	Building Tools	Data Collection options					
			Model merging possibility					
			Library of usable modules					
			Model options (formal logic, Hierarchical modeling ... etc.)					
		Features	Functions (built-in, user defined...etc.)					
			Dialogue boxes available					
			Pick and click capability					
			Error detection					
			Language interface					
		Distributions	Statistical Distribution					
			Standard fitting					
			User Defined Distribution					
		Limits	Model size (no. of elements, entities, icons...etc.)					
			Number of tutorial examples					
	Simulation Techniques	Model Coding	Accessibility to source code					
			Programming tools					
			Attributes , Global Arrays , ...etc					
		Modeling Approaches	Variety of modeling approaches (event based, process interactions... etc.)					
		Shop Floor Control	Conditional routing option					
			Dispatching rules					
		Generators	Program schedules generator					
			Random numbers generators					
			Conceptual modeling generator					
	Presentati on Features	Animation	Icons (library, interface CAD, Bitmap, 3D, colors...etc.)					
			Model animation (concurrent, post-simulation.. etc.)					
		Display	Display (paths, values dynamically, state ...etc.)					
		Virtual Reality	Virtual Reality features available					
	Model Execution	Speed Control	Model speed control while runs					
		Run	Run options (Automatic batch run, multiple runs, step function...etc)					
		Warm-up	Warm-up period determination options					
		Clock	Time control options (backward clock... etc)					
		Options	More options in execution (breakpoints, multitasking .. etc.)					
	Model Output	Report	Standard set of reports					
			Customized reports					
		Tracing	Snapshots option					
		Integration	Integration with other packages (Excel, Access ... etc.)					
		Status	Static or dynamic results option					
		Statistical	Statistical analysis options (mean, variance, ...etc.)					
		Output form	Output form (hardcopy, file, software interface...etc.)					
		Graphics	Output presentation options (Pie chart, bar chart... etc.)					
		Validation &	Options (interactive debugger, error messages... etc.)					
		Optimization	Optimization module					

Figure C.8: Assessment Checklist

References

[1] Shannon, R.E, "Systems Simulation-The Art and Science", Prentice-Hall, 1995.

[2] Banks, J., "Introduction to Simulation", Proceedings of the Winter Simulation Conference J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds., 2000.

[3] Banks, J., Carson, J., Nelson, B. L., Nicol, D. M. "Discrete-Event System Simulation", Prentice-Hall, Inc. , NJ, 2001.

[4] Law, A. M., Kelton, W. D. "Simulation Modeling and Analysis", 3rd Ed., McGraw-Hill, New York, 2000.

[5] Pritsker, A.A.B., "Introduction to Simulation and SLAM II", Halsted Press and Systems Publishing Corporation, 1984.

[6] Kochhar, A.K., "Computer simulation of manufacturing system – 3 decades of progress", Proceedings of the third European Simulation Congress, Edinburgh, UK, pp 3-9, 1989.

[7] Law, A. M., Kelton, W. D. "Simulation Modeling and Analysis", 3rd Ed., McGraw-Hill, New York, 2000.

[8] Carrie, Allan, " Simulation of Manufacturing Systems". *John WILEY & Sons, 1998.* ISBN: 0471 915 74 2.

[9] Hlupic, V. and Paul, Ray J. "Methodological approach to manufacturing simulation software selection", Computer Integrated Manufacturing Systems, Vol. 9, No.1, pp 49-55, 1996.

[10] Eldabi, T., and Paul, R., "Flexible Modeling of Manufacturing Systems with Variable Levels of detail", Proceedings of the Winter Simulation Conference WSC, S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, pp 801-808, 1997.

[11] Greenwood, Nigel R. "Implementing Flexible Manufacturing System", Macmillan education Ltd., 1988. ISBN: 0 333 39156x

[12] Mills, R.I., "Simulation for manufacturing systems- a critical review", Proc. FMS-5, 5th Int. Conf. on Flexible Manufacturing Systems, IFS (Publications) Ltd, Bedford, England, 1986.

[13] Lenz, John E. "Flexible Manufacturing: Benefits for the low-inventory factory", Marcel Dekker Inc., 1989. ISBN: 0 8247 7683 6.

[14] Banks, J., and Gibson, R. R., "Don't Simulate when: 10 rules for determining when simulation is not appropriate", IEE Solutions, Sept.97.

[15] Law, A.M. and McComas, M.G., "Simulation of Manufacturing Systems", Proceedings of the 1999 Winter Simulation Conference (WSC), pp56-59, 1999.

[16] Harrington, J.H. and Tumay, K., "Simulation Modeling Methods", McGraw-Hill, 2000.

[17] Arisha, A., and El Baradie, M., "Simulation Based Rescheduling of Manufacturing Resources: A Real-World Application", ESDA 2002, 6th Biennial Conference on Engineering Systems Design and Analysis, Istanbul, Turkey, July 8-11, 2002, Ref. No. ESDA2002/MAN-063.

[18] Imagine That, Inc., 6830 Via Del Oro, Suite 230, San Jose, CA 95119 USA, 2002. http://www.imaginethatinc.com/frame_simulation.html.

[19] Law, A. M., Haider, S. W. "Selecting simulation software for manufacturing applications: practical guidelines & software survey", Industrial Engineering, Vol. 34, pp 33 – 46, 1989.

[20] Pidd, M., "Computer Simulation in Management Science", 3rd edition, Wiley, Chichester, 1992.

[21] Hlupic, V. and Ray J. P., "Simulation Software in Manufacturing Environment: A Users' Survey", Journal of Computing and Information Technology-CIT 1, Vol. 3, pp 205- 212, 1993.

