International Journal of Scientific & Engineering Research (IJSER), May 2013
ISSN 2229-5518

**Dedicated to**

**My Beloved Wife**

**Sonika Aggarwal**

**&**

**My Son**

**Devarsh Agarwal**

# ACKNOWLEDGEMENT

**“The Stars & Euphoria I feel at my successful completion of Dissertation”**

No thesis/dissertation is created entirely by an individual, many people have helped to create this thesis and each of their contribution has been valuable. This dissertation has been possible by bountiful mercy and grace of God. It is my great pleasure to express my thanks to all the magnanimous persons who rendered their full support to my work directly or indirectly.

First and foremost, I express my sincere gratitude and indebtedness to my thesis supervisor, **Mr. Rajan Garg, Assistant Professor & Head of Department of Mechanical Engineering, Shanti Devi Institute of Technology & Management (NC & SD Group of Institutions), Israna, Panipat (Haryana)** for proposing this area of research, for his kind and valuable guidance in the completion of the present work, for his deep involvement, invaluable and continuous motivation throughout this work. His consistent support and intellectual guidance made me energize and to innovate new ideas.

I would like to take this privileged opportunity to express my sincere gratitude and to say that the work would not have reached the zenith without the worthy guidance of **Prof. H. M. Khurana – Dean (Faculty of Engineering & Technology) and Head of Department of Mechanical Engineering, MBU, Solan (Himachal Pradesh)** for his unstinted support and commitment to make my work see light of today. I am also thankful to all the faculty members and staff of Mechanical Engineering Department, for providing me support and advice in preparing my thesis work. Last, but not the least I would like to thank my beloved wife and my naughty son who are involved directly and indirectly in successful completion of the present work.

**Ajay Kumar Agarwal**
(Author)

# ABSTRACT

Global competition has engendered a trend among companies towards continuously improving operations so that they can provide customers the right products in the right quantities at the right times. In modern manufacturing the trend is the development of Computer Integrated Manufacturing, CIM technologies which is a computerized integration of manufacturing activities (Design, Planning, Scheduling and Control) produces right products at right time to react quickly to the global competitive market demands. The productivity of CIM is highly depending upon the scheduling of Flexible Manufacturing System (FMS). Shorting the make span leads to decreasing machines idle time which results improvement in CIM productivity. Conventional methods of solving scheduling problems based on priority rules still result schedules, sometimes, with significant idle times. To optimize these, these pages model the problem of a flowshop scheduling with the objective of minimizing the make span. The work proposed here deal with the production planning problem of a flexible manufacturing system. These pages model the problem of a flowshop scheduling with the objective of minimizing the make span. The objective is to minimize the makespan of batch-processing machines in a flowshop. The processing times and the sizes of the jobs are known and non-identical. The machines can process a batch as long as its capacity is not exceeded. The processing time of a batch is the longest processing time among all the jobs in that batch. The problem under study is NP-hard for makes pan objective. Consequently, comparison based on Gupta's heuristics, RA's heuristics, Palmer's heuristics, CDS's heuristics are proposed in this work. Gantt chart is generated to verify the effectiveness of the proposed approaches.

# CONTENTS

**Description** **Page No.**

## UNIT I

# UNIT II

# LIST OF FIGURES

## UNIT I



## UNIT II

# LIST OF TABLES

## UNIT II

# UNIT - I

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction

CIM stands for Computer Integrated Manufacturing. In 1973, Joseph Harrington published the initial concepts of CIM in his book called Computer Integrated Manufacturing (Harrington, 1979). It was not until about 1984 that people began to realize the potential benefits these concepts promised. Since 1984, thousands of articles have been published on this subject. Thanks to the contribution of researchers and practitioners from industries, CIM has become a very challenging and fruitful research area. Different people with different disciplines proposed their understanding about CIM. They use their knowledge to solve different problems in industry practice and contribute to the development of CIM methodologies and theories.

This chapter is organized as follows:
In the section 1.2, manufacturing environments, CIM definitions, concepts, and integration issues are discussed.
In section 1.3, CIMS (CIM System) components and their functions are introduced.
In section 1.4, flexible manufacturing system is discussed.



## 1.2. CIM Definitions and Concepts

### *1.2.1 Manufacturing Environment*

Before the definitions and concepts of CIM are discussed. It is beneficial to give a general description for the environment that CIM concepts are applied. From the acronym C (Computer) I (Integrated) M (Manufacturing), it can be seen that the application area of CIM is manufacturing, or manufacturing company. The manufacturing companies today have faced intensive market competition; major changes are being experienced with respect to resources, markets, manufacturing processes, and product strategies. Manufacturing companies must respond to the rapidly changing market and to the new technologies being implemented by their competitors. Furthermore, manufacturing, which has been treated as an outcast by corporate planning and strategy, must become directly involved in these critical long-range decisions. Manufacturing can indeed be a “formidable competitive weapon”, but only if we plan for it and provide the necessary tools and technologies.

Besides the traditional competition requirements for low cost and high quality, the competition pressure for today’s manufacturing companies are more complex products, shorter product life cycle, shorter delivery time, more customized products and fewer skilled workers. The importance of these elements varies among industries. Even among different companies in the same industry their impact varies from each company’s strategy.

Today's products are becoming much more complex and difficult to design and manufacture. One example is the automobile that is becoming more complex with computer-controlled ignition, braking, and maintenance systems. To avoid long design time for the more complex products, companies should develop tools and use new technologies, such as concurrent engineering, and at the same time it should also improve their design and manufacturing processes.

Higher quality is the basic demand of the customers who want their money be worth for the products they buy. This applies to both consumers and industrial customer. The improved quality can be achieved through better design and better quality control in the manufacturing operation. Besides the higher quality demand, customers are not satisfied with the basic products with no options. There is a competitive advantage in having a broad product line with many versions, or with a few basic models that can be customized. A brand new concept in manufacturing is to involve users in the product design, with the aid of design tools or modeling box, the company will allow the users to design the products with their own favor.

In the past, once a product was designed, it had a long life over which to recover its development costs. Today many products, especially high-technology products, have a relative short life cycle. This change has two implications. First, companies must design products and get them to the market faster. Second, a shorter product life provides less time over which we can to recover the development costs. So, the companies should use new technologies to reduce both the time and cost in product design. The concurrent engineering is one of the methods in improving product design efficiency and reducing product costs. Another method is to distribute the cost and risks of new product development to partners, and to share benefits among the partners, this new manufacturing paradigm is called agile manufacturing. This paradigm requires the change or reengineering of traditional organization structures.

Several demographic trends are serious affecting manufacturing employment. The education level and expectations of people are changing. Fewer new workers are interested in manufacturing jobs, especially the unskilled and semiskilled ones. The lack of new employees for the skilled jobs that is essential for a factory even more critical. On the other hand, many people may not have sufficient education background so that they are not qualified for these jobs (Bray, 1988).

In order to win in the global market, manufacturing companies should improve their competition ability. Some key elements include creative new products, higher quality, better service, great agility and low pollution to the environment. Creative new product is of vital importance to companies in the current "knowledge economy" era.

Fig.1.1, presents a market cycle change graph. From this Figure, it can be seen that the numbers for lot size and repetitive order is decreasing, the product life cycle is shortening, and the product variety is increasing rapidly.

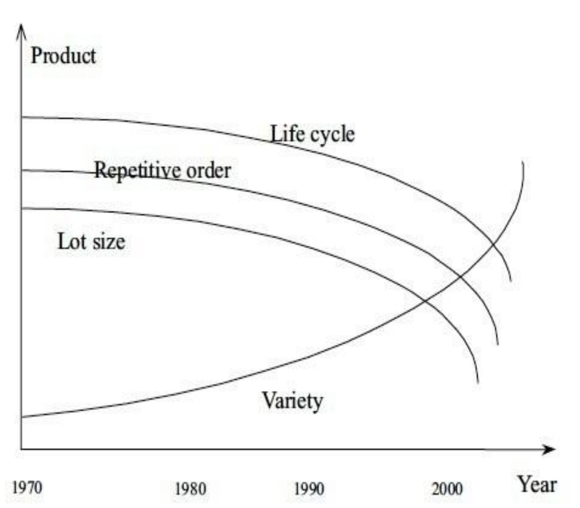

**Fig. 1.1: Market Cycle Change Graph**

The end user or customers always need new products with advancements in function, operation, energy consumption. The company can get higher benefit through new products. To some extent, we believe that a manufacturing company without new products has no chance to survive in the future market. Better services are needed for any kind of companies. However, for manufacturing companies, better service means fast delivery of products, easy use of the products and satisfying, customer need with low price, and rapid response to customer maintenance request.



### *1.2.2 Features of a General Manufacturing System*

The manufacturing company in itself is a quite complex system. It is a complex, dynamic, and stochastic entity consisting of a number of semi-independent subsystems interacting and intercommunicating in an attempt to make the overall system function profitably. The complexity comes from the heterogeneous environment (both hardware and software), huge quantity of data, and the uncertainty external environment. The complex structure of the system and the complex relationships between the interacting semi-autonomous subsystems are also the affecting factors to make the system more complicated.

A simple model of a manufacturing system can be a black box that takes input materials, energy, and information and gives output products. The internal details of the manufacturing system depend on the particular industry involved, but the key features common to all manufacturing organizations are that the system processes both materials and information. The general manufacturing systems can be decomposed into seven levels of decision hierarchies (Rogers, Upton, and Williams, 1992) (Fig.1.2). Decisions at the upper levels are made at less frequent intervals (but have implications for longer periods into the future) and are made on the basis of more abstract (and slower to change) information on the state of the system. Decisions at the lower levels are made more frequently using much more detailed information on the state of the system.

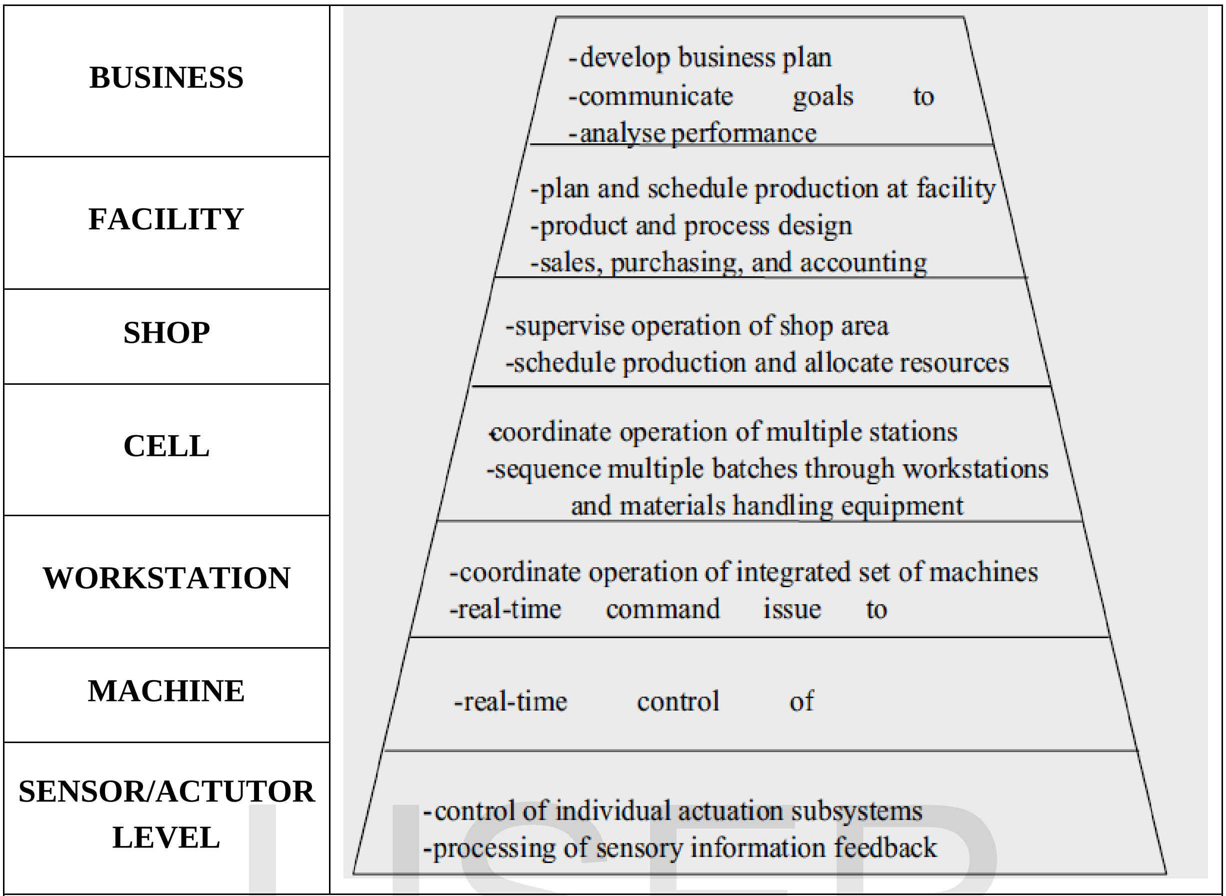


**Fig.1.2: A Seven Level Manufacturing Hierarchy**



Three kinds of decisions should be made for any manufacturing company:

a) what kinds of products will be made,
b) what resource is needed to make the products, and
c) how to control the manufacturing systems

It should be pointed out that these decisions cannot be made separately. If the company wants to make a decision at a certain level, for example, at business level, it should also get access to the information at other levels. In the whole processes of decision making, the core concept is integration. This is the fundamental requirements for the research and development of Computer Integrated Manufacturing.

### 1.2.3 CIM Definitions

There are many definitions for CIM emphasizing different aspects of it as a philosophy, a strategic tool, a process, an organizational structure, a network of computer systems, or as a stepwise integration of subsystems. These different definitions are given by peoples working at different areas and at different times from different viewpoints. Since the concept of CIM was put forward in 1973, it has been enriched due to the contributions of many researchers and practitioners. One earlier definition of CIM given by Kochan and Cowan (1986) is: the concept of a totally automated factory in which all manufacturing processes are integrated and controlled by a CAD/CAM System. CIM enables production planners and schedules, shop floor, foremen, and accountants to use the same database as product designers and engineers. This definition does not put much emphasis on the role of information.

Another definition given by Digital Equipment Corporation (DEC) (Ayres, 1991) has put much emphasis on the role of information. The definition is: ***CIM is the application of computer science technology to the enterprise of manufacturing in order to provide the right information to the right place at the right time, which enables the achievement of its product, process and business goals.*** This definition points out the importance of information in manufacturing enterprise, but unfortunately it does not give much emphasis to the very important concept of integration.

Some other definitions have pointed out that CIM is a philosophy in operating a manufacturing company. One definition given by Greenwood (1988) is: ***CIM is an operating philosophy aiming at greater efficiency across the whole cycle of product design, manufacturing, and marketing, thereby improving quality, productivity, and competitiveness.***

In order to stress the importance of integration, the Computer and Automation Systems Association of the Society of Manufacturing Engineers has given the following CIM definition (Singh. 1996). ***CIM is the integration of the total manufacturing enterprise through the use of integrated systems and data communications coupled with new managerial philosophies that improve organizational and personnel efficiency.***

It should be pointed that CIM does not mean to replace man by machine or computer, so as to create a totally automatic business and manufacturing processes. It is not necessary to build a fully automatic factory in order to implement a CIM system. Especially, it is not wise to put a huge amount of investment in purchasing highly automation flexible manufacturing systems to improve manufacturing standards, if the bottleneck for the company’s competition is not in this area. In current situation, the design standards for creative and customized products are more important than production ability in winning the market competition.

The importance of human factors should be much emphasized. Human plays a very important role in CIM design, implementation, and operation. Although computer applications and artificial intelligence technologies have gained much progress, even in the future, computer cannot replace people. In order to stress the importance, someone has presented the idea of human centered CIM.

From the above definitions, two views can be drawn from CIM concepts. They are the system view and information view. In CIM concepts, the whole activities of a company form a system. The different functions and activities cannot be analyzed and improved separately. The whole company can operate in an efficient and profitable way only if these different functions and activities are running in an integrated and coordinated environment, and these activities are optimized in a global system range. The SME CIM wheel provides a clear portrayal of relationships among all parts of an enterprise. It illustrates a three-layered integration structure of an enterprise as shown in Fig.1.3.

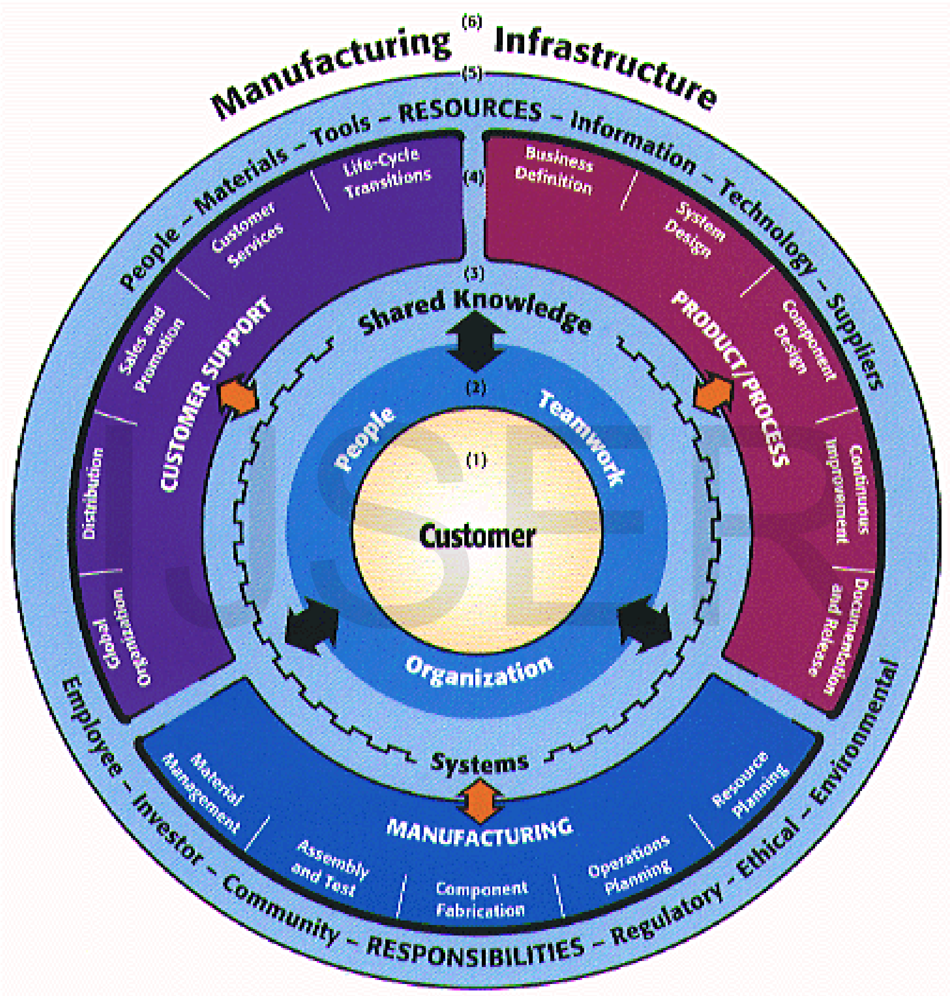


**Fig.1.3: CASA/SME CIM Wheel**

The outer layer represents general management, and human resources management. The middle layer has three process segments: product and process definition, manufacturing planning and control, and factory automation. These process segments represent all the activities in the design and manufacturing phases of a product life cycle making the product from a concept to its assembly. The center of the wheel represents the third layer, which includes information resources management and the common database.