[22] Christy, D. P., Watson, H. J., "The application of simulation: a survey of industry practice", Interfaces, Vol. 13, No. 5, pp 47-52, 1983.

[23] Pritsker, A. A. B., "Principles of Simulation Modeling Methods", McGraw-Hill, 2000.

[24] Morris, W.T., "On the Art of Modeling", Management Science, Vol. 13, No. 12, 1976.

[25] Centeno, M. A., Carrillo, M., "Challenges of Introducing Simulation As a Decision Making Tool", Proceedings of the Winter Simulation Conference, B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds., 2001.

[26] Nioukaran, J., Pual Ray J., "Software Selection for Simulation in Manufacturing: a review", Simulation Practice and Theory, 7, 1999, pp 1-14.

[27] Arisha, A., El Baradie, M. "On the Selection of Simulation Software for Manufacturing Application", IMC-19, Proceedings of the Nineteenth International Manufacturing Conference, Queen's University Belfast, August 28- 30, pp 495 – 507, 2002.

[28] Blazewicz, J., Ecker, Klaus H., Schmidt, Guenter and Weglarz, Jan " Scheduling in Computer and Manufacturing Systems". *Springer Verlag*, Berlin 1994. ISBN: 3540580492.

[29] Dzielinski, B. P. and Baker, C. T., "Simulation of Simplified Job Shop", Management Science, Vol. 6, No. 3, Apr. 1960.

[30] Banks, J., Aviles, E., McLaughlin, J.R., Yuan, R.C., "The simulator: new member of the simulation family", Interfaces, Vol.21, No. 2, pp 76 – 86, 1991.

[31] Holder, K., "Selection simulation software", OR Insight, Vol. 3, No.4, pp 19 – 24,1990.

[32] Mackulak, G. T., Savory, P. A., and Cochran, J. K., "Ascertaining important features for industrial simulation environments", Simulation, Vol. 63, pp 211 – 221, 1994.

[33] Davis, L., and Williams, G., "Evaluation and selecting simulation software using the analytic hierarchy process", Integrated Manufacturing Systems, Vol. 5, No. 1, pp 23 – 32, 1994.

[34] Banks, J., "Interpreting simulation software checklists", ORMS Today, Vol. 22, No. 3, pp 74 – 78, 1996.

[35] Kuljis, J., " HCI and simulation packages", Proceedings of The 1996 Winter Simulation Conference, California, USA, pp. 687-694, 1996.

[36] Hlupic, V., "Simulation software selection using SimSelect", Simulation, Vol. 69, No. 4, pp 231 – 239, 1997.

[37] Nikoukaran, J., Hlupic V., and Paul R. J., "A hierarchical Framework for Evaluation Simulation Software", Simulation Practice and Theory, 7, pp 219 – 231, 1999.

Appendix D

General Flow Shop Scheduling Model (Phase 1)

```
/* The optimum Make-span Program Codes */


/* For Data Input ,Rows are the jobs and the column are the machines */

#include <conio.h>
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <time.h>
#define RANGE 11
#define MAX 32767

typedef struct NODE{
      long num;
      long val;
      struct NODE *next;
}
NODE;
NODE *make_NODE(void);
NODE *link_data(long);
void del_NODE(NODE*);
void insert_NODE(NODE*,NODE*);
void remove_NODE(NODE*);
void dump_list(NODE*);
void perm2(int*,int);
void store(void);
void ran();
void readdata(void);
NODE *head;

#define OFF 0
#define ON 1

void process(int n,int m);
int T[250][250],f[250][250],temp[250][250];
int i,j,n,m,number,x,index[250],seq[250],maximum=1000000000;
long count =0;
int num;
int amout,column;
long now;
long int yy,runs,y,ave,average,result;
int output=ON;
char c;
FILE *fp,*fp2;

/* to define function max. */
int max(int x, int y)
{
        if (x>y) return x; else return y;
}
```

```
void insert_NODE(NODE *at,NODE *ins)
{
      ins->next=at->next;
      at->next=ins;
}

void remove_NODE(NODE *node)
{
      /* Remove the next node down */
NODE *old;
old=node->next;
node->next=node->next->next;
free(old);
}

NODE *make_NODE()
{
      NODE * ret;
      if ((ret=(NODE *)malloc(sizeof(NODE)))==NULL)
      {
            printf("out of memory program halted\n");
            exit(0);
      }
            return(ret);

}
void del_NODE(NODE *node)
{
      free(node);
}

NODE* link_data(long data)
{
      static int flag=0;
      static NODE * head;
      NODE * new_NODE;
      NODE * current;
      NODE * old;

      if(flag==0) {
            head=make_NODE();
            head->val=0;
            head->num=0;
            head->next=NULL;
            flag=1;
      }

old=head;
current=head->next;
while(current){
      if (data<current->val)
      {
            new_NODE=make_NODE();
            new_NODE->val=data;
            new_NODE->num=1;
            insert_NODE(old,new_NODE);
            return(head);
      }
if (data==current->val)
{
```

```
        current ->num++;
        return(head);
}
if (data>current -> val)
{
        old=current;
        current=current->next;
}
}
new_NODE=make_NODE();
new_NODE->val=data;
new_NODE->num=1;
insert_NODE(old,new_NODE);

return (head);
}

void dump_list(NODE* head)
{
        NODE* current;
        FILE *out;

        out = fopen("out.dat","w+");
        current=head->next;
        while(current)
        {
                if ( output==ON)
                {
                        printf("For the Make-span =%ld : The Frequency =
%ld\n",current->val,current->num);
                        fprintf(fp2,"For the Make-Span =%ld: the Frequency =%ld
\n",current->val,current->num);
                }

                fprintf(out,"%ld\t\t%ld \n",current->val,current->num);
                current=current->next;
        }

        fclose(out);
        printf("\n");
        for(i=0;i<n;i++)
        {
                fprintf(fp2,"%d",seq[i]);
        }

        fprintf(fp2,"= Optimum Job Sequence , Optimal Make-Span=
%d",maximum);
        printf("Problem matrix size:");printf("(%d X %d) \n",n,m);
}


void main (int argc,char * argv[])
{
        {
                unsigned seed;
                if (argc > 1) if (*argv[1]=='n') output=OFF;
                fp2=fopen("out.dat","w+");
                printf("    \nEnter number of jobs ..... " ); /* = %d\n",n);*/
                scanf ("%d",&n);
                printf("Number of Jobs     = %d\n",n);
```

```
            printf("\nEnter number of Machines... " );
            scanf("%d",&m);
            printf("Number of machines =%d\n",m);
            printf("\nEnter number of runs ... ");
            scanf("%d",&runs);
            printf("Number of runs = %d\n",yy);
            column=n;
            printf("\nEnter random seed ......");
            scanf("%d",&seed);
            printf("Random seed      = %d\n",seed);
            printf("\n");
            srand(seed);
            {
                  clock_t start, end;

                  start= clock();

                  /* Read in data in matrix (n X m) problem */

                  for (yy=0;yy<runs;yy++)
                  {
                        maximum=1000000000;
                  /* in case of stop generating random processing times,
block the next line */
                        /* ran(); */
                        readdata();
                  /*    fprintf(fp2,"      \n Optimal job sequence problem
:");

                        fprintf(fp2,"   \n\n"); */
                        fprintf(fp2,"   %d Jobs,%d Machines, Random
seed=%d\n",n,m,seed);
                        fprintf(fp2,"    Number of runs            =
%d\n",yy);

                        for (i=0;i<n;i++)
                              index[i]=i+1;
                        perm2(index,n);
                        printf("Number of job Sequence    = %ld\n",count);
                        fprintf(fp2,"  Number of Job Sequences =
%ld\n",count);

                        fprintf(fp2,"\n");


                        for (i=0;i<n;i++)
                        {
                              printf("%d",seq[i]);
                              fprintf(fp2,"%d",seq[i]);
                        }

                        printf(" = Optimum Job Sequence, Optimal Make-span
= %d\n",maximum);
                        fprintf(fp2,"  = Optimum Job Sequence , Optimal
Make-span = %d\n", maximum);
                        result=maximum;
                        ave=ave+result;
                  /*  system ("cls");
                      fprintf(fp2,"\n");
                        fprintf(fp2,"%d Jobs , %d Machines, the Pure Flow-
Shop , (Number of runs)=%d\n",n,m,yy);
                        fprintf(fp2,"\n");
```

```
                        average =ave/runs;
                /*  printf("Average Optimal Make-span= %d \n",average);
                        fprintf(fp2,"Average Optimal make-span=
%d\n",average);
                */
                        end = clock();
                        printf("CPU timein seconds =%f \n ", (float)(end-
start)/CLOCKS_PER_SEC);
                        printf("    \n");
                /*  fprintf(fp2,"      Make-span's Frequencies : " );
                    printf("(%d X %d) \n\n",n,m);    */
                /*  fprintf(fp2,"\n");    */
                        fprintf(fp2,"CPU time in seconds =%f \n" ,
(float)(end-start)/CLOCKS_PER_SEC);
                        fprintf(fp2,"\n");
                        fprintf(fp2,"fp2,     Make-Span's Frequencies:");
                            fprintf(fp2,"\n\n");
                /*  fprintf(fp2,"Problem matrix size"); fprintf(fp2,"(%d
X %d)\n\n ",n,m); */

                        dump_list(head);

                }
            }
        }
}

void process (int n, int m )
{
        /* The make_Span Formula used */
        int i , j ;
        f[1][1]=T[1][1];
        for (i=2;i<m+1;i++)
        {
                f[1][i]=f[1][i-1]+T[1][i];
        }

        for (i=2;i<n+1;i++)
        {
                f[i][1]=f[i-1][1]+T[i][1];
        }

        for (j=2;j<m+1;j++)
        {
                for (i=2;i<n+1;i++)
                {
                        f[i][j]=max(f[i-1][j],f[i][j-1])+T[i][j];
                }
        }
        return;
}

void store(void)
{
        static int co=0;
        count++;

        for (i=1;i<=n;i++)
        {
                for (j=1;j<=m;j++)
```

```
			{
				T[i][j]=temp[index[i-1]][j];
			}
		}

		process(n,m);

		if(output) {
			for (i=0;i<n;i++)
			{
				printf("%d",index[i]);

				/* Try to stop the details in the output */
				fprintf(fp2,"%d",index[i]);


			}
			printf("Make-Span =%d\n",f[n][m]);
				/* Try */

			fprintf(fp2,"Make-Span=%d\n",f[n][m]);


			head=link_data((long)f[n][m]);
		}

		if (maximum> f[n][m])
		{
			maximum = f[n][m];
			for (i=0;i<n;i++)
			{seq[i]=index[i];
			}
		}
}

void perm2 (int *s,int n)
{
		int i ;
		int tmp;

		if (n==1)
			store();
		for (i=0;i<n;++i)
		{
			tmp=s[0];
			s[0]=s[i];
				s[i]=tmp;
			perm2(&s[1],n-1);
			tmp=s[0];
			s[0]=s[i];
			s[i]=tmp;
		}

}

void ran()
{
		FILE *fp;
		fp=fopen("in.dat","w");
		int amount=n*m;
```

```
        count=0;
        for(i=0;i<amount;i++)
        {
                y=rand()%11;
                fprintf(fp,"%-3i",y);
                count++;
                if (count==column)
                {
                        fprintf(fp,"\n");
                        count=0;
                }
        }
fclose(fp);

}

void readdata(void)
{
        FILE *fp;
        fp=fopen("in.dat","r");
        for (i=1;i<m+1;i++)
        {
                for(j=1;j<n+1;j++)
                {
                        fscanf(fp,"%d",&number);
                        T[j][i]=number;
                        temp[i][j]=number;
                }
        }

        fclose(fp);
}
```