Another important view of CIM concepts is the information view. As stated in the definition given by Digital Equipment Corporation, the objective of CIM implementation is to enable the right information to be sent to the right person at the right time. The information system plays a vitally important role in the operation of CIM. Although there are many kinds of activities in managing a manufacturing company, each activity has different function in business management and production control, the associated function unit for the information system of CIM normally can be classified into three kinds of tasks: information collection, information processing, and information transfer.

Information collection task is the basic function of an information system, the collected information forms the basis of decision making at different levels from business management to device control. There are many methods of information collection depending on the information sources and technologies used. Device sensors may provide data regarding device status; barcode scanner may provide data about the production status of the on line products, form scanner and database table view interfaces may provide data about order, raw material purchasing, and user requirements. Some data may also come from e-mail systems. The data collected can be stored in different data formats and in different repositories.

The second task of information systems is information processing, which is closely related with the business functions of a company. The business functions vary from strategy planning, process planning, product design, warehouse management, material supply, to production management and control. The upper-steam process data is processed by algorithms or human interference; the produced instructions are used for the down-steam process. In the data processing process, different decisions will be made. The decisions made can be used in optimizing the production processes or satisfying some user requirements, such as delivery time and quality requirements.

The third task of information system is data transfer between different function units. It has three main functions, i.e., data output from application software in certain data format to one kind of data repository, data format transformation, and data transfer from one application to another application within the same computer or in a network environment.

### 1.2.4 Integration-Core of CIM

Nowadays most people believe the core is integration. In our opinion, computer technology is the basis of CIM, manufacturing is the aim of CIM, and integration is the key technology. Why should integration be considered as the core of CIM? This can be seen from different aspects. As we stated above, a system view is an important view in CIM concepts. By system we mean the whole company, including man, business, and technology. In order to form a coordinated system, the man, business, and technology in a company must be integrated. So the first aims of the integration are the integration of three basic elements (man, business, and technology) of the company. Another aim of the integration is the integration of material flow, information flow, and capital flow. Although the aims of integration seem to be clear, the technology for realizing the integration is far from mature. We still have a long way to go in the route of integration.

CIMOSA (AMICE, 1993) identifies that enterprise integration has to be an ongoing process. Enterprise will evolve over time according to both internal needs and external challenges and opportunities. The level of integration should remain a managerial decision and should be open to change over a period of time. Hence, one could find in some parts of a CIM enterprise. a set of tightly coupled systems and elsewhere, a set of loosely coupled systems according to choices made by this particular enterprise. The need to implement multi-vendor systems both in terms of hardware, software and an easy re-configuration requires the prevision of standard interfaces. To solve the many problems of the industry, integration has to recognize and proceed on more than one operational aspect. The AMICE (European Computer Integrated Manufacturing Architecture) project identifies three levels of integration covering physical systems, application and business integration (See Fig.1.4).

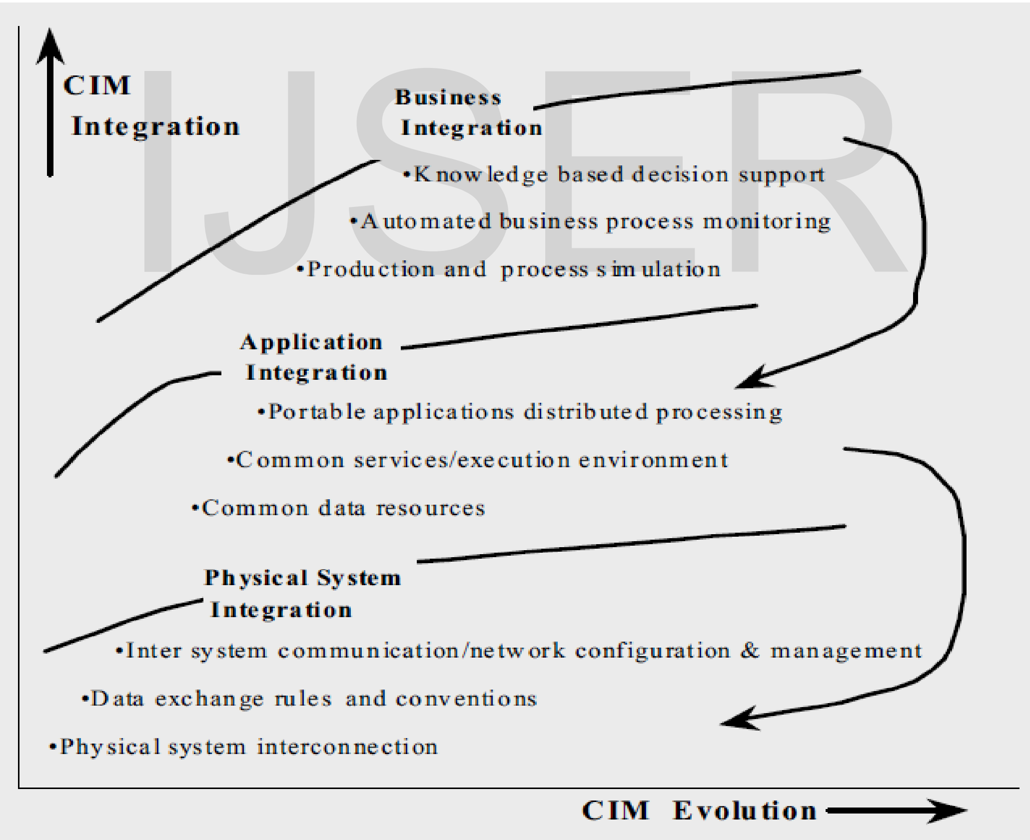


**Fig.1.4: Three Levels of Integration**

Business integration is concerned with the integration of those functions that manage, control and monitor business processes. It provides supervisory control of the operational processes and coordinates the day-to-day execution of activities at the application level.

Application integration is concerned with the control and integration of applications. Integration at this level means providing a sufficient information technology infrastructure to permit the system wide access to all relevant information regardless of where the data reside.

The physical system integration is concerned with the interconnection of manufacturing automation and data processing facilities to permit interchange of information between the so called 'islands of automation' (inter system communications). The interconnection of physical systems was the first integration requirement to be recognized and fulfilled.

Even when business integration has been achieved at one point in time, business opportunities, new technologies, modified legislation will make integration a vision rather than an achievable goal. However, this vision will drive the management of the required changes in the enterprise operation.

The classification of integration can also be given in another method that is different from that given by CIMOSA. Regarding integration objectives and methods, integration can be classified as information integration, process integration, and enterprise-wide integration.

Information integration enables data to be shared between different applications. The transparent data access and data consistency maintenance under heterogeneous computing environment is the aim of information integration. The information integration needs the support of communication system, data represent standards, and data transfer interfaces. Communication system provides data transfer mechanism and channel between applications located at different computer nodes. Data represent standards are severed as common structures for data used by different applications. Data transfer interfaces are used to transfer data from one application to another. They fulfill two kinds of functions; one function is the data format transfer (from application specified data structure to common structure and vice versa), another function is the data transfer from application to interface module and vice versa. The traditional information integration methods include database data integration, file integration, and compound data integration. The most efficient support tool for information integration is the integration platform (Fan and Wu, 1997).

Process integration is concerned with the collaboration between different applications in order to fulfill some business functions, such as the product design or process control. The need for implementing process integration comes from the pursuit of the company for shorter product design time, higher product quality, shorter delivery time, and high business process efficiency. Business Process Reengineering (BPR) (Jacobson, 1995) and Concurrent Engineering (CE) (Prased, 1996) have promoted the research and application of process integration. Business process modeling, business process simulation, and business process execution are three main research topics related to process integration.

There are a number of methods that can be used in modeling business processes. They are CIMOSA business process modeling, IDEF3 (Mayer, Cullinane. et al. 1992), Petri nets (Zhou, 1995), event driven process chain (Keller, 1995), and workflow (Georgakopoulos, Hornick, and Sheth, 1995) modeling methods. The modeling objective is to define the activities within a business process and the relationships between these activities. The activity is a basic function unit within a business process. The control and data flow between these activities form the business process which fulfils the business task of a company. The optimization of the flow path and shortening the flow time can help the company in increasing their working efficiency and reducing cost.

The third integration is called enterprise-wide integration. Following the concept of agile manufacturing, the need of virtual organization is ever more important than before. In agile manufacturing mode, a number of companies collaborate in a virtual company form to obtain a new chance in the market. The enterprise-wide integration is required to enhance the information change between the companies. Success of virtual organizations is predicated on empowerment of people within the enterprise with the aid of computer technology including communication networks, database management systems, and groupware. These facilitate team members of the virtual organization to make effective and faster group decisions. Such interaction lays the foundation for enterprise-wide integration, encompassing various plants and offices of an enterprise, possibly located in different counties and cities, as well as customers and suppliers worldwide. Therefore, enterprise-wide integration is much broader than factory automation integration. It is the integration of people, technology, and the business processes throughout the enterprise.



Enterprise-wide integration is required to ensure that all the technical and administrative units can work in unison. This, however, requires a great deal of information about a large number of activities, from product conception through manufacturing, customer delivery, and in-field support. All these life-cycle steps require a large volume of data. The transformation process from one stage to another yields volumes of new data. Furthermore, many of these design manufacturing, distribution, and service activities, responsible for generating and using volumes of data are scattered across a wide spectrum of physical locations. The information is generated by a diverse set of highly specialized software tools on heterogeneous computing hardware systems. Often, incompatible storage media with divergent data structures and formats are used for data storage. This is due to the peculiarities of the tools and systems that generate data without any regard to the needs of the tools or systems that would eventually use the data.

The main idea of enterprise-wide integration is the integration of all the processes necessary for meeting the enterprise goals. Three major tools for integration, required for overcoming the local and structural peculiarities of an enterprise’s data processing applications, are network communications, database management systems, and groupware. A number of methods regarding enterprise-wide integration have been much proposed. They are supply chain management, global manufacturing, and virtual information system supporting dynamic collaboration of companies. The Internet, Web, and CORBA (Otte, Patrick, and Roy, 1996) technologies are playing important roles in the realization of enterprise-wide integration.

## 1.3. CIMS Structure and Functions

### *1.3.1 CIMS Structure*

The components of CIMS include both hardware and software. The hardware includes computer hardware, network, manufacturing devices, and peripherals. The software includes operating systems, communication software, database management systems, manufacturing planning and control software, management information software, design software, office automation software, decision support software, and so on. These different hardware and software systems have different functions. They work together to fulfill the company's business goals. In order to understand such a complex CIM system, people normally decomposes CIMS into a number of subsystems interacting with each other. Unfortunately, there does not exist a unique and standard decomposition method for CIMS. Every company can define system decomposition method according to its specific situation and requirements. In our opinion, one decomposition method of CIMS may be as shown in Fig.1.5.

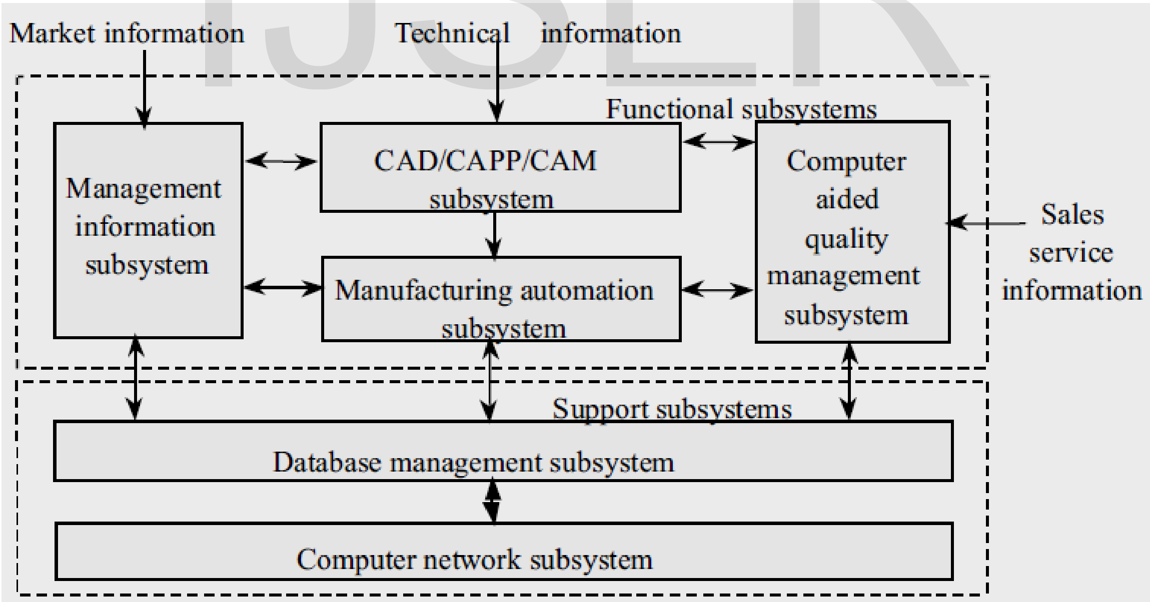


**Fig.1.5: Decomposition of CIMS**

From Fig.1.5, it can be seen that CIMS consists of four functional sub-systems and two support sub-systems. Four functional subsystems are Management Information Subsystem, CAD/CAPP/CAM Subsystem, Manufacturing Automation Subsystem, and Computer Aided Quality Management Subsystem. These functional subsystems cover the business processes of a company. Two support subsystems are Computer Network Subsystem and Database Management Subsystem; they are the basis for the functional subsystems to fulfill their tasks. The arcs denote the interfaces between different subsystems. Through these interfaces, shared data are exchanged between different subsystems.

### *1.3.2 Components of CIMS*

In this section, we give a brief description about the components of CIMS.

#### *1.3.2.1 Management Information System*

Management Information System (MIS) plays an important role in the company's information system. It manages business processes and information based on market strategy, sales prediction, business decision, order processing, material supply, finance management, inventory management, human resource management, company production plan, and etc. The aims of MIS are to shorten delivery time, reduce cost, and help the company to make rapid decision to react to market change. Currently, ERP (Enterprise Resource Planning) software is normally used as the key application software in MIS. There are many commercial ERP software products in the market, such as SAP R/3 developed by SAP Company and Baan ERP developed by Baan Company.



- *Basic Concept of ERP*

  In balancing manufacturing, distribution, financial and other business functions to optimize company productivity. ERP systems are considered to be the backbone of corporate infrastructure. ERP concept is derived from MRPII (Wright, 1992) (Manufacturing Resources Planning) system. It extends the MRPII functions. Besides the traditional functions of MRPII in manufacturing management, material supply management, production planning, finance, and sales management, ERP introduces new functions, such as transportation management, supply chain management, corporate strategy planning, workflow management, and electronic data exchange, into the system. So the ERP system provides more flexibility and ability to the company in business process reengineering, integration with customers, and integration with material suppliers as well as product dispatchers.

- *Manufacturing Resource Planning*

  The basis of MRPII is MRP (Material Requirements Planning) which comes out in l940's. MRPII uses computer-enhanced materials ordering and inventory control methods, it has the advantages in enhancing the speed and accuracy of issuing raw materials to factory work station. It immediately becomes apparent that linking materials with production demand schedules could optimize the flow of the product as it is being constructed in the factory. This could be done in such a manner that material queue times could be minimized (e.g., have the material show up only when needed), and the amount of material needed throughout the factory at any one time could be reduced ultimately. This is an optimization technique that allocates identified sets of materials (sometimes called kits) to specific jobs as they go through the manufacturing processes.

Since it is possible for a computer to keep track of large numbers of kits, it is reserves or mortgages materials for specific jobs in time-order sequences. Linking these sequences with a production plan based on customer need dates allows management to release and track orders through the shop accurately. Prior to releasing orders by means of the kiting process based on the production schedule, it was necessary to obtain supplies. The supplies are based on a gross basis depending on the number of orders expected to be shipped to customers over the selected time period and by having the gross amount of inventory on hand at the start of the period to support production. Obviously, the kit will result in less extra materials on hand at any point in the production period. This results in large raw material reductions and reductions in material needs for work in process and, hence, lower operation costs.

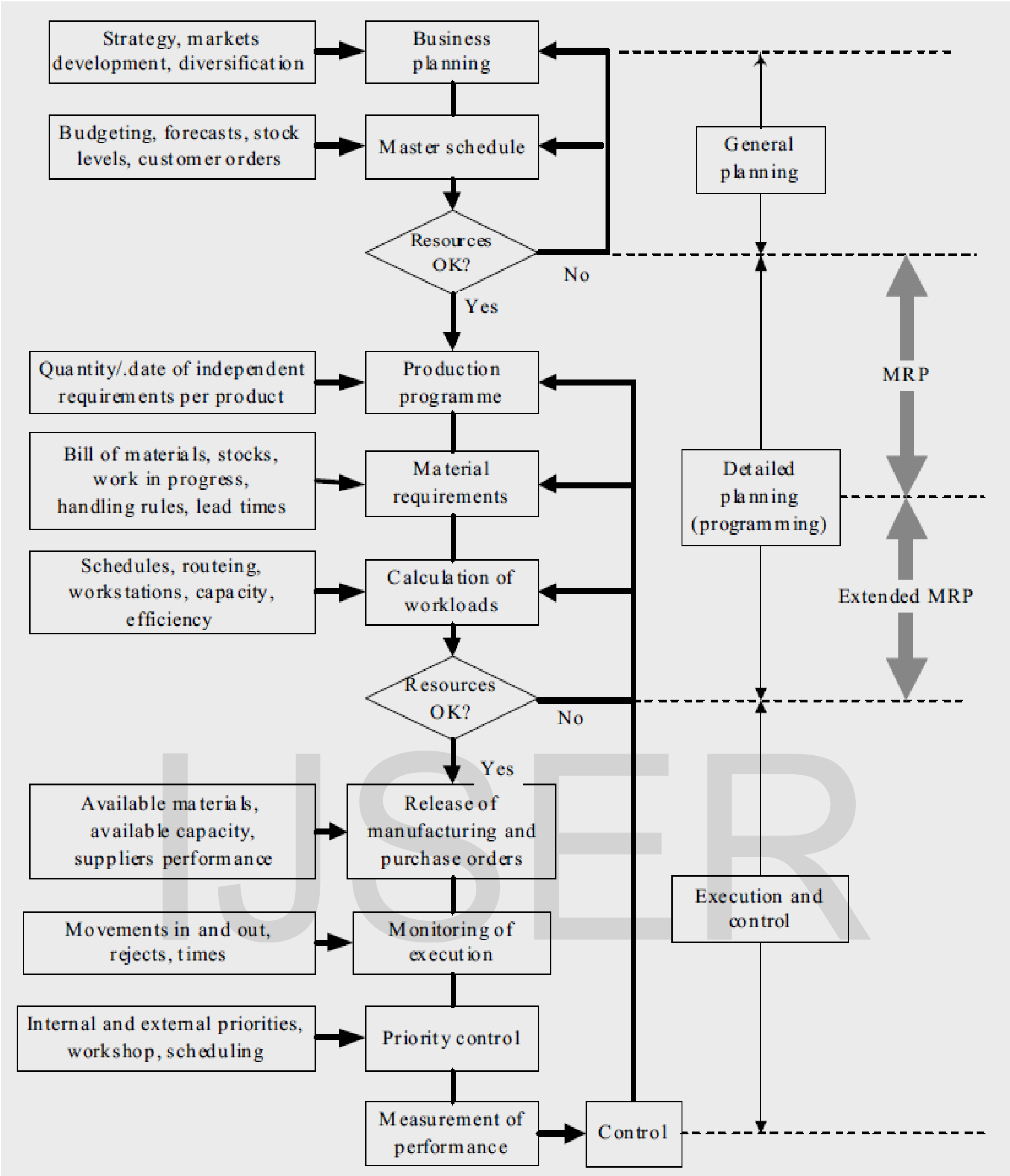


**Fig.1.6: Flow Diagram of MRPII System (Waldner, 1992).**

- *Just-in-Time*

  Another method that received much attention for production planning and control is Just-in-Time theory. In contrast to MRPII which can be referred to as a "push" oriented, the JIT philosophy of management is totally "pull" oriented. i.e., to manufacture something only when there is a firm order for it. JIT is a productivity enhancer based on a simple proposition that all waste in the manufacturing process must be eliminated.

  JIT theory states that wastes can only begin to be eliminated if the push production control system is replaced with a pull production control system. It can be seen that a very large volume of wastes are bloated at inventory levels. Therefore, we must find a way to minimize inventory levels. If you do this without the analytical capability of the computer, then it is logical to assume you would conceive a system that will not let material move or be used until it is necessary. This is what Toyota did. They instituted a backward scheduling technique that started with the desired ship date. They had to know when the product needed to be at final assembly and before that when it needed to be at the subassembly levels and so forth, back through component part manufacturing. Ultimately, it means determining precisely when the raw materials should show up at the receiving dock. This, in itself, is not unusual or unique.

Although JIT proposed ways to reduce wastes in greatest extent, it can not be implemented without the help of CIM and MRPII systems. For example, the means for producing products only at the rate the customer wants them can be best realized using the feedback control system production schedule of MRPII. By using the MRPII system, we can monitor progress of all workstations carrying out the dictates of the strategic plan and thus speed up or slow down the preceding operation to optimize the usage of materials and labor. In his book, Koenig (1990) explained in detail about the relationship of JIT with MRPII and CIM systems.

Since JIT and MRPII have their advantages as well as limitations in applications, it is proposed that the combination of JIT and MRPII systems in the common framework of CIM may produce excellent results in production scheduling and control.

***1.3.2.2 CAD/CAPP/CAM System***

CAD/CAPP/CAM stands for Computer Aided Design/Computer Aided Process Planning/Computer Aided Manufacturing. CAD/CAPP/CAM system is sometimes called design automation system. It means that CAD/CAPP/CAM is used to promote the design automation standard and provide means to design high quality products faster.

- *Computer Aided Design*

  CAD is a process that uses computers to assist in the creation, modification, analysis, or optimization of a product design. It refers to the integration of computers into design activities by providing a close coupling between the designer and the computer. Typical design activities involving a CAD system are preliminary design, drafting, modeling, and simulation. Such activities may be viewed as CAD application modules interfaced into a controlled network operation under the supervision of a computer. General CAD system consists of three basic components: hardware, which includes computer and input-output devices, application software, and the operating system software (Fig.1.7). The operating system software acts as the interface between the hardware and the application software system.

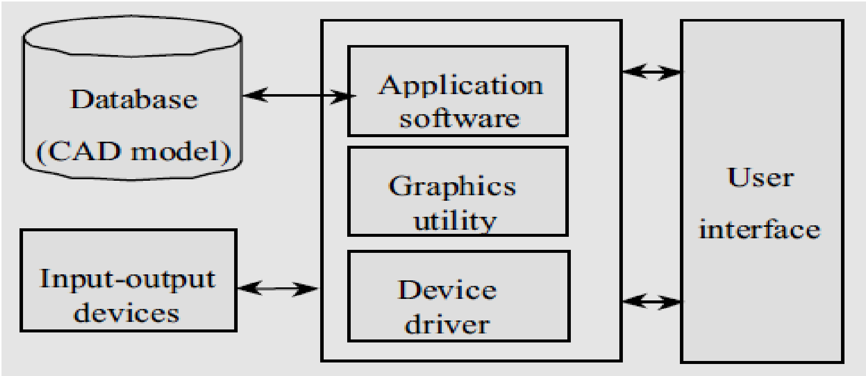


**Fig.1.7: Basic Components of CAD**

The CAD system function can be grouped into three categories. They are geometric modeling, engineering analysis, and automated drafting. Geometric modeling constructs the graphic images of a part using basic geometric elements, such as points, lines, and circles under the support of CAD software. Wireframe is one of the first geometric modeling methods. It uses points, curves, and other basic elements to define objects. Then the surface modeling, solid modeling, and parametric modeling methods are presented in the area of geometric modeling area.

Engineering design completes the analysis and evaluation of product design. A number of computer-based techniques are used to calculate the product's operational, functional, and manufacturing parameters. The major analysis includes finite-element analysis, heat transfer analysis, static and dynamic analysis, motion analysis, tolerance analysis. In the analysis, finite-element analysis is the most important method, by dividing an object into a number of small building blocks, called finite elements; FEA will fulfill the task of the functional performance analysis of an object. Various methods and packages are developed to analyze different performance of the product design. The introduction of the objectives and methods can be found in any comprehensive book discussion CAD techniques. After the analysis, the product design will be optimized according to the analysis results.

The last function of CAD system is the automated drafting. The automated drafting function includes the 2-dimension and 3-dimension product design drafting, converting of 3-dimension entity model into 2-dimesion representation.

- *Computer Aided Process Planning*

  Computer Aided Processing Planning is responsible for detailed plans for the production of a part or an assembly. It acts as a bridge between design and manufacturing by translating design specifications into manufacturing process details.

This operation includes a sequence of steps to be executed according to the instructions in each step and is consistent with the controls indicated in the instructions. Closely related to the process planning function are the functions that determine the cutting conditions and set the time standards. The foundation of CAPP is group technology (GT), which is the means of coding parts on the basis of similarities in their design and manufacturing attributes. A well-developed CAPP system can reduce clerical work in manufacturing engineering and provide assistance in production.

One of the first tasks of CAPP system is to complete the selection of raw work piece. According to the functional requirements of the designed part, it determines the attributes of the raw work piece, such as shape, size (dimension and weight), and materials. Other jobs for the CAPP system are determining manufacturing operations and their sequences, selecting machine tools, selecting tools, fixture and inspection equipment. Manufacturing conditions and manufacturing time determination are also part of the work of CAPP; these conditions will be used in optimizing manufacturing cost.

CAPP system consists of computer programs that allow planning personnel interactively to create, store, edit, and print fabrication and assembly planning instructions. Such a system offers the potential for reducing the routine clerical work of manufacturing engineers. Fig.1.8 presents the classification of various CAPP systems.

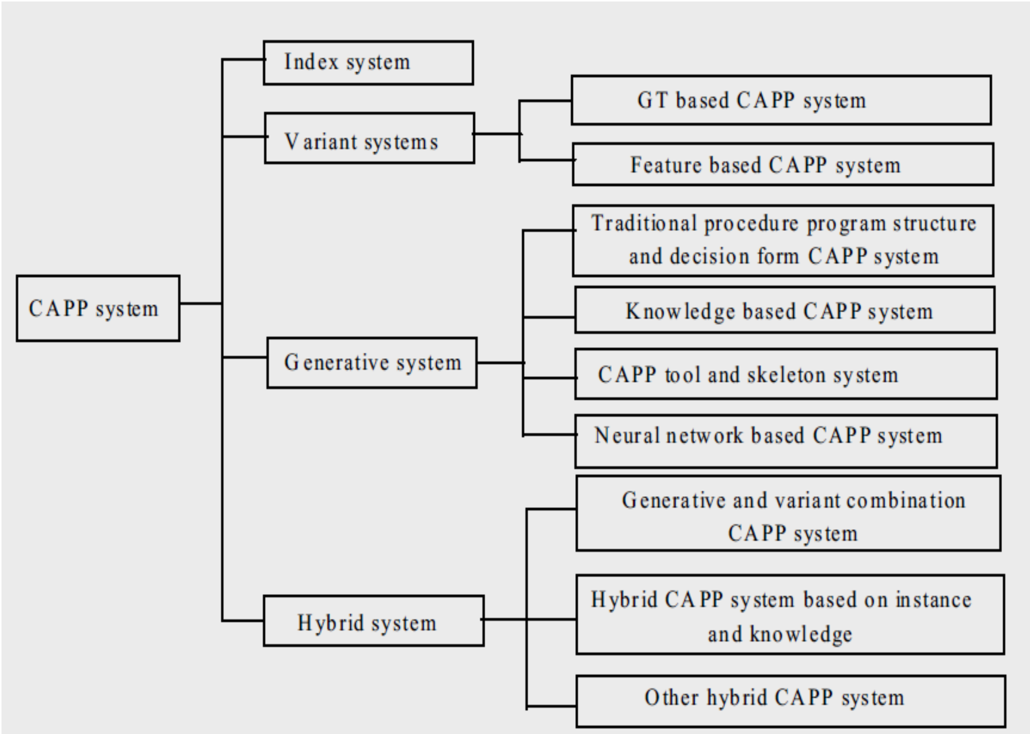


**Fig.1.8: Classification of CAPP System**



- *Computer Aided Manufacturing*

  In this section, we refer Computer Aided Manufacturing to very restricted areas which do not include general production control functions. The production control functions will be introduced in the Manufacturing Automation Subsystem (MAS) section. Here. CAM is referred to preparing data for MAS, including producing NC code for NC machine, generating tool position, planning tool motion route, and simulating tool movement. Automatic NC code generation is a very important work in increasing work efficiency. Before the NC code for numerical control machine centers can be generated, a number of parameters regarding machine tool specification, performance, computer numerical control system behavior, and coding format should be determined first. According to these parameters, geometric dimensions, solid forms, and designed part specifications, the manufacturing method and operations will be selected. The CAM system will calculate the tool position data. Then the data regarding the part dimension, the tool motion track, cutting parameters, and numerical control instructions are generated in a program file. This file is called NC program that is used by the machine tool to process part automatically.

- *CAD/CAPP/CAM Integration*

  Besides the utilization of CAD, CAPP, and CAM technology alone, the integration of CAD, CAPP, and CAM is an important way in enhancing the company's product design standards. There are three methods that can be used in the integration of

CAD/CAPP/CAM: exchange product data through specific defined data format; exchange product data through standard data format, such as STEP, IGES, DXF; define unified product data model to exchange product information.

Fig.1.9 is a STEP based CAD/CAPP/CAM integration system developed at State CIMS Engineering Research Center of China (located at Tsinghua University, Beijing), it is developed as a part of the CIMS application integration platform (Fan and Wu, 1997) for manufacturing enterprises. This system focuses on part-level CAD/CAPP/CAM integration, XPRESS language and STEP development tool ST-developer are used to define and develop the integration interfaces. Different kinds of CAD, CAPP, and CAM system can be integrated using the interfaces provided.

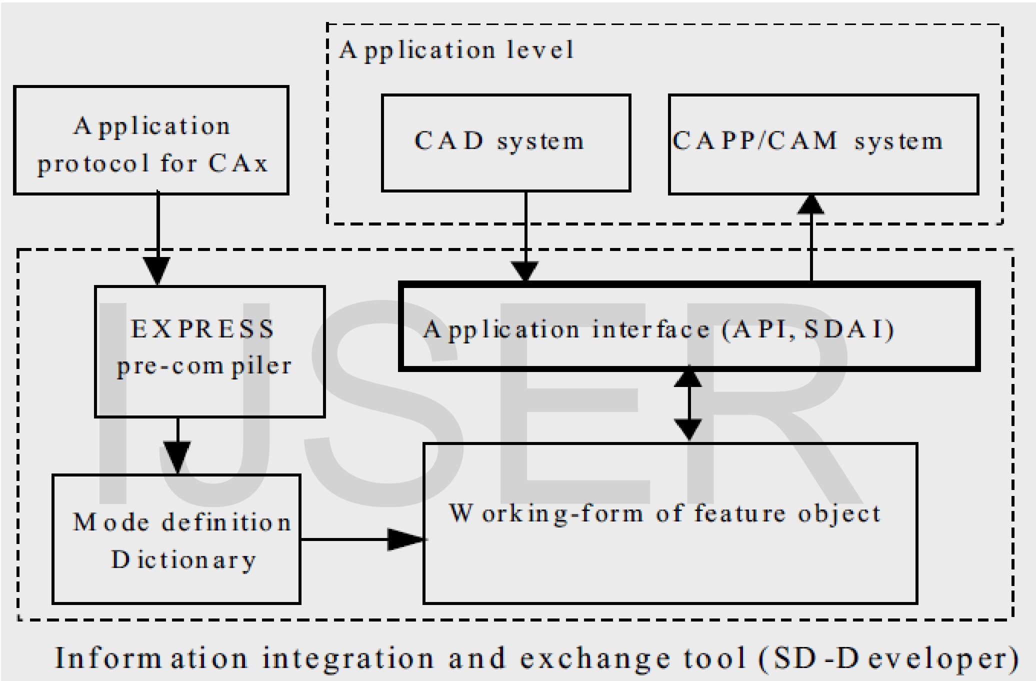


**Fig.1.9: CAD/CAPP/CAM Integration System**

### *1.3.2.3 Manufacturing Automation System*

Manufacturing Automation System is the value-added system. The material flow and information flow comes together in MAS. For discrete manufacturing company, MAS consists of a number of manufacturing machines, transportation system, high-bay store, control devices, computers, and MAS software. The whole system is operated under the control and monitor of MAS software system. For process industry, MAS consists of a number of devices controlled by DC S, monitor system, and control software system. The objectives of MAS are to increase the productivity, reduce cost, reduce work-in-progress, improve product quality and reduce production time.

MAS can be described from three different aspects: structural description, function description, and process description. Structural description defines the hardware, software system associated with the production processes. Function description defines the MAS with a number of functions that combine together to finish the task of transforming raw material into products. The input-output mapping presented by every function associates with the production activity of the MAS. Process description defines the MAS with a series of processes covering every activity in the manufacturing process.

In the research field of MAS, a very important topic is the study of the control methods for manufacturing devices, from NC machine to automatic guided vehicle. But in this chapter, our focus is to study MAS from the system point of view of CIM. In the following, we will describe the shop-floor control and management system functions and components.

Shop-floor control and management system is a computer software system that is used to manage and control the operations of MAS. It is generally composed of several modules as shown in Fig.1.10. It receives production plan from MRPII (ERP) system weekly. It optimizes the sequence of jobs using production planning and scheduling algorithms, assigns jobs to specific devices and manufacturing groups, controls the operation of material handling system, and monitors the operations of manufacturing process.

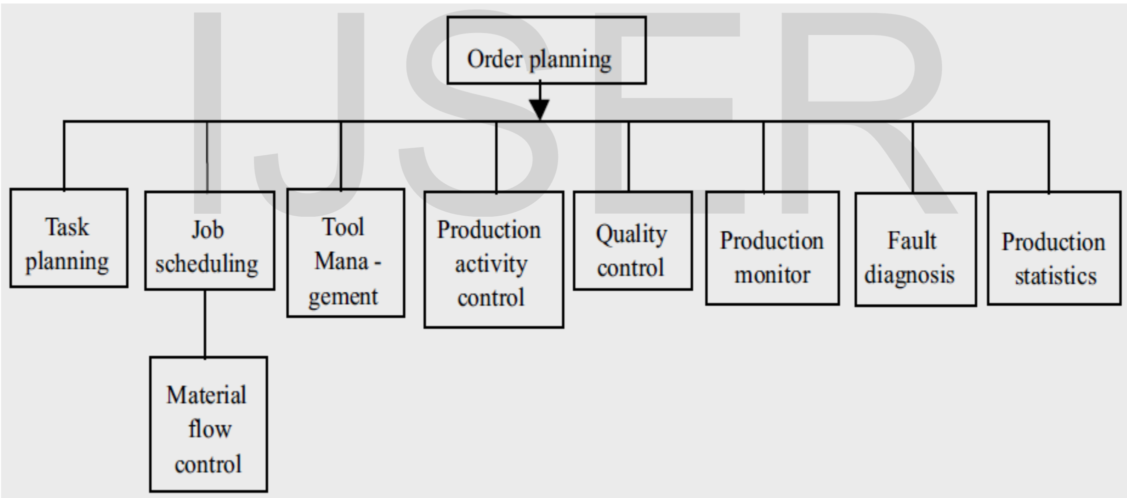


**Fig.1.10: Function Modules of Shop-floor Control and Management System**

Task planning decomposes the order plan from MRPII system into daily task. It assigns job to specific work group and a set of machines according to needed operations. Group technology and optimization technology are used to smooth the production process, better utilize the resources, reduce production setup time, and balance the load for manufacturing devices. Hence good task planning is basis for improving productivity and reducing cost of production.