Appendix E

General Flow Shop Scheduling Model (Phase 2)

```
                         Phase 2

            /* Effect of the priority Rules */

#include <conio.h>

#include <stdio.h>

#include <string.h>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#define RANGE 10

#define MAX 32767

#define SIZE 250



int intcmp(int a , int b);
void process (void);
void rule2(void);
void rule3(void);
void display (void);
void readdata(void);
void rule4(void);
void rule5(void);
void sort4(void);
void ran();
int compare(const void *arg1, const void *arg2);

int T[SIZE][SIZE],f[SIZE][SIZE],temp[SIZE][SIZE];
int i,j,k,m,n,l,number;
char x;
int
index[SIZE],store[SIZE],store2[SIZE],sum[SIZE],store3[SIZE],p[SIZE],z[SIZE
],su[SIZE];
char c;
int rulenum;
int amount,count,y,column;
long now;
```

```
double num;
int yy,runs;
long int idle,idle1,result6,result7,av1,av2,av3,av4,av5,av,result8;
long int a,a1,a2,a3,a4,a5;
long int average,average1,average2,average3,average4,average5,result;
long int
result1,result2,ave,ave1,ave2,ave3,ave4,ave5,result3,result4,result5;
FILE *fp,*fp2;
int max(int x, int y)
{
        if (x>y) return x; else return y;
}
void main()
{
      {
            unsigned seed;
            fp2=fopen("out.dat","w+");
            printf(" \n Enter Number of Jobs  .. ");
            scanf("%d",&n);
            printf("\n Enter number of machines .. ");
            scanf("%d",&m);
            printf("\n Enter number of runs .. ");
            scanf("%d",&runs);
            printf("\n Enter number of seed .. ");
            scanf("%d",&seed);
            srand(seed);
            column=n;
            {
                  clock_t start, end;
                  start= clock();

                  /* Read in data in matrix */
            /* printf(%d jobs, %d machines \n\n",n,m); */
            for ( yy=0;yy<runs;yy++)
            {
                  printf("MAKESPAN RUN %d \n",yy);

            /* In case that the data should be read from the file, Cancel
the next line */

            ran();

            rulenum=1;
            readdata();
            process();
            display();
            average1=average1+result;
            ave1=ave1+result2;
            av1=av1+result8;
            a1=a1+result5;
            rule2();
            display();
            average2=average2+result;
            ave2=ave2+result2;
            av2=av2+result8;
            a2=a2+result5;
            readdata();
            process();
            rule3();
            display();
```

```
            average3=average3+result;
            ave3=ave3+result2;
            av3=av3+result8;
            a3=a3+result5;
            readdata();
            process();
            rule4();
            display();
            average4=average4+result;
            ave4=ave4+result2;
            av4=av4+result8;
            a4=a4+result5;
            readdata();
            process();
            rule5();
            display();
            average5=average5+result;
            ave5=ave5+result2;
            av5=av5+result8;
            a5=a5+result5;
            readdata();
            process();
            }
            /* end of yy loop */
            system ("cls");
            printf("\n");
            fprintf(fp2,"\n");
            fprintf(fp2,"%d Jobs, %d Machines, Random seed = %d,(number of
runs)=%d \n\n", n,m,seed,yy);
            /* printf("%d Jobs,%d Machines \n\n",n,m);
            fprintf(fp2,"Average MAKE-SPAN \n\n");
            printf("Average MAKESPAN ( Number of runs) = %d\n\n",yy);
            printf("\n\n"); */
            average=average1/runs;
            /* printf("Rule FCFS =%d \n",average); */
            fprintf(fp2,"Rule 1 FCFS = %d\n\n",average);
            average=average2/runs;
            /* printf("Rule SPT (SI)=%d\n,average); */
            fprintf(fp2,"Rule 2 SPT(SI)=%d \n\n",average);
            average=average3/runs;
            /* printf("Rule LPT(LI)=%d \n", average ); */
            fprintf(fp2,"Rule 3 LPT(LI)=%d \n\n",average);
            average=average4/runs;
            /* printf("Rule SRPT = %d \n " , average); */
            fprintf(fp2,"Rule 4 SPRT = %d \n\n",average );
            average=average5/runs;
            /* printf("Rule LRPT = %d \n",average); */
            fprintf(fp2,"Rule 5 LPRT = %d \n\n",average );
            /* end= clock();
            fprintf(fp2,"\n");
            fprintf(fp2,"Computation time in seconds =%f\n ",(end-
start)/CLK_TCK); */
            fprintf(fp2,"\n");
            /*fprintf(fp2,"problem matrix size: (%d X %d)\n",n,m);
            printf("\n");
            fprintf(fp2,"\n"); */
            fprintf(fp2,"average mean completion time \n\n");
            ave=ave1/runs;
            fprintf(fp2,"Rule1 FCFS= %d \n\n",ave);
            ave=ave2/runs;
```

```
            fprintf(fp2,"Rule 2 SPT(SI)=%d\n\n",ave);
            ave=ave3/runs;
            fprintf(fp2,"Rule 3 LPT(LI)= %d \n\n",ave);
            ave=ave4/runs;
            fprintf(fp2,"Rule 4 SRPT = %d\n\n",ave);
            ave=ave5/runs;
            fprintf(fp2,"Rule 5 LRPT = %d \n\n",ave);
            fprintf(fp2,"Average Total Waiting Time \n\n");
            av=av1/runs;
            fprintf(fp2,"Rule 1 FCFS = %d \n \n ",av);
            av=av2/runs;
            fprintf(fp2,"Rule 2 SPT  = %d \n\n",av);
            av=av3/runs;
            fprintf(fp2,"Rule 3 LPT = %d\n\n",av);
            av=av4/runs;
            fprintf(fp2,"Rule 4 SRPT = %d \n\n",av);
            av=av5/runs;
            fprintf(fp2,"Rule 5 LRPT = %d\n\n",av);
            fprintf(fp2,"\n");
            fprintf(fp2,"Average Total Idle Time \n\n");
            fprintf(fp2,"\n\n");
            a=a1/runs;
            fprintf(fp2,"Rule 1 FCFS  = %d \n\n",a);
            a=a2/runs;
            fprintf(fp2,"Rule 2 SPT  =%d \n\n",a);
            a=a3/runs;
            fprintf(fp2,"Rule 3 LPT  = %d \n\n",a);
            a=a4/runs;
            fprintf(fp2,"Rule 4 SRPT  = %d \n\n",a);
            a=a5/runs;
            fprintf(fp2,"Rule 5 LRPT  = %d \n\n",a);
            end=clock();
            fprintf(fp2,"Computational time in seconds =%f\n",
(float)(end-start)/CLOCKS_PER_SEC);
            }}
            fclose(fp2);
            }
            void ran ( void)
            {
                  fp=fopen("in.dat","w");
                  amount=n*m;
                  count=0;
                  for (i=0;i<amount;i++)
                  {
                        y=rand()%11;
                        count++;
                        fprintf(fp,"%-3i",y);
                        if (count==column)
                        {
                              fprintf(fp,"\n");