Job scheduling is used to determine the entry time and sequence for different production jobs. It consists of three main functions: static scheduling, dynamic scheduling, and real time resource scheduling. Material flow control is one of the tasks for real time resource scheduling. Static scheduling is an off line scheduling method; it determines operation sequences before the production starts. The aim of static schedule is to reduce the make span (the time duration is between when the first task enters the system and when the last task leaves the system). Operation research is a major method in generating static scheduling. Since there may be errors and uncertainties caused by machine breakdown, task priorities change, dynamic scheduling is needed to rescheduling the operation sequences and production routes. It is the best method to increase the flexibility of production system. Heuristic rules are normally used in generating dynamic scheduling. Job scheduling aims to optimize the operation of production system and to increase the system flexibility.

Production activity control is used to control the operations of tasks, material flow and manufacturing resources. Real time data collecting, processing, and decision making are major tasks of production activity control. It aims to regulate and smooth the production processes even when some errors and disturbances are occurred.

Tool management is also a very important task for shop-floor control and management system. In a manufacturing system, there are a large number of tools needed; the supply of necessary tools on time has vital importance in improving productivity. The quality of tool is important to the product quality. The parameters of every tool should be maintained in a correct and real time fashion, because these parameters will be used by machine centers in controlling the manufacturing processes.



Quality control, production monitor, fault diagnosis, and production statistics are important supplementary functions for the shop-floor control and management system to be operated efficiently and effectively.

### *1.3.2.4 **Computer Aided Quality Management System***

Since 1970s, quality has become a vitally important factor for a company to win market competition. The customers always want higher product quality for their investment. The computer aided quality management system of CIMS is a system used to guarantee the product quality. It covers a wider range from product design, material supply, to production quality control. The international standard organization (ISO) has established a series of quality insurance standards, such as ISO9000, 9001, 9002, 9003, and 9004. Someone also calls the computer aided quality management system the integrated quality system.

The computer aided quality system consists of four components: quality planning, inspection and quality data collection, quality assessment and control, and integrated quality management.

The quality planning system completes two kinds of functions: computer aided product quality planning and inspection plan generating. According to the historical quality situation, production technology status, the computer aided product quality planning first determines the quality aims, assigns responsibility and resources to every step. Then it determines the associated procedure, method, instruction file, and quality inspection method, generates quality handbook. The computer aided inspection planning determines inspection procedures and standards according to the quality aims, product model, and inspection devices. It also generates automatic inspection programs for automatic inspection devices, such as 3-dimension measuring machine.

Under the guidance of quality plan, the computer aided quality inspection and quality data collection gets quality data during different phases. The phases include purchased material and part quality inspection, part production quality data collection, and final assembly quality inspection. The methods and techniques used in the quality inspection and data collection are discussed in special books regarding quality control (Taguchi, Elsayed, and Hsiang, 1990).

Quality assessment and control fulfils the tasks of manufacturing process quality assessment and control, supply part and supplier quality assessment and control. Integrated quality management includes the functions of quality cost analysis and control, inspection device management, quality index statistics and analysis, quality decision making, tool and fixture management, quality personal management, and product using quality problem feedback information store and quality problem back track into manufacturing steps.

Quality cost has an important role in a company's operation. The quality cost analysis needs to determine the cost bearer and the cost consume point, to generate quality cost plan and calculate real cost. It also optimizes the cost in the effort to solve quality problem. Fig.1.11 presents the quality cost analysis flow chart.

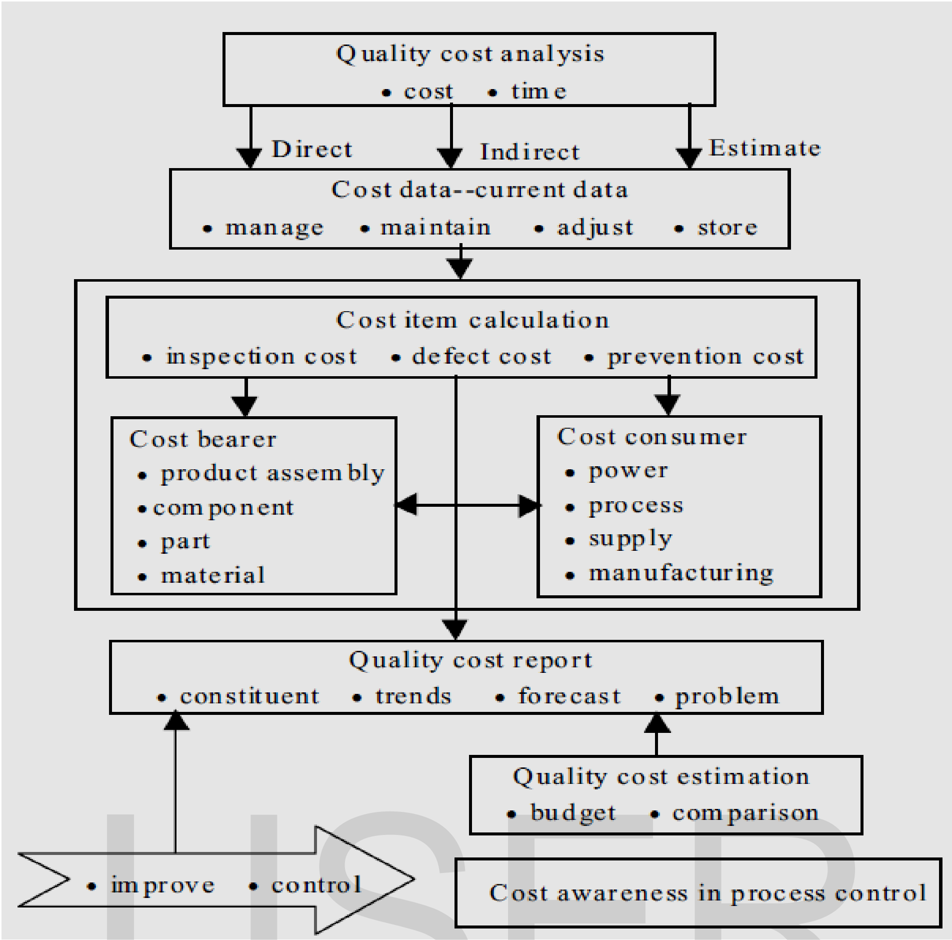


**Fig.1.11: Quality Cost Analysis Flow Chart**



### *1.3.2.5 <u>Computer Network and Database Management Systems</u>*

Computer network and database management systems are supporting systems for CIMS. The computer network consists of a number of computers (called nodes in the network), network devices, and network software. It is used to connect different computers together, so as to enable the data communications between different computers. The computer network can be classified as local area network (LAN) and wide area network (WAN). LAN is normally refers to a restricted area network, such as in a building, in a factory, or in a campus. WAN is referred to a much wider area network; it may be across a city, or in international area. Today, network technology is being rapidly developed. The introduction of Internet concept has changed the manufacturing company's operation method greatly. Global manufacturing, agile manufacturing, and network based manufacturing paradigms have been under rapid development. Computer network is the infrastructure for these new manufacturing paradigms can be realized in a cost effectively way.

Database management system provides a basic support for the data store and information sharing of manufacturing company. Currently relational database management systems are the major databases used. The information integration of a company is concerned with integration data sources in different locations and with different kinds of database management systems. The heterogeneous properties of computer operating systems and database management systems are the major difficulties in the information integration. Now some advanced software techniques have been developed to cope with the heterogeneity problem. One technique is CORBA. Other techniques are OLE/DCOM developed by Microsoft Company, Java language developed by SUN Computer Company.

There are hundreds of books discussing computer network and database techniques. Readers can find them in almost any bookstore.

**1.4. Flexible Manufacturing Systems**

Flexible Manufacturing System (FMS) is a manufacturing system with high degree of flexibility. Flexible Manufacturing System (FMS) is an automated manufacturing system which consists of group of automated machine tools, interconnected with an automated material handling and storage system and controlled by computer to produce products according to the right schedule. Manufacturing scheduling theory is concerned with the right allocation of machines to operations over time. FMS scheduling is an activity to select the right future operational program or diagram of an actual time plan for allocating competitive different demands of different products, delivery dates, by sequencing through different machines, operations, and routings for the combination of the high flexibility of Jobshop type with high productivity of flow-shop type and meeting delivery dates.



FMS Scheduling system is one of the most important information-processing subsystems of CIM system. The productivity of CIM is highly depending upon the quality of FMS scheduling. The basic work of scheduler is to design an optimal FMS schedule according to a certain measure of performance, or scheduling criterion. This work focuses on productivity oriented-make span criteria. Make span is the time length from the starting of the first operation of the first demand to the finishing of the last operation of the last demand.

It is developed due to the need to increase the productivity, improve product quality, and reduce cost for product production under the constraints of various uncertainties or disturbances both internal in and external to the manufacturing system.

The inherent efficiency of a flexible manufacturing system (FMS) combined with additional capabilities, can be harnessed by developing a suitable production plan. Machine scheduling problems arises in diverse areas such as flexible manufacturing system, production planning, computer design, logistics, communication etc. A common feature of many of these problems is that no efficient solution algorithm is known yet for solving it to optimality in polynomial time.

### *1.4.1 Flexibility and Components of FMS*

#### *1.4.1.1 Flexibility of Manufacturing System*

A number of papers have been published to study FMS from different aspects. Gupta and Goyal provide a comprehensive review of the literature on flexibility (Gupta and Goyal, 1989). Flexibility can be defined as a collection of properties of a manufacturing system that supports changes in production activities or capabilities (Carter, l986).

In a manufacturing system, various types of flexibility are needed to fulfill different requirements. Types of flexibility that are mostly discussed are machine flexibility, routing flexibility, process flexibility, product flexibility, production flexibility, and expansion flexibility. Machine flexibility refers to the capability of a machine to perform a variety of operations on a variety of part types and sizes. Machine flexibility can reduce the changeover frequency, setup time, and tool changing time, hence reduce the lead time and make the small lot size production more economic. Machine flexibility is the basis for routing and process flexibility.

Routing flexibility provides the chances for a part to be manufactured or assembled along alternative routes. Routing flexibility is required to manage shop floor uncertainties caused by machine breakdown, tool error, controller failures, and others. It can also be used to tackle the problems caused by external events, such as the change of product mix, product due date, or emergency product introduction. These changes alter machine workloads and cause bottlenecks, the use of alternative routing helps to solve these problems and finally increase productivity.

Process flexibility, also called mix flexibility, refers to the ability to absorb changes in the product mix by performing similar operations or producing similar produces or parts on multipurpose, adaptable, CNC machining centers. Product flexibility, also known as mix change flexibility, is referred to the ability to change over to a new set of products economically and quickly in response to markets or engineering changes or even to operate on a market-to-order basis. In the current global market, high product flexibility is a very important factor for a company to win the competition.

Expansion flexibility is referred to the ability to change a manufacturing system with a view to accommodating a changed product envelope. It is even more important in the current agile manufacturing era; improving expansion flexibility can significantly reduce system expansion or change cost, shorten system reconfiguration time, and hence shorten the delivery time for new products.

### *1.4.1.2 FMS Definition and Components*

An FMS is an automated, mid-volume, mid-variety, central computer-controlled manufacturing system. It can be used to produce a variety of products with virtually no time lost for changeover from one product to the next. Sometimes FMS can be defined ***as "a set of machines in which parts are automatically transported under computer control from one machine to another for processing" (Jha, 1991).***

A more formal definition about FMS is ***"a flexible manufacturing system" consists of a group of programmable production machines integrated with automated material handling equipment and under the direction of a central controller to produce a variety of parts at non-uniform production rate, batch sizes, and quantities" (Jha, 1991).***

From the above definition, it can be seen that an FMS is composed of automated machines, material handling systems, and control systems. In general, the components of an FMS can be classified as follows:

1) *Automated manufacturing devices:* include machining centers with automatic tool interchange ability, measuring machines, and washing machines. They can perform multiple functions according to the NC instructions, thus fulfill parts fabrication task with a great flexibility. In an FMS, the number of automated machining centers is normally greater than or at least equal to 2.
2) *Automated material handling system:* includes load-unload station, high-bay storage, buffers, robot, and material transfer devices. The material transfer devices can be automatic guided vehicle, transfer line, robots, or the combination of these devices. The automated material handling systems are used to prepare, store, and transfer materials (raw materials. unfinished parts. and finished parts) between different machining centers, load-unload stations, buffers, and high-bay storage.
3) *Automated tool system:* it is composed of tool setup devices, central tool storage, tool management systems, and tool transfer systems. They are used to prepare tools for the machining centers as well as transfer tools between machining centers and the central tool storage.
4) *Computer control system:* it is composed of computers and control software. The control software fulfils the functions of task planning, job scheduling, job monitoring, and machine controlling of the FMS.

### 1.4.2 General FMS Considerations

Although FMS was originally developed for metal-cutting applications, the principles of FMS are more widely applicable. Now, it covers a wide spectrum of manufacturing activities such as machining, sheet metal working, welding, fabricating, and assembly.

The research areas regarding the design, implementation, and operation of an FMS are very broad, they have attracted many researchers' interest, and many results have been obtained. In this section, we just present the research topics, problems to be solved, and methods that can be used in solving the problems.

### *1.4.2.1 FMS Design*

FMS is a capital-investment intensive and complex system. In order to get the best economic benefits, the design of FMS should be carefully made. The design decisions regarding to the FMS implementation are the system configuration and layout, manufacturing devices, material handling system, central tool storage, buffers and high-bay storage.

Before these decisions can be made, the part types to be made, the processes needed to make them, and the possible numbers of processing parts (workload) should be first determined. Based on these basic requirements, the number of machines, its abilities, tools, buffers, and storage system can be roughly determined. A rough system layout and material handling system can be designed. The designed FMS is simulated using FMS simulation tool to test its ability to fulfill the requirements.

The design of an FMS is a system approach. Besides the above mentioned basic requirements to meet the part manufacturing ability. There are many other factors that should be considered when designing an FMS. The economic assessment should always be done for every FMS plan obtained. System reliability, productivity, and performance evaluation should be done also for every FMS plan. So, the design of FMS is an iterative process that needs many experts from different disciplines work together. Many alternative plans are compared and modified before an optimized plan is determined.

In the research of FMS design methodology, Talavage and Hannam (l988) summarize other persons work and present a five steps approach to F MS design. The five steps are: (1) Development of goals; (2) Establishment of criteria on which goal achievement can be judged; (3) Development of alternate candidate solutions; (4) Ranking of alternatives by applying the criteria to the alternate solutions; (5) Iteration of the above four steps to obtain a deeper analysis of alternate solutions and to converge on an acceptable solution.

The detailed descriptions about the five steps are described in Talavage and Hannam (1988). The other considerations regarding FMS design can be found in Tetzlaff (1990).

### *1.4.2.2 FMS Planning, Scheduling, and Control*

Planning, scheduling, and control are the important and difficult problems in FMS operations. A good planning and scheduling system will improve the FMS operation efficiency, and get high economic benefits, the research and development of FMS planning and scheduling have been done extensively. The general optimization indexes are: (1) Maximizing the productivity at certain period of time; (2) Minimizing the make span for a group of parts; (3) Minimizing the cost for parts manufacturing; (4) Maximizing the utility for key manufacturing devices; (5) Minimizing the work in progress; (6) Minimizing the production time for certain parts; (7) Satisfying the due dates of parts.

Fig.1.12 presents the function model for FMS planning, scheduling, and resource management.

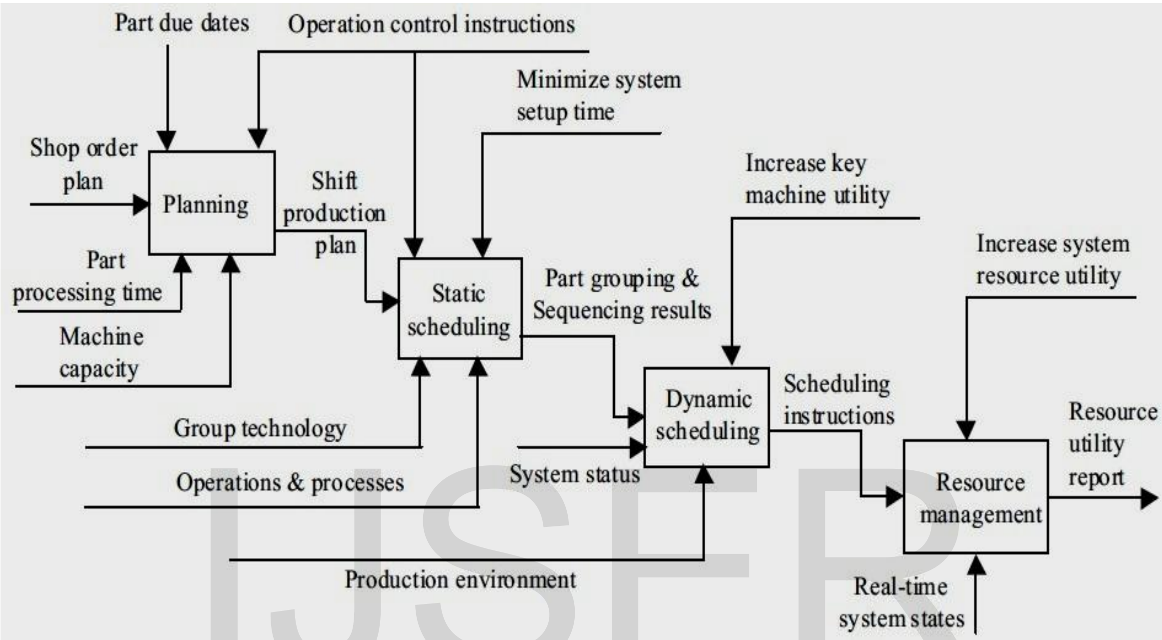


**Fig.1.12: Function Model of FMS Planning and Scheduling**

The resource management and real-time control functions of FMS are closely related to dynamic scheduling system. The resource management system should be activated by a dynamic scheduling system to allocate resources to production process to achieve a real-time control for FMS. The resources to be controlled involve tools, automatic guided vehicle, pallets and fixtures, NC files, and human resources.

- *Planning*

  Planning seeks to find best production plan for the parts entered into the FMS. Its aim is to make an optimized shift production plan according to the shop order and part due dates. FMS planning system receives shop order plan in the weekly time scale from MRPII system, according to the product due dates, it analyzes the shop order, and generates daily or shift production plan. Group technology is used in grouping parts into families of parts. For every shift plan generated, its capacity requirement is calculated, and capacity balance and adjustment work should be carried out if the required capacity is higher than that provided by machines.

After feasibility analysis, capacity balancing, and optimization, a shift plan is then generated. The shift plan gives detailed information for the following questions: (1) what kind of parts will be machined? (2) In what sequence will be the parts entering the FMS? (3) What operations are needed to process the parts? What is the operation sequence? (4) What are the start time and complete time for processed parts? (5) What materials are needed? In what time? (6) What kinds of tool are needed?

- Static Scheduling

Static scheduling is the refinement of shift production plan. it seeks to optimize the machine utility and reduce system setup time. Three functions are performed by a static scheduling system. The three functions are part grouping, workload allocating and balancing, and part static sequencing. Since all these functions are performed before the production starts, the static scheduling is also called off-line sequencing.

A number of factors affecting, production sequence should be taken into account for static scheduling. For example, the part process property, FMS structure, and optimization index. The part process property determines what kind of production method should be used. Flow-shop, flexible-flow-line, and job-shop are three major forms to produce parts. There are different methods that can be used in generating static scheduling for the different production forms.



The second factor affecting static scheduling is FMS structure. The main structure properties are whether there is a central tool system, whether there is a fixture system. or whether there are bottleneck devices. The third factor is the optimization index chosen. The general optimization index is a combination of several optimization indexes. i.e., the FMS static scheduling is a multi-objectives optimization process.

The following parameters have important affection in getting an optimal static scheduling.

1) *Time distribution:* such as time distributions for part arrival, tool setup, part fixture, part transfer, machine failure, and delivery time;
2) *Shop conditions:* such as device type, transfer system, storage method, shop layout, and device condition;
3) *Shop control conventions:* such as priority rule, operation method, hybrid processing route, task decomposition, performance evaluation method, and workload;
4) *Alternate processing route:* such as alternate processing device, alternate processing routing, and alternate processing sequence.

- Dynamic Scheduling

  Dynamic scheduling is used to control the operations of FMS according to the real-time status of the AFMS. It is a real-time (on-line) system that focuses on solving the uncertainty problems such as device failures, bottlenecks on certain machines, workload unbalance, and resource allocation conflict. These problems are not anticipated by off-line static scheduling, they can only be solved using real-time dynamic scheduling or re-scheduling.

  Three strategies can be used to complete the re-scheduling functions. The first one is a periodical scheduling. In order to make a periodical scheduling, it is needed to set a certain time interval as production cycle. A periodical scheduling system calculates a period operation sequence before the next period starts. The calculated sequence is the job list execution instructions followed by the FMS. The second strategy is continuous scheduling that monitors the FMS and executes scheduling whenever an event (such as an new part arrives. or a machine completes the production of a part) has happened and system states has been changed. Since the calculation of work content is effective to re-schedule the FMS operations for every event (so as to get optimal scheduling at every point), the third strategy called hybrid scheduling is frequently used. The hybrid strategy combines periodical and continuous scheduling in the way that only when an unexpected event has happened, then the continuous scheduling algorithm is used, otherwise periodical scheduling is executed at a certain interval.



  For the dynamic manufacturing environment with possible disturbances both internal in and external to the FMS, dynamic scheduling seeks to optimize the sequencing for the queue before manufacturing device. Since the dynamic scheduling of an FMS is an NP-hard problem, it is impossible to search and get the optimal solution at a short time, especially for the continuous scheduling that has very high speed requirement; normally a sub-optimal solution is used in real-time FMS operations. A number of heuristic rules are frequently used in getting, the sub-optimal solutions in dynamic scheduling. The heuristic rules that frequently used are:

  1) *RANDOM:* assigns a random priority to every part entering, the queue. selects a part with smallest priority to be processed;
  2) *FIFO (LIFO)*: first-in-first-out (last-in-first-out);
  3) *SPT (LPT):* selects the part that has the smallest (largest) current operation processing time to be processed;
  4) *FOPNR (MOPNR):* selects the part that has the fewest (most) remaining operations to be processed;
  5) *LWKR (MWKR):* selects the part that has the smallest (largest) remaining

processing time to be processed;
6) *DDATE:* selects the part that has the earliest due date to be processed;
7) *SLACK:* selects the part that has the smallest slack time (due date minus remaining processing time) to be processed.

In most cases, several rules will be used in a dynamic scheduling system in getting the satisfied sequencing solution. Besides the rule based scheduling, simulation based and knowledge based scheduling systems are also widely used.

#### *1.4.2.3 FMS Modeling and Simulation*

Modeling and simulation are important topics both for design and operation of FMS. The FMS modeling is the basis for simulation, analysis, planning and scheduling. Since FMS is a typical discrete event dynamic system (DEDS), a number of methods for DEDS modeling and analysis can be used to model an FMS, such as Petri nets, network of queue (Agrawal, 1985), Activity-Cycle-Diagram (Carrie, 1988), etc.

### 1.4.3 Benefits and Limitations of FMS

FMS offers manufacturers more than just a manufacturing system that is flexible. It offers a concept to improve productivity in mid-variety, mid-volume production situations, an entire strategy for changing company operations ranging from internal purchasing and ordering procedures to distribution and marketing. The benefits of FMS can be summarized as follows:

1) Improving manufacturing system flexibility that is the key advantages of FMS;
2) Improving product quality. increasing equipment utility;
3) Reducing equipment cost, work-in-progress, labor cost, and floor space;
4) Shortened lead times and improving market response speed;
5) Financial benefits gained from above advantages.

In Talavage's (1988) book, there is a chapter discussing the economic justification of FMS.

The difficulties with FMS should also be paid much attention. The first difficulty is that FMS is expensive; it normally requires a large sum of capital resources. The small company may not be able to afford the intensive investment; it may even be not financially beneficial if the company does not have much product variety and volume. Second, the design, implementation, and operation of FMS is a quite complex process, it may cause lost of money if any of the work in this process is not well done. Third, the rapid changing market may urge the company to change its product; this change may also have bad impact on the production system, it may cause the large investment on FMS cannot be returned before it comes out of working.

# CHAPTER 2
## NP-HARD PROBLEMS

### 2.1 Introduction

NP-hard (non-deterministic polynomial-time hard), in computational complexity theory, is a class of problems that are, informally, "at least as hard as the hardest problems in NP". A problem H is NP-hard if and only if there is an NP-complete problem L that is polynomial time Turing-reducible to H (i.e., $L \leq_T H$). In other words, L can be solved in polynomial time by an oracle machine with an oracle for H. Informally, we can think of an algorithm that can call such an oracle machine as a subroutine for solving H, and solves L in polynomial time, if the subroutine call takes only one step to compute. NP-hard problems may be of any type: decision problems, search problems, or optimization problems.

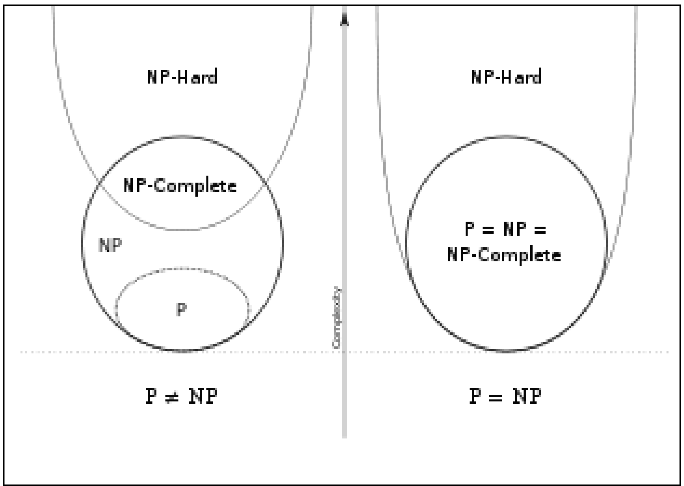


**Fig.2.1: Euler diagram for P, NP, NP-complete, and NP-hard set of problems**



As consequences of definition, we have (note that these are claims, not definitions):

- Problem H is at least as hard as L, because H can be used to solve L;
- Since L is NP-complete, and hence the hardest in class NP, also problem H is at least as hard as NP, but H does not have to be in NP and hence does not have to be a decision problem (even if it is a decision problem, it need not be in NP);
- Since NP-complete problems transform to each other by polynomial-time many-one reduction (also called polynomial transformation), all NP-complete problems can be solved in polynomial time by a reduction to H, thus all problems in NP reduce to H; note, however, that this involves combining two different transformations: from NP complete decision problems to NP-complete problem L by polynomial transformation, and from L to H by polynomial Turing reduction;
- If there is a polynomial algorithm for any NP-hard problem, then there are polynomial algorithms for all problems in NP, and hence P = NP;
- If P ≠ NP, then NP-hard problems have no solutions in polynomial time, while P = NP does not resolve whether the NP-hard problems can be solved in polynomial time;
- If an optimization problem H has an NP-complete decision version L, then H is NP-hard.

A common mistake is to think that the NP in NP-hard stands for non-polynomial. Although it is widely suspected that there are no polynomial-time algorithms for NP-hard problems, this has never been proven. Moreover, the class NP also contains all problems which can be solved in polynomial time.

An example of an NP-hard problem is the decision subset sum problem, which is this: given a set of integers, does any non-empty subset of them add up to zero? That is a decision problem, and happens to be NP-complete. Another example of an NP-hard problem is the optimization problem of finding the least-cost cyclic route through all nodes of a weighted graph. This is commonly known as the travelling salesman problem.

There are decision problems that are NP-hard but not NP-complete, for example the halting problem. This is the problem which asks "given a program and its input, will it run forever?" That's a yes/no question, so this is a decision problem. It is easy to prove that the halting problem is NP-hard but not NP-complete. For example, the Boolean satisfiability problem can be reduced to the halting problem by transforming it to the description of a Turing machine that tries all truth value assignments and when it finds one that satisfies the formula it halts and otherwise it goes into an infinite loop. It is also easy to see that the halting problem is not in NP since all problems in NP are decidable in a finite number of operations, while the halting problem, in general, is undecidable. There are also NP-hard problems that are neither NP-complete nor undecidable. For instance, the language of True quantified Boolean formulas is decidable in polynomial space, but not non-deterministic polynomial time (unless NP = PSPACE).

An alternative definition of NP-hard that is often used restricts NP-hard to decision problems and then uses polynomial-time many-one reduction instead of Turing reduction. So, formally, a language L is NP-hard if $\forall L' \in NP, L' \leq L$. If it is also the case that L is in NP, then L is called NP-complete. However, under this definition, the trivial decision problem (the one that accepts everything) and its complement would probably not be in NP-hard, even if P = NP, since no other problems can many-one reduce to these two problems.

### 2.2 NP-naming Convention

The NP-family naming system is confusing: NP-hard problems are not all NP, despite having NP as the prefix of their class name. However, the names are now entrenched and unlikely to change. On the other hand, the NP-naming system has some deeper sense, because the NP family is defined in relation to the class NP:

- *NP-hard*
  At least as hard as the hardest problems in NP. Such problems need not be in NP; indeed, they may not even be decision problems.
- *NP-complete*

These are the hardest problems in NP. Such a problem is NP-hard and in NP.

- *NP-easy*
  At most as hard as NP, but not necessarily in NP, since they may not be decision problems.
- *NP-equivalent*
  Exactly as difficult as the hardest problems in NP, but not necessarily in NP.

**2.3 Application Areas**

NP-hard problems are often tackled with rules-based languages in areas such as:

- Configuration
- Data mining
- Selection
- Diagnosis
- Process monitoring and control
- Scheduling
- Planning
- Rosters or schedules
- Tutoring systems
- Decision support

# CHAPTER 3
## GANTT CHARTS

### 3.1 Introduction
A Gantt chart is a type of bar chart, developed by Henry Gantt that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Some Gantt charts also show the dependency (i.e., precedence network) relationships between activities.

Although now regarded as a common charting technique, Gantt charts were considered revolutionary when first introduced. In recognition of Henry Gantt's contributions, the Henry Laurence Gantt Medal is awarded for distinguished achievement in management and in community service. This chart is also used in Information Technology to represent data that has been collected.

### 3.2 Historical Development
The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a harmonogram. Adamiecki only published his chart in 1931, however, in Polish, which limited both its adoption and recognition of his authorship. The chart is named after Henry Gantt (1861-1919), who designed his chart around the years 1910-1915.



One of the first major applications of Gantt charts was during World War I. On the initiative of General William Crozier, then Chief of Ordnance these included that of the Emergency Fleet, the Shipping Board, etc.

In the 1980s, personal computers allowed for widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers. With the advent of the Internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative groupware.

### 3.3 Advantages and Limitations
Gantt charts have become a common technique for representing the phases and activities of a project work breakdown structure (WBS), so they can be understood by a wide audience all over the world.

A common error made by those who equate Gantt chart design with project design is that they attempt to define the project work breakdown structure at the same time that they define schedule activities. This practice makes it very difficult to follow the 100% Rule. Instead the

WBS should be fully defined to follow the 100% Rule, and then the project schedule can be designed.

Although a Gantt chart is useful and valuable for small projects that fit on a single sheet or screen, they can become quite unwieldy for projects with more than about 30 activities. Larger Gantt charts may not be suitable for most computer displays. A related criticism is that Gantt charts communicate relatively little information per unit area of display. That is, projects are often considerably more complex than can be communicated effectively with a Gantt chart.

Gantt charts only represent part of the triple constraints (cost, time and scope) on projects, because they focus primarily on schedule management. Moreover, Gantt charts do not represent the size of a project or the relative size of work elements, therefore the magnitude of a behind-schedule condition is easily miscommunicated. If two projects are the same number of days behind schedule, the larger project has a larger effect on resource utilization, yet the Gantt does not represent this difference.

Although project management software can show schedule dependencies as lines between activities, displaying a large number of dependencies may result in a cluttered or unreadable chart.



Because the horizontal bars of a Gantt chart have a fixed height, they can misrepresent the time-phased workload (resource requirements) of a project, which may cause confusion especially in large projects. In practice, many activities (especially summary elements) have front-loaded or back-loaded work plans, so a Gantt chart with percent-complete shading may actually miscommunicated the true schedule performance status.

# UNIT - II

# CHAPTER 1
## SEQUENCING AND SCHEDULING

***Manufacturing scheduling plays a very important function in successful operation of the production planning and control department of an organization. It also offers a great theoretical challenge to the researchers because of its combinatorial nature. Earlier, researchers emphasized classical optimization methods such as linear programming and branch-and-bound method to solve scheduling problems. However, these methods have the limitation of tackling only small-sized scheduling problems because of the consumption of high computational (CPU) time. As a result, heuristics as well as various efficient optimization methods based on the evolutionary computing paradigm such as genetic algorithms, simulated annealing, and artificial immune system have been applied to scheduling problems for obtaining near optimal solutions. These computational tools are currently being utilized successfully in various engineering and management fields. We briefly discuss the overview of these emerging heuristics and metaheuristics and their applications to the scheduling problems. Given the rise in attention by the researchers, more emphasis has been given to explore artificial immune system in details.***

### 1.1 Sequencing and Scheduling

*Sequencing* is a technique to order the jobs in a particular sequence. There are different types of sequencing which are followed in industries such as first in first out basis, priority basis, job size basis and processing time basis etc. In processing time basis sequencing for different sequence, we will achieve different processing time. The sequence is adapted which gives minimum processing time.

*Scheduling* is concerned with the assignment of time to a set of jobs for processing through a group of machines (or their service sector equivalents) in order to best satisfy some criteria. A great deal of research has been carried out and will continue to be done on manufacturing scheduling problems (Baker, 1974). The reason is that scheduling offers a great theoretical challenge for researchers because of its combinatorial nature. Also, from the practical point of view, it plays a significant Heuristics and Metaheuristics for Solving Scheduling Problems role in the successful operation of production, planning, and control department.

The general flowshop scheduling problem is known to be nondeterministic polynomial (NP) complete (Gonzalez & Sahni, 1978). For solving scheduling problems, simple exact analytical methods such as integer programming (Sriker & Ghosh, 1986) or branch-and-bound (Lomnicki, 1965) have the limitation of dealing with only small-sized problems because of large computational effort. Heuristic polynomial-time algorithms (Campbell Dudek, & Smith (CDS), 1970; Johnson, 1954; Nawaz, Enscore, & Ham, 1983) probably are the most suitable means to solve large scheduling problems that are frequently encountered in many real-world situations. In general, heuristics provide good satisfactory (but not necessarily optimal) solutions in reasonable time and use problem-specific information.

The classical flowshop scheduling problem is one of the most well known scheduling problems. Informally the problem can be described as follows:
There are set of jobs and a set of machines. Each job consists of chain of operation, each of which needs to be processed during an uninterrupted time period of a given length on a given machine. Each machine can process at most one operation at a time. A schedule is an allocation of operations to time intervals of the machines. The problem is to find the schedule of minimum length. This work tries to minimize the makespan of batch-processing machines in a flowshop. The processing times and the sizes of the jobs are known and non-identical. The machines can process a batch as long as its capacity is not exceeded. The processing time of a batch is the longest processing time among all the jobs in that batch. The problem under study is NP-hard for the makespan objective. Consequently, comparison based on Gupta's heuristics, RA heuristic's, Palmer's heuristics, CDS heuristics are proposed. Gantt chart was generated to verify the effectiveness of the proposed approaches.

The problems of manufacturing scheduling (Sarin & Lefoka, 1993) may be segregated based on (1) requirements, (2) complexity of the processes, and (3) scheduling objectives. Requirements may be produced either by open shop (customer orders) or closed shop (inventory replenishment). The complexity of the processes is primarily determined by the order in which the different machines appear in the operations of individual jobs. Broadly, manufacturing scheduling can be classified as flowshop scheduling and jobshop scheduling. In flowshop scheduling, it is generally assumed that all jobs must be processed on all machines in the same technological or machine order. In jobshop scheduling, the jobs may be processed following different machine orders. There is no common path of movement of jobs from machine to machine. Each machine is likely to appear for processing each operation of each job. The scheduling objectives are evaluated to determine the optimum schedule of jobs. Some of the objectives include makespan, total flow time, average job tardiness, and number of tardy jobs.

A variety of scheduling problems has been developed over the past years to address different production systems. The two commonly scheduling problems found in the scheduling literature of the past 50 years are flowshop scheduling and jobshop scheduling. Scheduling problems may be deterministic/stochastic and static/dynamic (Simons, 1992). The problem is deterministic or stochastic when the time required to process a task over respective machine takes a fixed or a random value. The scheduling problem is considered as static if ordering of jobs on each machine is determined once and will remain unchanged as opposed to the dynamic case that can accommodate changes of job ordering for accessing new jobs to the system.

## 1.2 Objectives of Scheduling

Most scheduling research has considered optimizing a single objective. The different performance measures or objectives include makespan, total flowtime, and job tardiness. Makespan of a schedule of jobs is the completion time of the last job in that schedule (it is assumed that the schedule starts at zero time). The total flow time of a schedule of jobs is the sum of completion times of all jobs in that schedule. Job tardiness indicates the lateness of the job with respect to its due date. Minimization of makespan results in maximization of overall resource utilization, whereas total flow time aims at minimizing work-in-process inventory and minimum tardiness yields minimum penalty.