                              count=0;
                        }
                  }

            fclose(fp);
            }

            void readdata( void)
            {
                  fp=fopen("in.dat","r");
```

```
        for ( i=0;i<m;i++)
        {
              for(j=0;j<n;j++)
              {
                    fscanf(fp,"%d",&number);
                    T[j][i]=number;
                    temp[j][i]=number;
              }
        }
  fclose(fp);
  }

  void rule4(void)
  {
        for(i=0;i<m;i++)
        {
              sum[i]=0;
              index[i]=0;
        }

        rulenum = 4;
        for (j=0;j<n;j++)
        {
              for(i=0;i<m;i++)
              {
                    sum[j]=sum[j]+temp[j][i];
                    index[j]=sum[j];
                    result3=sum[j];
              }
        }
        sort4();
  }

  void sort4(void)
  {
//      qsort(index,n,sizeof(index[1]),compare);
        qsort( (void *)index, (size_t)n, sizeof(int), compare );
        for (i=0;i<n;i++)
        {
              j=0;
              while((index[i] !=sum[j]) && (j<n))
                    j++;
              for (k=0;k<m;k++)
              {
                    T[i][k]=temp[j][k];
              }
        sum[j]=9999;
        }
        for (i=0;i<n-1;i++)
        {
              temp[i][0]=store2[i];
        }
        process();
  }

  void rule5(void)
  {
        for(i=0;i<n+m;i++)
        {
              sum[i]=0;
```

```
                index[i]=0;
        }
        rulenum=5;
        for (i=0;i<n;i++)
        {
                for (j=0;j<m;j++)
                {
                        sum[i]=sum[i]+temp[i][j];
                        index[i]=index[i]+temp[i][j];
                }
        }
//      qsort(&index,n,sizeof(index[1]),compare);
        qsort( (void *)index, (size_t)n, sizeof(int), compare );
        for (j=0;j<n;j++)
        {
                store[n-j-1]=index[j];
        }
        for(j=0;j<n;j++)
        {
                index[j]=store[j];
        }
        for (i=0;i<n;i++)
        {
                j=0;
                while((index[i] != sum[j])&& (j<n))
                        ++j;
                for (k=0;k<m;k++)
                {
                        T[i][k]=temp[j][k];
                }
                sum[j]=9999;
        }
        for (i=0;i<n;i++)
        {
                temp[i][j]=store2[i];
        }
        process();
}

void rule2(void)
{
        /* it allows the program to sort the values */
        rulenum=2;
        for(j=0;j<n;j++)
        {
                index[j]=temp[j][0];
        }

        /* sorting the first row using the qsort method */

        qsort( (void *)index, (size_t)n, sizeof(int), compare );
        for(i=0;i<n;i++)
        {
                j=0;
                while((index[i] !=temp[j][0] )&& (j<n))
                        ++j;
                for(k=0;k<m;k++)
                {
                        T[i][k]=temp[j][k];
                }
```

```
                store2[j]=temp[j][0];
                temp[j][0]=9999;
        }
        for (i=0;i<n;i++)
        {
                temp[i][0]=store2[i];
        }
        process();
}

void rule3(void)
{
        rulenum=3;
        for(j=0;j<n;j++)
        {
                index[j]=temp[j][0];
        }

//      qsort(&index,n,sizeof(index[1]),compare);
        qsort( (void *)index, (size_t)n, sizeof(int), compare );
        for (j=0;j<n;j++)
        {
                store[n-j-1]=index[j];
        }
        for (j=0;j<n;j++)
        {
                index[j]=store[j];
        }
        for(i=0;i<n;i++)
        {
                j=0;
                while((index[i] !=temp[j][0] )&& (j<n))
                        ++j;
                for(k=0;k<m;k++)
                {
                        T[i][k]=temp[j][k];
                }
                temp[j][0]=9999;
        }
        process();
}

int intcmp(int a,int b)
{
        return(a-b);
}

void process(void)
{
f[0][0]=T[0][0];        /* make span formula */
for (i=1;i<m;i++)
{
        f[0][i]=f[0][i-1]+T[0][i];
}
for (i=1;i<n;i++)
{
        f[i][0]=f[i-1][0]+T[i][0];
}
for (j=1;j<m;j++)
{
```

```
	for (i=1;i<n;i++)
	{
f[i][j]=max(f[i-1][j],f[i][j-1])+T[i][j];
	}
}
}

void display(void)
{
	system("cls");
	printf("\n");
	i=m-1;
	for (j=0;j<n-1;j++)
	{
		result1=f[j][i];
	}
	result=f[n-1][i];
	{
		i=m-1;
		for(j=0;j<n-2;j++)
		{
			result1=result1+f[j][i];
		}
		result2=(result1+result)/n;
	}
	for (i=0;i<n+m;i++)
	{
		p[i]=0;
		sum[i]=0;
	}
	count=0;
	p[0]=0;
	for(i=1;i<m;i++)
	{
		p[i]=p[i-1]+T[0][i-1];
	}
	for (i=0;i<m;i++)
	{
		count=0;
		for(j=0;j<n;j++)
		{
			count=count+T[j][i];
		}
		sum[i]=count;
	}
	{
		for (i=0;i<m;i++)
		{
			for(j=0;j<n;j++)
			{
			}
			printf("\n");
			idle=f[n-1][i]-sum[i]-p[i];
			result5=f[n-1][i]-sum[i]-p[i];
		}
		printf("\n");
	}
	j=n-1;
	for (i=0;i<m;i++)
	{
```

```
                        result5=result5+f[n-1][i]-sum[i]-p[i];
                }
                {
                        for (j=0;j<n+m;j++)
                        {
                                z[j]=0;
                                su[j]=0;
                        }
                        z[0]=0;
                        for (j=1;j<n+1;j++)
                        {
                                z[j]=z[j-1]+T[j-1][0];
                        }
                        for(j=0;j<n;j++)
                        {
                                count=0;
                                for(i=0;i<m;i++)
                                {
                                        count=count+T[j][i];
                                }
                                su[j]=count;
                        }
                        {
                                printf("\n");
                                i=m-1;
                                for(j=0;j<n;j++)
                                {
                                        printf("\n");
                                        result8=f[j][i]-su[j]-z[j];
                                        {
                                                printf("\n");
                                        }
                                        i=m-1;
                                        for(j=0;j<n;j++)
                                        {
                                                result8=result8+f[j][i]-su[j]-
z[j];
                                        }
                                }
                        }
                }
        }

        int compare(const void *arg1, const void *arg2)
        {
                return (*(int *)arg1 == *(int *)arg2);
        }
```