A number of assumptions for flowshop or jobshop scheduling are considered. They are primarily considered for simplicity of the structure of the problems, but at the same time they help to build the generalized model. Most of the different applications using these models require relaxing one or several of these assumptions, so that they are not entirely realistic models for the applications. Some of the assumptions include availability of jobs, noninterference of machines, non-passing of jobs, and so forth. Dudek and Teuton (1964) provide a complete list of these assumptions in their paper.

## 1.3 Types of Scheduling

Basically there are three types of scheduling:

### *1.3.1 Single Machine Scheduling:*

Here we arrange the order of jobs in a particular machine. We achieve the best result when the jobs are arranged in the ascending order of their processing time i.e. the job having least processing time is put first in sequence and processed through the machine and the job having maximum processing time is put last in sequence.

### *1.3.2 Flowshop Scheduling:*

It is a typical combinatorial optimization problem, where each job has to go through the processing in each and every machine on the shop floor. Each machine has same sequence of jobs. The jobs have different processing time for different machines. So in this case we arrange the jobs in a particular order and get many combinations and we choose that combination where we get the minimum makespan.

### *1.3.3 Jobshop Scheduling:*

It is also a typical combinatorial optimization problem, but the difference is that, here all the jobs may or may not get processed in all the machines in the shop floor i.e. a job may be processed in only one or two machines or a different job may have to go through the processing in all the machine in order to get completed. Each machine has different sequence of jobs. So it is a complex web structure and here also we choose that combination of arrangements that will be giving the least makespan.

### 1.4 Significance of Work

Establishing the timing of the use of equipment, facilities and human activities in an organization can:

1) Determine the order in which jobs at a work center will be processed.
2) Results in an ordered list of jobs
3) Sequencing is most beneficial when we have constrained capacity (fixed machine set; cannot buy more) and heavily loaded work centers
4) Lightly loaded work centers = no big deal (excess capacity)
5) Heavily loaded
    a) Want to make the best use of available capacity.
    b) Want to minimize unused time at each machine as much as possible.

### 1.5 Parameters of the Work

1) Average job flow time
    a) Length of time (from arrival to completion) a job is in the system, on average
    b) Lateness
       Average length of time the job will be late (that is, exceeds the due date by)
    c) Makespan
    d) Total time to complete all jobs
    e) Average number of jobs in the system
    f) Measure relating to work in process inventory
    g) Equals total flow time divided by makespan.

### 1.6 Objective of the Work

1) To deal with the production planning problem of a flexible manufacturing system. I model the problem of a flowshop scheduling with the objective of minimizing the makespan
2) To provide a schedule for each job and each machine. Schedule provides the order in which jobs are to be done and it projects start time of each job at each work center.
3) To select appropriate heuristics approach for the scheduling problem through a comparative study.
4) To solve FMS scheduling problem in a flow-shop environment considering the comparison based on Gupta’s heuristics, RA heuristic’s, Palmer’s heuristics, CDS heuristics are proposed. Gantt chart was generated to verify the effectiveness of the proposed approaches.

My objective of scheduling can yield:

1) Efficient utilization …
    a) Staff

   b) Equipment
   c) Facilities
2) Minimization of …
   a) Customer waiting time
   b) Inventories
   c) Processing time

**1.7 Organization of the thesis**

The section of thesis (i.e., Unit II) describing the present research work covers six chapters.

*Chapter-II* describes several diverse streams of literature on the optimization of various scheduling problems regarding single machining, flowshop and Jobshop.

*Chapter-III* describes the optimization methods of scheduling for the efficient utilization of staff, equipment, facilities utilization with the objective of minimizing customer waiting time, Inventories, Processing time.

*Chapter-IV* describes various proposed methods of optimization for flowshop scheduling problem.

*Chapter-V* deals with the results and discussion for the problem.

*Chapter-VI* concludes with implication and a suggestion for the extension of the study.



**1.8 Summary**

Sequencing is a technique to order the jobs in a particular sequence. There are different types of sequencing which are followed in industries such as first in first out basis, priority basis, job size basis and processing time basis etc. In processing time basis sequencing for different sequence, we will achieve different processing time. The sequence is adapted which gives minimum processing time.

By Scheduling, we assign a particular time for completing a particular job. The main objective of scheduling is to arrive at a position where we will get minimum processing time. The various assignments associated with staff, equipment, facilities utilization and customer waiting time, Inventories, Processing time minimization are:

1) Need to assign a job to a machine/resource to process it.
2) Loading.
3) Need to decide how many jobs can be assigned to each machine.
4) Scheduling.
5) Need to decide on a starting time for each job at each workstation.
6) Sequencing.
7) Need to order processing of individual jobs at each workstation.

The work associated with staff, equipment, facilities utilization and customer waiting time, inventories, processing time minimization are presented.

# CHAPTER 2
## LITERATURE SURVEY

### 2.1 Introduction

Another important consideration is the choice of appropriate criteria for scheduling. Although the ultimate objective of any enterprise is to maximize the net present value of the shareholder wealth, this criterion does not easily lend itself to operational decision-making in scheduling. Some researchers are developing methodologies which take revenue and cost effects of schedules into consideration. Researchers and practitioners have so far used operational surrogates that influence costs and revenues. These include: number of parts tardy, average tardiness, weighted tardiness, throughput (this is a revenue influencing surrogate), as well as average number of parts in the system, machine utilization, and work in- process inventory, for example. Analyses of these surrogates indicate that a scheduling procedure which does well for one criterion is not necessarily the best for some other. For example, attempts to reduce mean tardiness can lead to an increase in mean flow time. Minimizing makespan can result in higher mean flow time.

### 2.2 Previous Literatures



Ecker and Gupta [1] developed optimization models for scheduling tasks on a flexible manufacturing machine to minimize tool change delays. They considered the problem of scheduling a given set of precedence constraint tasks on a flexible machine equipped with a tool magazine where each task requires exactly one of the tools during its execution changing from one tool to another requires a certain amount of time that depends on the pair of tools being exchanged. They present a new algorithmic approach for general task precedence relations when it is desired to sequence the tasks in such a way that the total time required for tool changes is minimized. The approaches they used are

a) A heuristic algorithm
b) Simulation.

Lia et.al [2] developed efficient composite heuristics models for total flow time minimization in permutation flowshops considering flowshops with total flow time. Flowshop scheduling is an important manufacturing system widely existing in industrial environments. A flowshop can be described as *n* jobs being processed on *m* machines and each job having the same machine order .Thus they proposed a composite heuristics model to minimize the flow time.

Felix T.S. Chan et.al [3] developed optimization models for solving distributed FMS scheduling problems subject to maintenance: [Genetic algorithms approach]. The authors have made an attempt to optimize the following things during the cycle in the work:

a) Allocation of jobs to suitable factories
b) Determination of the corresponding production scheduling in each factory.

Their objective is to maximize the system efficiency by finding an optimal planning for a better collaboration among various processes. They proposed a genetic algorithm with dominant genes (GADG) approach to deal with distributed flexible manufacturing system (FMS) scheduling problems subject to machine maintenance constraint.

Kedad Sidhoum et.al [4] developed optimization models for lower bounds for the earliness tardiness scheduling problem on parallel machines with distinct due dates, considering the parallel machine scheduling problem in which the jobs have distinct due dates with earliness and tardiness. They considered the earliness–tardiness problem in a parallel machine environment. Their objective is related with the parallel machine scheduling problem in which the jobs have distinct due dates with earliness and tardiness costs. The main objective of their model is to optimize tardiness.

Sharafali et.al [5] developed optimization models for production scheduling in a flexible manufacturing system under random demand. They considered the problem of production scheduling in a Flexible Manufacturing System (FMS) with stochastic demand.

Drstvensek et.al [6] developed a model of data flow in lower CIM levels considering, models of production automation based on the idea of five levels CIM hierarchy where the technological database (TDB) represents a backbone of the system . Their main objective is to provide a common environment where the evaluation of a given general order and later composition of work orders, and designation of production resources could be done automatically under the operator's supervision.



Ezedeen Kodeekha (Department of Production, Informatics, Management and Control Faculty of Mechanical Engineering Budapest University of Technology and Economics) [7] developed "A new method of FMS scheduling using optimization and scheduling". Conventional methods of solving scheduling problems such as heuristic methods based on priority rules still result schedules, sometimes, with significant idle times. To optimize these, the author proposes a new high quality scheduling method. He uses multi-objective optimization and simulation method .The method is called "Break and Build Method", BBM.

Clarence H Martin [8] developed a hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming. He developed a new aspect of the problem related with sub lots, the size of sub lots and the interleaving of sub lots from different jobs in the processing sequence. His approach allows for quicker movement of items through the manufacturing facility that is a key element of synchronous manufacturing. Of course, lot streaming raises new issues such as determining the number of sub lots and their sizes.

Chia and Lee [9] developed the total completion time problem in a permutation flowshop with a learning effect. The concept of learning process plays a key role in production environments. Their objective is to minimize the sum of completion times or flow time. They used the dominance rule and several lower bounds to speed up the search for the optimal solution.

Hamania et.al, [10] developed a model for Reactive mode handling of flexible manufacturing systems. They deal with a new Modelling approach for mode handling of flexible manufacturing systems (FMS). Based on a review of the Modelling methods and the specification formalisms in the existing approaches, they show that the mutual benefit of functional Modelling and synchronous languages is very convenient for mode handling problem.

Hsu, et.al. [11] developed a model for cyclic scheduling for FMS: Modelling and evolutionary solving approach. They concern the domain of flexible manufacturing systems (FMS) and focuses on the scheduling problems encountered in these systems. They have chosen the cyclic behavior to study this problem with the objective to reduce its complexity. This cyclic scheduling problem, whose complexity is NP-hard in the general case, aims to minimize the work in process (WIP) to satisfy economic constraints. They study the problem of FMS control by a predictive approach to compute a cyclic and deterministic schedule.



Sadykov [12] developed a branch-and-check algorithm for minimizing the weighted number of late jobs on a single machine with release dates. He considers the scheduling problem of minimizing the weighted number of late jobs on a single machine. He proposed a branch-and check algorithm, where a relaxed integer programming formulation is solved by branch-and-bound and infeasible solutions are cut off using infeasibility cuts.

Koulamas and Kyparisis [13] developed single-machine scheduling with waiting-time dependent due dates in which due dates are linear functions of the job waiting-times. They construct an optimal sequence and assign the optimal due dates analytically in a single machine setting when due dates are linear functions of the job waiting-times and their objective is to minimize the maximum job lateness.

Das, et.al [14] developed, Optimization of operation and changeover time for production planning and scheduling in a flexible manufacturing system and deals with the production planning problem of a flexible manufacturing system. They specifically addresses issues of machine loading, tool allocation, and part type grouping with the objective of developing an operation sequencing technique capable of optimizing operation time, non-productive tool change times, and orientation change times when processing a group's design features.

Chen and Lee [15] developed a model for Logistics scheduling with batching [LSB] and transportation. Their objective is to minimize the sum of weighted job delivery time and total transportation cost. Since their problem involves not only the traditional performance measurement, such as weighted completion time, but also transportation arrangement and cost, key factors in logistics management.

Poulos and Zografos [16] developed a model for solving the multi-criteria time-dependent routing and scheduling problem in a multimodal fixed scheduled network. Their objective is to present the formulation and algorithmic solution for the multi-criteria itinerary planning problem that takes into account the aforementioned features. Their main objectives are:

1) Formulate the itinerary planning problem as a multi-criteria shortest path problem with intermediate stops in a multimodal time dependent network,
2) Present a decomposition scheme for handling the constraint of visiting intermediate locations within specified time windows, and
3) Introduce a dynamic programming based algorithm for solving the individual elementary multi-criteria time-dependent shortest path problems between any pair of sequential intermediate stops.

In addition, they proved that the Basic Unit of Concurrency (BUC) is a set of the executed control flows based on the behavioral properties of the net.

Wu and Zhou [17] developed a model for Stochastic scheduling to minimize expected maximum lateness. They concerned with the problems in scheduling a set of jobs associated with random due dates on a single machine so as to minimize the expected maximum lateness in stochastic environment. This is a difficult problem and few efforts have been reported on their solution.

Rossi and Dini [18] developed Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimization method. They propose an ant colony optimization-based software system. Their main objective is to solve FMS scheduling problem in a job-shop environment considering routing flexibility, sequence-dependent setup and transportation time. Routing flexibility leads to the problem of flexible (or multiprocessor) job-shop scheduling (FJS) which extends the classic problem of job-shop scheduling where no alternative machine is present for processing an operation. They concerns two sub-problems:

1) Assignment of each operation to one of the alternative machines (assignment sub-problem);
2) Ordering of the operations on each assigned machine (sequencing sub problem), with the aim of optimizing them.

Wang et.al [19] developed FBS-enhanced agent-based dynamic scheduling in FMS. The main objective is to show the feasibility of the approach and to evaluate the approach via computational experiments. They propose a multi-agent approach integrated with a filtered beam- search (FBS)-based heuristic algorithm to study the dynamic scheduling problem in a FMS shop floor consisting of multiple manufacturing cells.

Tang and Zhao [20] developed a model for scheduling a single semi-continuous batching machine. They address a new problem, called semi-continuous batch scheduling, which arises in the heating-operation of tube billets in the steel industry. Each heating furnace can be regarded as a semi continuous batching machine, which can handle up to *C* jobs simultaneously. Their objectives are to schedule jobs on the machine so that the makespan and the total completion time are minimized.

Eren and Guner [21] developed a bi-criteria flowshop scheduling problem with a learning effect. They consider learning effect in a two-machine flowshop. Their objective is to find a sequence that minimizes a weighted sum of total completion time and makespan.

Wu and Zhou [22] developed a model for Stochastic scheduling to minimize expected maximum lateness. They concerned with the problems in scheduling a set of jobs associated with random due dates on a single machine so as to minimize the expected maximum lateness in stochastic environment. This is a difficult problem and few efforts have been reported on their solution.



Yang and Geunes [23] developed a predictive-reactive scheduling model on a single resource with uncertain future jobs. Their objective is to minimize the sum of expected tardiness cost, schedule disruption cost, and wasted idle time cost.

Mosheiov and Sarig [24] developed a model for Due-date assignment on uniform machines. Their objective is to find the job schedule and the due-date that minimize a linear combination of all three (earliness, tardiness and due-date) cost factors.

Biskup and Herrmann [25] developed a model for Single-machine scheduling against due dates with past-sequence-dependent setup times. Their objective is to minimize the due date.

Chena et.al [26] developed a model for dense open-shop schedules with release times. They study open-shop scheduling problems with job release times. Their objective is to minimize the makespan. Dense schedules, easy to construct, are often used as approximate solutions. Performance ratios of the makespans from dense schedules and that of the optimal schedule of the problem are used to evaluate the quality of approximate schedules.

Goncalves, et.al [27] developed a genetic algorithm for the resource constrained multi-project scheduling problem. They presents a genetic algorithm for the resource constrained multi-project scheduling problem. The chromosome representation of the problem is based on random keys. They constructed schedules using a heuristic that builds parameterized active schedules based on priorities, delay times, and release dates defined by the genetic algorithm with the objective to optimize the resource constrained multi-project scheduling problem.

Cheng et.al [28] developed a model for Single-machine scheduling of multi-operation jobs without missing operations to minimize the total completion time. They consider the problem of scheduling multi-operation jobs on a single machine to minimize the total completion time. Each job consists of several operations that belong to different families. In a schedule each family of job operations may be processed as batches with each batch incurring a set-up time. A job is completed when all of its operations have been processed. Their objective is to minimize the total completion time.

Teunter et.al [29] developed a model for Multi-product economic lot scheduling problem with separate production lines for manufacturing and remanufacturing. They study the economic lot scheduling problem with two production sources, manufacturing and re-manufacturing, for which operations are performed on separate, dedicated lines. Their objective is to develop an exact algorithm for finding the optimal common- cycle-time policy. Their algorithm combines a search for the optimal cycle time with a mixed integer programming (MIP) formulation of the problem given a fixed cycle time.



Tang and Gong [30] developed a hybrid two-stage transportation and batch scheduling problem They study the coordinated scheduling problem of hybrid batch production on a single batching machine and two-stage transportation connecting the production, where there is a crane available in the first-stage transportation that transports jobs from the warehouse to the machine and there is a vehicle available in the second-stage transportation to deliver jobs from the machine to the customer. Their objective is to minimize the sum of the makespan and the total setup cost.

Tseng and Liao [31] developed a discrete particle swarm optimization for lot streaming flowshop scheduling problem. They consider an n-job, m-machine lot streaming problem in a flowshop with equal-size sub lots where their objective is to minimize the total weighted earliness and tardiness.

Chang et.al [32] developed a hybrid genetic algorithm with dominance properties for single machine scheduling with dependent penalties. They developed a hybrid genetic algorithm to solve the single machine scheduling problem with the objective to minimize the weighted sum of earliness and tardiness costs.

Cheng and Lin [33] developed Johnson's rule, composite jobs and the relocation problem. They consider two-machine flowshop scheduling with the objective to minimize makespan. Johnson's rule for solving this problem has been widely cited in their work. They introduce the concept of composite job, which is an artificially constructed job with processing times such that it will incur the same amount of idle time on the second machine as that incurred by a chain of jobs in a given processing sequence.

Seong-Jong Joo et.al [34] developed a model for Scheduling preventive maintenance for modular designed components: A dynamic approach. Their objective is to develop a dynamic approach for scheduling preventive maintenance at a depot with the limited availability of spare modules and other constraints. They proposed a backward allocation algorithm and applied it to scheduling the preventive maintenance of an engine module installed in T-59 advanced jet trainers in the Republic of Korea Air Force.

He and Hui [35] developed a rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units. They present a genetic algorithm-based on heuristic rules for large-size SMSP. In their work, the size of the problems was enlarged, and the problems are first solved by MILP methods and then a random search (RS) based on heuristic rules has been proposed.



Chen and Askin [36] developed a model for Project selection, scheduling and resource allocation with time dependent returns. They formulate and analyze the joint problem of project selection and task scheduling. They study the situation where a manager has many alternative projects to pursue such as developing new product platforms or technologies, incremental product upgrades, or continuing education of human resources. Project return is assumed to be a known function of project completion time. Resources are limited and renewable. Their objective is to maximize present worth of profit.

Janiak et.al [37] developed a scheduling problem with job values given as a power function of their completion times. They deal with a problem of scheduling jobs on the identical parallel machines, where job values are given as a power function of the job completion times. Minimization of the total loss of job values is the main objective of their work.