Appendix F

Results

/* Results of Dispatching Rules Comparison */

- Average Make-Span Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 5	58	54	62	58	59
10 x 5	93	86	100	91	94
30 x 5	211	196	231	209	209
50 x 5	324	310	370	323	324
80 x 5	512	490	587	510	508
100 x 5	693	666	803	693	691
150 x 5	936	900	1082	936	934
200 x 5	1188	1140	1380	1176	1173
250 x 5	1382	1289	1622	1342	1427
5 x 20	173	165	182	173	173
10 x 20	210	198	221	208	208
30 x20	319	310	349	319	320
50 x 20	469	381	523	469	439
80 x 20	591	495	689	569	546
100 x 20	772	616	890	721	738
150 x20	1000	782	1115	946	968
200 x 20	1260	989	1368	1160	1205
250 x 20	1505	1220	1712	1370	1425
5 x 50	310	300	318	310	308
10 x 50	434	419	452	434	434
30 x 50	557	541	587	554	554
50 x 50	723	699	764	720	718
80 x 50	933	906	996	930	930
100 x 50	1134	1107	1224	1128	1134
150 x 50	1442	1407	1561	1435	1435
200 x 50	1745	1710	1864	1791	1774
250 x 50	2024	1980	2220	2044	2083
5 x 80	543	529	553	539	543
10 x 80	583	566	600	580	583
30 x 80	818	792	811	784	764
50 x 80	954	933	1002	951	954
80 x 80	1182	1117	1243	1182	1155
100 x 80	1368	1341	1464	1368	1368
150 x 80	1715	1687	1838	1712	1712
200 x 80	2092	2014	2200	2047	2007
250 x 80	2339	2293	2528	2335	2331
5 x 130	833	810	846	828	828
10 x 130	889	870	916	893	889
30 x 130	1081	1057	1119	1078	1078
50 x 130	1325	1291	1373	1317	1314
80 x 130	1591	1558	1672	1591	1591
100 x 130	1789	1758	1915	1782	1805
150 x 130	2144	2104	2272	2136	2132
200 x 130	2561	2507	2727	2459	2523
250 x 130	2834	2792	3027	2829	2825

Cont. Average Make-Span Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 200	1012	990	1034	946	1062
10 x 200	1177	1150	1205	1095	1238
30 x 200	1326	1293	1364	1232	1397
50 x 200	1440	1405	1490	1340	1515
80 x 200	1581	1534	1628	1446	1649
100 x 200	1715	1676	1793	1602	1793
150 x 200	1945	1910	2045	1825	2040
200 x 200	2244	2206	2371	2107	2354
250 x 200	2428	2388	2571	2286	2542
5 x 250	1407	1376	1437	1315	1475
10 x 250	1412	1379	1445	1313	1485
30 x 250	1521	1482	1569	1400	1592
50 x 250	1608	1561	1663	1458	1679
80 x 250	1755	1703	1807	1605	1830
100 x 250	1921	1877	2009	1795	2009
150 x 250	2237	2197	2352	2099	2346
200 x 250	2513	2470	2655	2359	2636
250 x 250	2720	2675	2879	2560	2847