Grzegorz Waligo´ra [38] developed a model named Tabu search for discrete- continuous scheduling problems with heuristic continuous resource allocation. His objective is to minimize the makespan .He considered problems of scheduling non-preempt able, independent jobs on parallel identical machines under an additional continuous renewable resource.

Valls et.al [39] developed skilled workforce scheduling in Service Centers. Their main objective with SWPSP is to quickly obtain a feasible plan of action satisfying maximum established dates and timetable worker constraints. Secondary their objectives deal with the urgency levels imposed by the criticality task levels, to obtain well-balanced worker workloads and an efficient assignment of specialists to tasks.

Tiwari et.al [40] developed a model for scheduling projects with heterogeneous resources to meet time and quality objectives. Their approach guides decision-making concerning which workers to cross-train in order to extract the greatest benefits from worker-flexibility. They demonstrates how the output of the model can be used to identify bottlenecks (or critical resource skills), and also demonstrates how cross-training the appropriately skilled groups or individuals can increase throughput.

P.Y. Fung [41] developed a model for Lower bounds on online deadline scheduling with preemption penalties. He generalizes and improves results of online preemptive deadline scheduling with preemption penalties. He consider both the preemption-restart and the preemption resume models, and give new or improved lower bounds on the competitive ratio of deterministic online algorithms. In many cases his proposed bounds are optimal when the job deadlines are tight.

**2.3 Summary**

Scheduling is an activity to select the right future operational program or diagram of an actual time plan for allocating competitive different demands of different products, delivery dates, by sequencing through different machines, operations, and routings for the combination of the high flexibility of Jobshop type with high productivity of flow-shop type and meeting delivery dates. The different types of approaches to the Manufacturing scheduling theory have been reported. Here these approaches show their various advantages and disadvantages to the development of new design problem. Taking the old approach in to consideration the development of new approaches conceptualized through these literatures.

# CHAPTER 3
## DESCRIPTION OF METHODS

### 3.1 Introduction of Sequencing Methods

The selection of the job must be done on the basis of below mentioned criterions:

1) *Shortest processing time:* select the job having the least processing time
2) *Earliest due date:* select the job that is due the soonest.
3) *First-come, first served:* select the job that has been waiting the longest for this work-station.
4) *First-in-system, first-served:* select the job that has been in the shop the longest.
5) *Slack per remaining operations:* select the job with the smallest ratio of slack to operations remaining to be performed.

### 3.2 Overview of Scheduling Methods on Makespan Criterion

It has been observed that the flowshop as well as jobshop scheduling problems, with few exceptions, belong to the class of combinatorial problems, which are termed as NP-complete for which no efficient polynomial time algorithm is available (Gonzalez & Sahni, 1978). Simple exact analytical methods such as brand-and-bound have been developed by Lomnicki (1965), Brown and Lomnicki (1966), and Bestwick and Hastings (1976). Although it is the best optimizing method available for solving NP-complete scheduling problems, it requires high central processing unit (CPU) time to solve large scheduling problems. So, the heuristic algorithms probably are the only means to solve especially large-sized scheduling problems that are frequently encountered in many real-life situations. These heuristics guarantee good solutions that are satisfactory though they may not be globally optimal.

For the past 50 years, flowshop scheduling has been one of the most important areas in the scheduling literature. The scheduling heuristic approach generally cited as the foundation technique is the one developed by Johnson (1954). He presented a simple, well-known constructive heuristic algorithm to minimize makespan for the n-job, 2-machine scheduling problem. Due to the simplicity of Johnson's algorithm and its guarantee for giving optimal solutions, many researchers were encouraged to extend this idea to the general n-job, m-machine case, but without much success. Since then most of the efforts have been directed at finding optimal solution with m-machine scheduling problems. Ignall and Schrage (1965) developed an optimization algorithm using the branch-and-bound method for three-machine flowshop problems. Efficient heuristics that yield optimal solutions are desirable for generalized n-job, m-machine flowshop problems. Some of the noteworthy heuristics on the makespan criterion have been developed by Palmer (1965); King and Spachis (1980) (called KS); Dannnenbring (1977); Campbell, Dudek, and Smith (1970) (called CDS); Nawaz, Enscore, and Ham (1983) (called NEH); Koulamas (1998); Widmer and Hertz (1989) (called WH); Taillard (1990); Sarin and Lefoka (1993) (called SL); Osman and Potts (1989) (called

OP); and Ogbu and Smith (1990) (called OS). These heuristics can be broadly divided into two categories: constructive heuristics and improvement heuristics. A constructive heuristic generates a schedule of jobs so that once a decision is taken it cannot be changed for improvement. The heuristics of Campbell et al. (1970); Nawaz et al. (1983); and Koulamas (1998) are of the constructive type. An improvement heuristic starts with an initial sequence of jobs and an attempt is made to improve the objective function by changing the job positions in the sequence. Some improvement heuristics are due to Ben-Daya (1998); Taillard (1990); Osman and Potts (1989); and Ogbu and Smith (1990). These heuristics are also called the metaheuristics. The classification of these heuristic algorithms based on makespan criterion for flowshop scheduling is shown in Fig.3.1.

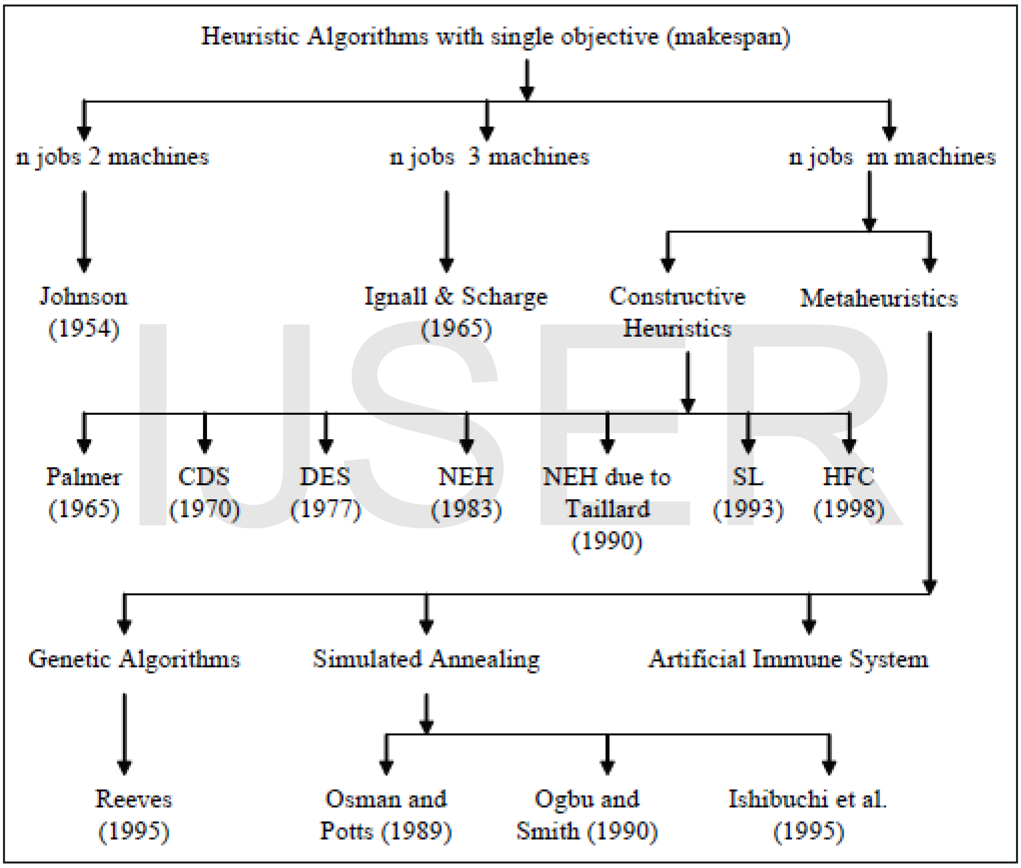


**Fig.3.1: Classification of Heuristic Algorithms Based on Makespan Criterion for Flowshop Scheduling**

Johnson's 2-machine algorithm gives the optimal solution with a view to minimizing the makespan but it fails to generalize to m-machine problems. Comparing CDS, NEH, and other heuristics, it is observed from Park's (1981) study that NEH is least biased and best operated of the heuristics and the CDS algorithm comes next. It is also proposed in the paper of Nawaz et al. that it would continue to perform better than CDS for problems with large numbers of machines and jobs (m, n > 100). But Park, during his study, omitted Dannengbring's heuristic "rapid access with extensive search" (RAES), which has been found superior to CDS as pointed out by Turner and Booth (1987). Turner and Booth also observed that NEH proved to be more efficient than RAES based on both makespan and CPU time as measures of performance. So, NEH is clearly an improvement over the other published heuristics and RAES comes next.

To compare between NEH and SL, Sarin and Lefoka have shown that NEH is less effective than SL for scheduling problems with large number of machines. The effectiveness of NEH tends to improve as the number of jobs increases. Sarin and Lefoka also noted that the SL heuristic produces inferior solution compared to NEH for small and medium number of machines (m<100) and outperforms the NEH heuristic consistently for m>150 regardless the number of jobs. Also, the CPU time of the SL heuristic is very small compared to that of NEH. NEH requires more computational time because the work involved in computing makespan is a function of the number of jobs and each partial sequence is also a function of the number of machines.

Later, Koulamas (1998) in his paper proposed an effective constructive heuristic (called HFC for “heuristic flowshop scheduling with $C_{max}$ objective”) for flowshop scheduling problem with makespan objective. Computational results indicate that HFC performs as well as NEH on scheduling problems where a permutation schedule is expected to be optimal. However, HFC shows superiority over NEH on problems where a nonpermutation schedule may be optimal.



**3.3 Overview of Scheduling Methods on Total Flowtime Criterion**

Apart from the heuristics on makespan criterion, there are some significant heuristics, which are either total-flow-time criterion based (Rajendran & Chaudhuri, 1991) or multiple criteria based (Rajendran, 1994). A survey of the flowshop scheduling literature has revealed that very little significant research work has been done on multi-objective criteria (considering more than two objectives) simultaneously.

Some noteworthy heuristics on total flow time criterion have been developed by Gupta (1971), Ho and Chang (1991), Rajendran and Chaudhuri (1991), Rajendran (1993), Ho (1995), Woo and Yim (1998), Liu and Reeves (2001), Allahverdi and Aldowaisan (2002), and Framinan and Leisten (2003). Framinan et al. (2005) present two new composite heuristics and the subsequent computational results show these heuristics to be efficient for the flowtime minimization in flowshops.

Rajendran and Chaudhuri (1991) propose three heuristics and compare with those of Gupta (1971), Miyazaki et al. (1978), and Ho and Chang (1991). The results reveal that their heuristics perform superior results in terms of both quality of the solution and computational time. Rajendran (1993) develops a new heuristic, which is better than that of Rajendran and Chaudhuri (1991), but at the expense of large computational effort.

Ho (1995) proposes an improvement heuristic based on finding local solution by adjacent pair wise interchange method, and later improves the solution by the insertion method. This heuristic performs better than the previous heuristics, but it consumes much higher CPU time for large problem sizes (Framinan & Leisten, 2003).

Framinan and Leisten (2003) propose a new heuristic based on the idea of optimizing partial Heuristics and Metaheuristics for Solving Scheduling Problems schedules, already presented in the heuristic by Nawaz et al. (1983). The computational results show that their heuristic is currently the best for total flow time minimization in flowshops. It is compared with that of Woo and Yim (1998) having the same time-complexity of O ($n^4$m).

It is revealed from the survey of scheduling literature that the three heuristics of Rajendran and Chaudhuri (1991) yield consistently near optional solutions and require smaller CPU time. However, the heuristic by Framinan and Leisten (2003) outperforms the current best heuristic but its only disadvantage is that it requires higher computational effort.



### 3.4 Some Methods for Scheduling and Sequencing

#### *3.4.1 Single machine scheduling methods:*

1) Shortest processing time rule (SPT)
2) Earliest due date rule (EDD)
3) Weighted mean flow time method
4) Naughton’s algorithm to minimize the number of tardy jobs
5) Hodgson’s algorithm to minimize tardiness.

#### *3.4.2 Flowshop scheduling methods:*

1) Two-Machine Flow-shop Problem
    a) Johnson’s Rule
    b) Kusiak’s Rule
2) Heuristics for general *m*-Machine Problems
    a) Palmer’s Heuristic Algorithm
    b) Gupta’s Heuristic Algorithm
    c) CDS Heuristic Algorithm
    d) RA Heuristic Algorithm

#### *3.4.3 Jobshop scheduling methods:*

1) JSP Mathematical / Graph Models
    a) Integer Programming Model
    b) Linear Programming Model
    c) Disjunctive Graph Model

2) Conventional Heuristics for JSP
    a) Priority Dispatching Heuristics
    b) Shifting Bottleneck Heuristic
    c) Randomized Dispatching Heuristics

**3.5 Performance Measure Used to Decide the Best Optimal Solution**

Average WIP inventory

1) Makespan (total time to finish processing)
2) Due date (lateness, earliness, and tardiness)
3) Machine Utilization
4) Labor Utilization

**3.6 Assumptions for Solving Scheduling Problems**

1) Set of Jobs to Schedule
    a) Typically assume that our set of jobs is fixed
2) Time
    a) Need to assume times are known,
    b) Usually assume times are fixed and independent of processing order or activities that take place elsewhere in the factory
    c) Quality
    d) Assume we never produce a bad part
    e) Machines
    f) Assume we never have breakdowns.



"Optimal" scheduling methods' assumptions can be violated in many ways:

Variability in

a) Setup times
b) Processing times
c) Interruptions
d) Changes in the set of jobs.

**3.7 Performance Evaluation Criteria for Scheduling Methods**

Choice of sequencing methodology to choose is dependent on the performance evaluation criteria to be applied:

1) Total job completion time.
2) Average job completion time.
3) Average job waiting time.
4) Average job lateness.
5) Average number of jobs in the system.
6) Average number of jobs waiting.

7) Set-up costs.
8) In-process inventory costs.

### 3.8 Goals of Scheduling Methods

Efficient utilization …

1) Staff
2) Equipment
3) Facilities

Minimization of …

1) Customer waiting time
2) Inventories
3) Processing time

### 3.9 Tools for Scheduling

Gantt Charts are used as a visual aid for loading and scheduling. The Gantt schedule can illustrate the relationship between work activities having duration, events without duration that indicate a significant completion, that represent major achievements or decision points.

### 3.10 Approaches to Scheduling

1) Forward scheduling
   a) Scheduling ahead from a point in time (e.g., now)
   b) Useful to answer the question "How long will it take to complete this job?"
2) Backward scheduling
   a) Scheduling backward from a future due date
   b) Useful to answer the questions:
      - "Can we complete this job in time?"
      - "When is the latest we can start this job and still complete it by the due date?"

### 3.11 Summary

By Scheduling, we assign a particular time for completing a particular job. The main objective of scheduling is to arrive at a position where we will get minimum processing time. Scheduling is the process by which you look at the time available to you, and plan how you will use it to achieve the goals you have identified. By using a schedule properly, you can:

1) Understand what you can realistically achieve with your time;
2) Plan to make the best use of the time available;
3) Leave enough time for things you absolutely must do;
4) Preserve contingency time to handle the unexpected and
5) Minimize stress by avoiding over-commitment to others.

# CHAPTER 4
## METHODS USED

### 4.1 Introduction
Scheduling is the process by which you look at the time available to you, and plan how you will use it to achieve the goals you have identified. Some priority rules that are tested for scheduling are FCFS, SPT, LPT, and PR/TR (assign the highest priority to the part whose proportion of required output is lagging behind most). Machine utilization and production rate are used as the criteria for evaluating part input and scheduling procedures.

### 4.2 Methodology
Manufacturing scheduling theory is concerned with the right allocation of machines to operations over time. The basic work of scheduler is to design an optimal FMS schedule according to a certain measure of performance, or scheduling criterion. This work focuses on productivity oriented-makespan criteria. Makespan is the time length from the starting of the first operation of the first demand to the finishing of the last operation of the last demand. The approaches used in this work were the comparison based on four heuristic algorithms namely Gupta's algorithm, CDS algorithm, RA algorithm and Palmer's algorithms were proposed. Here the main objective is to find the efficient heuristics algorithm for minimizing the makespan. In this work hierarchical approach were used to determine the optimal makespan criteria.



### 4.3 Problem Statement
There is a flowshop scheduling problem in which all the parameters like processing time, due date, re-fixturing time, and set-up time are given. The value of the makespan of batch-processing machines in a flowshop based on comparison of Gupta's heuristics, RA heuristic's, Palmer's heuristics, CDS heuristics are proposed. Analytic solutions in all the heuristics are investigated. Gantt chart was generated to verify the effectiveness of the proposed approaches. Here the heuristics approaches for planning problems are proposed which provides a way to optimize the makespan which is our objective function.

### 4.4 Flowshop Scheduling
It is a typical combinatorial optimization problem, where each job has to go through the processing in each and every machine on the shop floor. Each machine has same sequence of jobs. The jobs have different processing time for different machines. So in this case we arrange the jobs in a particular order and get many combinations and we choose that combination where we get the minimum makespan.

In an m-machine flowshop, there are m stages in series, where there exist one or more machines at each stage. Each job has to be processed in each of the 'm' stages in the same order. That is, each job has to be processed first in stage 1, then in stage 2, and so on. Operation times for each job in different stages may be different. We classify flowshop problems as:

1) Flowshop (there is one machine at each stage).
2) No-wait flowshop (a succeeding operation starts immediately after the preceding operation completes).
3) Flexible (hybrid) flowshop (more than one machine exist in at least one stage) and
4) Assembly flowshop (each job consists of specific operations, each of which has to be performed on a pre-determined machine of the first stage, and an assembly operation to be performed on the second stage machine).

### 4.5 Flowshop Scheduling Methods

#### *4.5.1 Two-Machine Flow-shop Problem*

1) Johnson's Rule
2) Kusiak's Rule

#### *4.5.2 Heuristics for general m-Machine Problems*

1) Palmer's Heuristic Algorithm.
2) Gupta's Heuristic Algorithm.
3) CDS's Heuristic Algorithm.
4) RA's Heuristic Algorithm.



### 4.6 General Description

1) There are *m* machines and *n* jobs.
2) Each job consists of *m* operations and
   a) each operation requires a different machine
3) *n* jobs have to be processed in the same sequence on *m* machines.
4) Processing time of job *i* on machine *j* is given by $t_{ij}$
   a) $i$ =1, 2, 3, 4, ……, *n* and j =1, 2, 3, 4, ……, *m*
5) Makespan: find the sequence of jobs minimizing the maximum flow time.