- Average Mean Completion Time Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 5	46	42	51	41	51
10 x 5	66	59	74	56	74
30 x 5	149	137	174	127	167
50 x 5	198	185	237	169	220
80 x 5	298	281	364	258	335
100 x 5	386	351	462	327	422
150 x 5	506	467	619	437	575
200 x 5	669	624	810	579	759
250 x 5	832	778	980	727	916
5 x 20	135	127	144	124	146
10 x 20	149	139	158	133	162
30 x20	251	239	276	225	272
50 x 20	326	312	366	293	351
80 x 20	419	403	475	380	454
100 x 20	530	512	606	480	572
150 x20	675	653	770	603	730
200 x 20	825	798	939	738	891
250 x 20	980	949	1113	879	1061
5 x 50	323	310	333	298	342
10 x 50	363	348	380	328	389
30 x 50	434	416	460	393	463
50 x 50	521	501	557	476	557
80 x 50	669	645	723	612	714
100 x 50	746	724	814	692	789
150 x 50	911	881	997	835	967
200 x 50	1110	1079	1227	1022	1182
250 x 50	1236	1200	1368	1136	1316
5 x 80	490	475	505	456	521
10 x 80	496	483	515	454	528
30 x 80	630	609	655	574	665
50 x 80	751	728	792	690	799
80 x 80	858	832	911	792	908
100 x 80	953	924	1033	880	1005
150 x 80	1103	1075	1183	1022	1162
200 x 80	1332	1300	1436	1236	1408
250 x 80	1505	1471	1634	1398	1591
5 x 130	769	750	789	715	804
10 x 130	781	756	802	722	823
30 x 130	907	877	937	834	959
50 x 130	1046	1017	1087	968	1099
80 x 130	1180	1152	1240	1100	1244
100 x 130	1278	1244	1340	1205	1373
150 x 130	1483	1449	1571	1382	1554
200 x 130	1707	1674	1817	1596	1794
250 x 130	1867	1834	1997	1747	1963
5 x 200	1001	982	1025	1001	1006
10 x 200	1272	1248	1301	1277	1272
30 x 200	1555	1525	1605	1555	1560
50 x 200	1773	1746	1836	1769	1773
80 x 200	2115	2075	2201	2111	2106
100 x 200	2282	2241	2381	2273	2273
150 x 200	2654	2617	2797	2650	2654
200 x 200	3065	3025	3245	3060	3045
250 x 200	3424	3381	3631	3424	3419

Cont. Average Mean Completion Time Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 250	1392	1365	1424	1392	1398
10 x 250	1526	1498	1561	1532	1526
30 x 250	1866	1830	1926	1866	1872
50 x 250	2092	2060	2166	2087	2092
80 x 250	2348	2303	2443	2343	2338
100 x 250	2555	2560	2770	2600	2610
150 x 250	3052	3010	3216	3047	3052
200 x 250	3433	3388	3634	3430	3420
250 x 250	3835	3787	4066	3835	3829

- Average Waiting Time Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 5	43	37	44	21	57
10 x 5	146	165	133	90	179
30 x 5	994	1602	824	781	1105
50 x 5	2091	3881	1703	1751	2212
80 x 5	3489	7041	2836	3016	3751
100 x 5	7425	14000	5640	6502	7300
150 x 5	11437	21235	7223	9223	11516
200 x 5	16470	30414	12540	14056	16578
250 x 5	26010	50400	18524	22050	25380
5 x 20	61	59	65	34	84
10 x 20	315	349	287	207	386
30 x20	1924	2722	1547	1581	2152
50 x 20	3866	5958	3017	3236	4169
80 x 20	7618	12108	5903	6525	8280
100 x 20	12780	20948	10084	11228	13842
150 x20	22500	36700	17600	18200	24100
200 x 20	32960	54416	25640	26000	35176
250 x 20	49800	81984	38700	40300	52860
5 x 50	166	158	164	92	222
10 x 50	715	788	678	480	878
30 x 50	3426	3972	3000	2472	4026
50 x 50	7896	10068	6486	6264	8892
80 x 50	16392	23148	12888	13656	18138
100 x 50	23338	33600	18144	19936	25256
150 x 50	35784	52119	27522	30735	38736
200 x 50	54528	82092	42420	47472	59508
250 x 50	79080	119160	60510	68595	84600
5 x 80	272	269	277	154	370
10 x 80	860	943	793	578	1055
30 x 80	4368	5051	3848	3140	5181
50 x 80	10554	13063	8779	8371	11927
80 x 80	21674	27622	17597	17524	24086
100 x 80	33609	44809	26632	28325	36730
150 x 80	52839	71996	41567	45563	57443
200 x 80	82579	114787	63501	70523	88757
250 x 80	117819	169480	90174	102467	126768
5 x 130	357	344	357	192	459
10 x 130	1095	1168	1014	742	1330
30 x 130	5975	6747	5374	4360	7129
50 x 130	12558	15267	10570	10108	14112
80 x 130	24272	30976	19696	20200	26960
100 x 130	35350	46047	28177	29764	38855
150 x 130	68100	93000	43400	54100	74900
200 x 130	104380	143378	81702	91137	113407
250 x 130	163208	226458	125074	142163	176663
5 x 200	273	271	282	156	367
10 x 200	1123	1204	1050	767	1386
30 x 200	7191	8092	6528	5355	8602
50 x 200	16129	19196	13694	12849	18311
80 x 200	34466	42809	28213	28540	38291
100 x 200	50765	66127	40698	43605	55957
150 x 200	95075	124646	74920	81881	103295
200 x 200	141900	187200	111700	124300	156100
250 x 200	199079	269331	155015	174701	214226

Cont. Average Waiting Time Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 250	379	376	391	217	511
10 x 250	1348	1445	1260	920	1663
30 x 250	8629	9710	7834	6426	10322
50 x 250	19033	22652	16159	15162	21607
80 x 250	38257	47518	31317	31679	42503
100 x 250	69600	88700	52000	60800	80000
150 x 250	119600	143343	86158	106800	131800
200 x 250	173600	209664	125104	153400	193200
250 x 250	236500	267200	173616	214900	252700

- Average Idle Time Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 5	40	42	42	61	23
10 x 5	69	62	86	91	45
30 x 5	150	120	195	179	115
50 x 5	205	161	300	239	154
80 x 5	262	204	411	304	203
100 x 5	320	260	527	400	252
150 x 5	392	317	665	503	285
200 x 5	497	430	825	630	378
250 x 5	645	585	986	797	510
5 x 20	732	710	710	990	440
10 x 20	1070	960	1210	1368	706
30 x20	1763	1612	2298	2200	1316
50 x 20	2145	1930	3120	2680	1690
80 x 20	2530	2400	3930	3140	2050
100 x 20	2940	2700	4600	3630	2270
150 x20	3450	3060	5290	3970	2680
200 x 20	3912	3438	6240	4512	3264
250 x 20	4620	4140	8050	5327	3810
5 x 50	2358	2376	2484	3303	1422
10 x 50	3924	3591	4194	4340	2700
30 x 50	6756	6252	7908	7410	4872
50 x 50	9132	8508	10640	10250	7080
80 x 50	11316	10332	13560	12380	8988
100 x 50	13400	12140	16710	15290	10560
150 x 50	15610	14500	19790	18000	12770
200 x 50	18432	17260	23410	21344	15530
250 x 50	23570	22470	29000	25930	21130
5 x 80	3293	3143	3210	4388	2010
10 x 80	6318	6003	6777	8073	4500
30 x 80	11500	9600	14602	16058	8100
50 x 80	15000	12870	18800	17200	10703
80 x 80	20500	17600	28100	25100	15700
100 x 80	25920	22600	34400	30496	19600
150 x 80	29600	26443	39711	34600	23514
200 x 80	36200	32652	49860	42700	29142
250 x 80	44300	41100	56600	51100	37200
5 x 130	9792	9728	10096	13552	6544
10 x 130	15747	15355	17129	20394	11414
30 x 130	26200	24900	28700	30300	20200
50 x 130	33611	31863	39368	39881	26049
80 x 130	43200	40620	53620	50960	34780
100 x 130	52946	50439	68632	62560	43516
150 x 130	61975	58850	80725	71600	51200
200 x 130	71300	65900	95700	81700	57700
250 x 130	82593	78660	114741	95304	69654
5 x 200	15084	14688	15012	20286	9504
10 x 200	29022	27573	30072	36582	20307
30 x 200	46200	41700	52400	59100	32000
50 x 200	61568	59280	72280	74178	48776
80 x 200	78300	74900	94000	90700	63600
100 x 200	92520	88920	115320	107880	75990
150 x 200	114300	109200	138000	128400	95200
200 x 200	134280	126792	168400	153900	112464
250 x 200	151320	143715	201006	180700	127257

Cont. Average Idle Time Criteria

n x m	FCFS	SPT	LPT	SRPT	LRPT
5 x 250	18101	17626	18014	24343	11405
10 x 250	34826	33088	50000	43898	24368
30 x 250	62300	50040	79100	70920	38400
50 x 250	81200	71136	100000	89014	58531
80 x 250	108800	89880	132400	121400	76320
100 x 250	131600	106704	168200	154500	91188
150 x 250	156100	131040	209500	185200	114240
200 x 250	180500	152150	258100	227800	134957
250 x 250	210000	172458	301100	267200	152708

Appendix G

Schematic Approach for Simulation Modeling (SASM)

This appendix provides table of the main blocks used in SASM models. Each block has an icon represents its function.

Block	***Function***
Informational Block	This block reads information of items/parts then passes them through. The block can be used to change or remove attribute. The block can send information to Excel file or store it.
Information Retrieval Block	This block acts like lookup icon. It retrieve information from external or local files and then use it to calculate the output or pass it another block.
Operation	Operation block simulates a machine or manufacturing cell operating on a single item/lot. This block is usually connected with a source of information about the processing time for every item/lot, delay times due to maintenance. It also sends information about utilization, productivity and other attributes to be stored in file.
Production Settings	Production setting block provides items based on scheduling information such as product-mix, product sequence, number of items/lots at the start, and priority setting.
External File	This block can be either input or ouput file. It writes/reads/stores data from/to any block. The common forms of files are Excel, and Access files.
Global File	Global or local files are general-purpose files available anywhere in the model. They can be set, reset, or modified. Any number of blocks can access the same file.
Buffer	A buffer is a place to hold number of items/lots and release them on preset rule such as FIFO, or with a particular attribute. The buffer has a maximum queue length (size).
Decision	Decision block selects where to send the item/lot that is present at the selected input. The decision can be made based on any preset criteria. In some cases, a default decision is set to go when the situation is random.
——Data——▶	Data arrow presents the flow of data.
A	Connector allows the user to follow the model flow.