### 4.7 Main Assumptions

1) Every job has to be processed on all machines in the order ($j$ = 1, 2, …….., *m*).
2) Every machine processes only one job at a time.
3) Every job is processed on one machine at a time.
4) Operations are not preemptive.
5) Set-up times for the operations are sequence-independent and are included in the processing times.

Operating sequences of the jobs are the same on every machine, and the common sequence has to be determined.

## 4.8 Three Categories of FSP

1) Deterministic flow-shop scheduling problem:
   - Assume that fixed processing times of jobs are known.
2) Stochastic flow-shop scheduling problem:
   - Assume that processing times vary according to chosen probability distribution.
3) Fuzzy flow-shop scheduling problem:
   - Assume that a fuzzy due date is assigned to each job to represent the grade of satisfaction of decision makers for the completion time of the job.

## 4.9 Two-Machine Flowshop Problem

### *4.9.1 Johnson's Rule:*

Johnson's Algorithm:

- An optimal sequence is directly constructed with an adaptation of this result by a one-pass scanning procedure.
- Let *I* denote the job list and let *S* denote the schedule:

Step 1: Let $U = \{j \mid t_{i1} < t_{i2}\}$ and $V = \{j \mid t_{i1} \geq t_{i2}\}$

Step 2: Sort jobs in *U* with non-decreasing order of $t_{i1}$

Step 3: Sort jobs in *V* with non-increasing order of $t_{i2}$

Step 4: An optimal sequence is the ordered set *U* followed by the Ordered set *V*.

***Consider 2-Machines, 8-Jobs problem:***

**Table 4.1: Two-Machine Flow-shop Problem for Johnson's Algorithm**

| **Job *i*** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $t_{i1}$ | 5 | <u>2</u> | <u>1</u> | 7 | <u>6</u> | <u>3</u> | 7 | 5 |
| $t_{i2}$ | <u>2</u> | 6 | 2 | <u>5</u> | 6 | 7 | <u>2</u> | <u>1</u> |

The solution constructed as follows

Step 1: Job sets U = {2, 3, 6} and V = {1, 4, 5, 7, 8}

Step 2: Sort jobs in U as follows:

| | | | |
|---|---|---|---|
| Job i: | 3 | 2 | 6 |
| $t_{i1}$ : | 1 | 2 | 3 |

Step 3: Sort jobs in V as follows:

| | | | | | |
|---|---|---|---|---|---|
| Job i: | 5 | 4 | 7 | 1 | 8 |
| $t_{i2}$ : | 6 | 5 | 2 | 2 | 1 |

Step 4: **An optimal sequence is {3, 2, 6, 5, 4, 7, 1, 8}**

Thus total processing time can be calculated as:

**Table 4.2: Total Processing Time for 8-Jobs, 2-Machines by Johnson's Heuristic Model**

| Job i | M/c 1 | | M/c 1 | |
|---|---|---|---|---|
| | Time | | Time | |
| | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 |
| **2** | 1 | 3 | 3 | 9 |
| **6** | 3 | 6 | 9 | 16 |
| **5** | 6 | 12 | 16 | 22 |
| **4** | 12 | 19 | 22 | 27 |
| **7** | 19 | 26 | 27 | 29 |
| **1** | 26 | 31 | 31 | 33 |
| **8** | 31 | 36 | 36 | 37 |

Therefore, total processing time = **37 (Units)**
Total Idle Time for M/c 1 = 37-36 = 1 (Units)
Total Idle Time for M/c 1 = 1+2+3 = 6 (Units)



The **Gantt chart** according to **Table 4.2** is shown in **Figure 4.1**

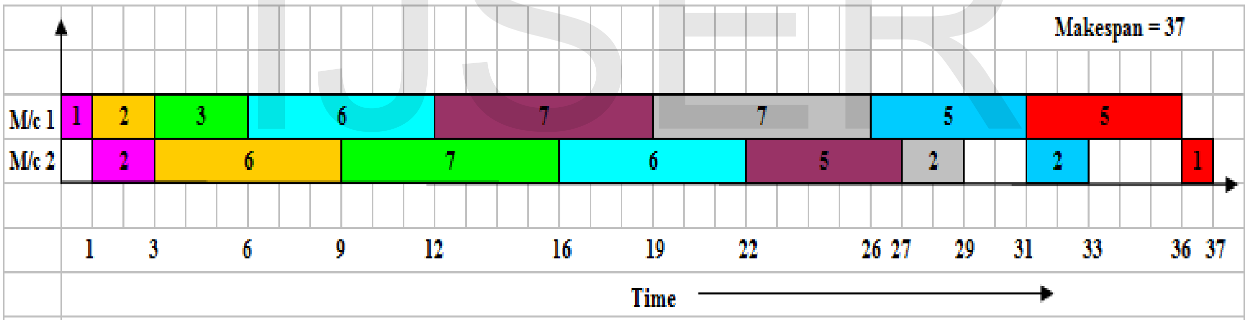


Fig. 4.1. Gantt Chart for 8-Jobs, 2-Machines Problem by Johnson's Heuristic Model

### *4.9.2 Kusiak's Rule*

Kusiak's Algorithm

Step 1: Set k = 1, l = n

Step 2: For each operation, store the shortest processing time, and corresponding machine number.

Step 3: Sort the resulting list, including the triplets "Operation number /processing time/machine number" in increasing value of processing time.

Step 4: For each entry in the sorted list:

If machine number is 1, then

(i) set the corresponding operation number in position *k*,

(ii) Set k = k + 1

else

(i) set the corresponding operation number in position *k*,

(ii) Set l = l - 1

end

Step 5: Stop if the entire list of operations has been exhausted.

***Consider 2-Machines, 8-Jobs problem:***

**Table 4.3: Two-Machine Flow-shop Problem for Kusiak's Algorithm**

| **Job *i*** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| $t_{i1}$ | 5 | 2 | 1 | 7 | 6 | 3 | 7 | 5 |
| $t_{i2}$ | 2 | 6 | 2 | 5 | 6 | 7 | 2 | 1 |

The solution constructed as follows:

| $t_{i1}$ | $t_{i2}$ | → | **Job i** | $t_{(i)}$ | **m** |
|---|---|---|---|---|---|
| 5 | 2 | | 3 | 1 | 1 |
| 2 | 6 | | 8 | 1 | 2 |
| 1 | 2 | | 2 | 2 | 1 |
| 7 | 5 | | 1 | 2 | 2 |
| 6 | 6 | | 7 | 2 | 2 |
| 3 | 7 | | 6 | 3 | 1 |
| 7 | 2 | | 4 | 5 | 2 |
| 5 | 1 | | 5 | 6 | 1 |

**M/c 1** ↓ **M/c 2** ↑



**An optimal solution sequence is {3, 2, 6, 5, 4, 7, 1, 8}**

Thus total processing time can be calculated as:

**Table 4.4: Total Processing Time for 8-Jobs, 2-Machines by Kusiak's Heuristic Model**

| **Job i** | **M/c 1 Time** | | **M/c 1 Time** | |
|---|---|---|---|---|
| | **In** | **Out** | **In** | **Out** |
| **3** | 0 | 1 | 1 | 3 |
| **2** | 1 | 3 | 3 | 9 |
| **6** | 3 | 6 | 9 | 16 |
| **5** | 6 | 12 | 16 | 22 |
| **4** | 12 | 19 | 22 | 27 |
| **7** | 19 | 26 | 27 | 29 |
| **1** | 26 | 31 | 31 | 33 |
| **8** | 31 | 36 | 36 | 37 |

Therefore, total processing time = **37 (Units)**
Total Idle Time for M/c 1 = 37-36 = 1 (Units)
Total Idle Time for M/c 1 = 1+2+3 = 6 (Units)

The **Gantt chart** according to **Table 4.4** is shown in **Figure 4.2**

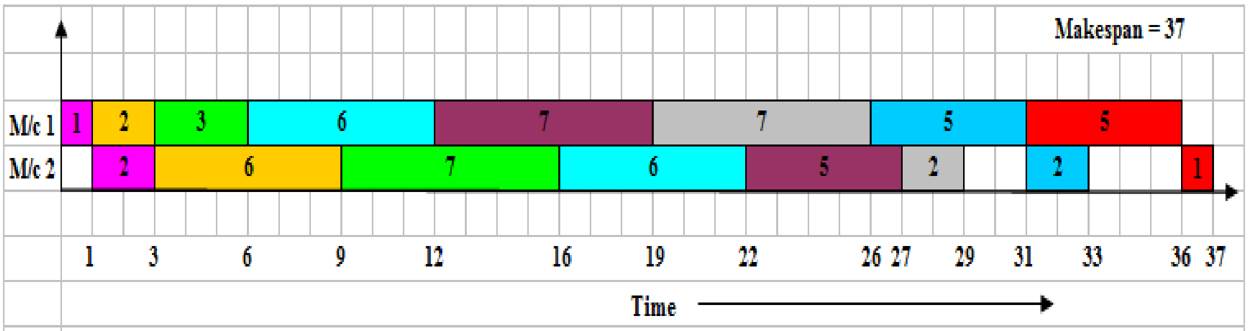

Fig. 4.2. Gantt Chart for 8-Jobs, 2-Machines Problem by Kusiak's Heuristic Model

**4.10 Heuristics for General 3-Machines and 8-Jobs Problems**
***4.10.1 Palmer's Heuristic Algorithm***
***4.10.2 Gupta's Heuristic Algorithm***
***4.10.3 CDS's Heuristic Algorithm***
***4.10.4 RA's Heuristic Algorithm***



***4.10.1 Palmer's Heuristic Rule:***
Algorithm: Palmer's Heuristic:
Procedure: Palmer's Heuristic
Input: job list i, machine m;
Output: schedule *S*;

begin
    for $i$ = 1 to $n$
    for $j$ =1 to $m$
        Calculates $s_i = s_i + (2j\text{-}m\text{-}1)t_{ij}$ ; // step 1:
    Permutation schedule is constructed by sequencing the jobs in
    Non-increasing order of $s_i$ such as: $s_{i1} \geq s_{i2} \geq \ldots\ldots \geq s_{in}$; // step 2:
end
        Output optimal sequence is obtained as schedule S; // step 3:
end.

***Consider 3-Machines, 8-Jobs problem:***

**Table 4.5: General 3-Machines and 8-Jobs Problem**

| Job ⇨ / M/c ⇩ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **1** | 5 | 2 | 1 | 7 | 6 | 3 | 7 | 5 |
| **2** | 2 | 6 | 2 | 5 | 6 | 7 | 2 | 1 |
| **3** | 3 | 4 | 6 | 2 | 1 | 5 | 4 | 7 |

The solution constructed as follows:

Step 1: Set the slope index $s_i$ for job i as:

$s_1$= {(2*1-3-1)*5}+{(2*2-3-1)*2}+{(2*3-3-1)*3}= -10 + 6 = -4

$s_2$= {(2*1-3-1)*2}+{(2*2-3-1)*6}+{(2*3-3-1)*4}= -4 + 8 = 4

$s_3$= {(2*1-3-1)*1}+{(2*2-3-1)*2}+{(2*3-3-1)*6}= -2 + 12 = 10

$s_4$= {(2*1-3-1)*7}+{(2*2-3-1)*5}+{(2*3-3-1)*2}= -14 + 4 = -10

$s_5$= {(2*1-3-1)*6}+{(2*2-3-1)*6}+{(2*3-3-1)*1}= -12 + 2 = -10

$s_6$= {(2*1-3-1)*3}+{(2*2-3-1)*7}+{(2*3-3-1)*5}= -6 + 10 = 4

$s_7$= {(2*1-3-1)*7}+{(2*2-3-1)*2}+{(2*3-3-1)*4}= -14 + 8 = -6

$s_8$= {(2*1-3-1)*5}+{(2*2-3-1)*1}+{(2*3-3-1)*7}= -10 + 14 = 4

Step 2: Jobs are sequenced according:

$s_3 \geq s_2 \geq s_6 \geq s_8 \geq s_1 \geq s_7 \geq s_4 \geq s_5$

$10 \geq 4 \geq 4 \geq 4 \geq -4 \geq -6 \geq -10 \geq -10$

Step 3: **Output optimal sequence is {3, 2, 6, 8, 1, 7, 4, 5}**

Thus total processing time can be calculated as:

**Table 4.6: Total Processing Time for 3-Machines, 8-Jobs by Palmer's Heuristic Model**

| Job i | M/c 1 Time | | M/c 2 Time | | M/c 3 Time | |
|---|---|---|---|---|---|---|
| | In | Out | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 | 3 | 9 |
| **2** | 1 | 3 | 3 | 9 | 9 | 13 |
| **6** | 3 | 6 | 9 | 16 | 16 | 21 |
| **8** | 6 | 11 | 16 | 17 | 21 | 28 |
| **1** | 11 | 16 | 17 | 19 | 28 | 31 |
| **7** | 16 | 23 | 23 | 25 | 31 | 35 |
| **4** | 23 | 30 | 30 | 35 | 35 | 37 |
| **5** | 30 | 36 | 36 | 42 | 42 | 43 |

Therefore, total processing time = **43 (Units)**

Total Idle Time for M/c 1 = 43-36 = 7 (Units)

Total Idle Time for M/c 2 = 1+4+5+1+(43-42) = 12 (Units)
Total Idle Time for M/c 3 = 3+3+5 = 11 (Units)

The **Gantt chart** according to **Table 4.6** is shown in **Figure 4.3**

### *4.10.2. Gupta's Heuristic Rule:*

Algorithm: Gupta's Heuristic
Procedure: Gupta's Heuristic
Input: job list *i*, machine *m*;
Output: schedule *S*;

```
begin
    for i = 1 to n
    for k =1 to m-1
        if t_i1 < t_im then
            e_i=1;
        else
            e_i = -1;
                calculate s_i= e_i /min{t_i,k + t_i,k+1}; // step 1:
end
    Permutation schedule is constructed by sequencing the jobs in
    Non-increasing order of s_i such as:; // step 2:
end
    Output optimal sequence is obtained as schedule S // step 3:
end .
```

***Consider the above 3-Machines, 8-Jobs problem:***

The solution constructed as follows:

Step 1: Set the slope index $s_i$ for job i as:

$s_1$ = -1 / min {$t_{11}$+$t_{12}$, $t_{12}$+$t_{13}$}= -1 / min {5+2, 2+3}= -1 / min {7, 5} = - 0.2
$s_2$ = 1 / min {8, 10} = 0.125
$s_3$ = 1 / min {3, 8} = 0.33
$s_4$ = -1 / min {12, 7} = -0.143
$s_5$ = -1 / min {12, 7} = -0.143
$s_6$ = 1 / min {10, 12} = 0.1
$s_7$ = -1 / min {9, 6} = -0.166
$s_8$ = 1 / min {6, 8} = 0.166

Step 2: Jobs are sequenced according:

$s_3 \geq s_8 \geq s_2 \geq s_6 \geq s_4 \geq s_5 \geq s_7 \geq s_1$
$0.33 \geq 0.166 \geq 0.125 \geq 0.1 \geq -0.143 \geq -0.143 \geq -0.166 \geq -0.2$

Step 3: **Output optimal sequence is {3, 8, 2, 6, 4, 5, 7, 1}**

Thus total processing time can be calculated as:

**Table 4.7: Total Processing Time for 3-Machines, 8-Jobs by Gupta's Heuristic Model**

| Job i | M/c 1 | | M/c 2 | | M/c 3 | |
|---|---|---|---|---|---|---|
| | Time | | Time | | Time | |
| | In | Out | In | Out | In | Out |
| 3 | 0 | 1 | 1 | 3 | 3 | 9 |
| 8 | 1 | 6 | 6 | 7 | 9 | 16 |
| 2 | 6 | 8 | 8 | 14 | 16 | 20 |
| 6 | 8 | 11 | 14 | 21 | 21 | 26 |
| 4 | 11 | 18 | 21 | 26 | 26 | 28 |
| 5 | 18 | 24 | 26 | 32 | 32 | 33 |
| 7 | 24 | 31 | 32 | 34 | 34 | 38 |
| 1 | 31 | 36 | 36 | 38 | 38 | 41 |



Therefore, total processing time = **41 (Units)**
Total Idle Time for M/c 1 = 41-36 = 5 (Units)
Total Idle Time for M/c 2 = 1+3+1+2+(41-38) = 10 (Units)
Total Idle Time for M/c 3 = 3+1+4+1 = 9 (Units)

The **Gantt chart** according to **Table 4.7** is shown in **Fig. 4.4**

ISSN 2229-5518

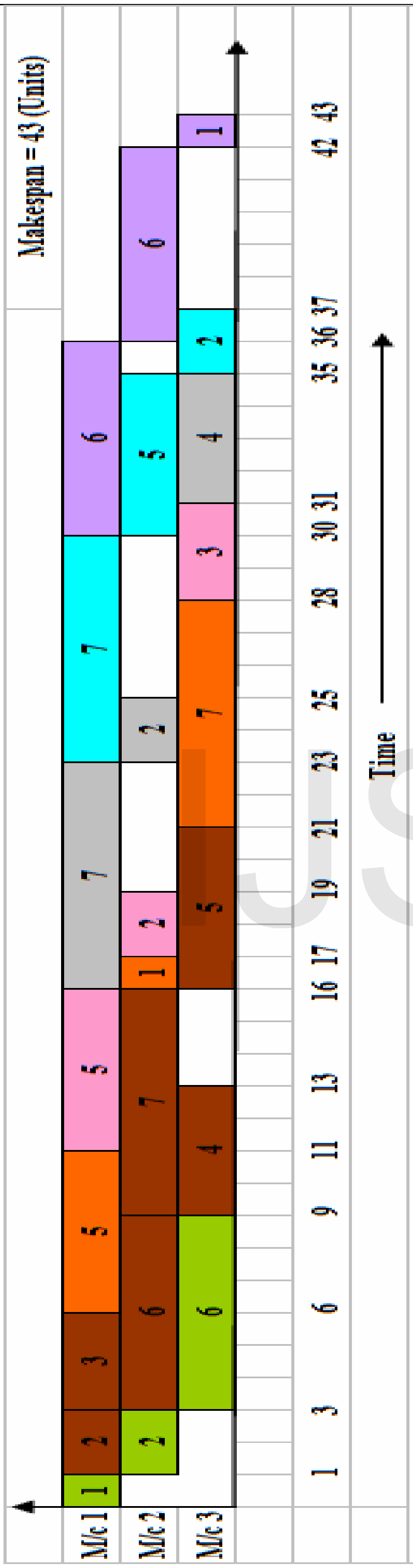


Fig. 4.3. Gantt Chart for 8-Jobs, 3-Machines Problem by Palmer's Heuristic Model

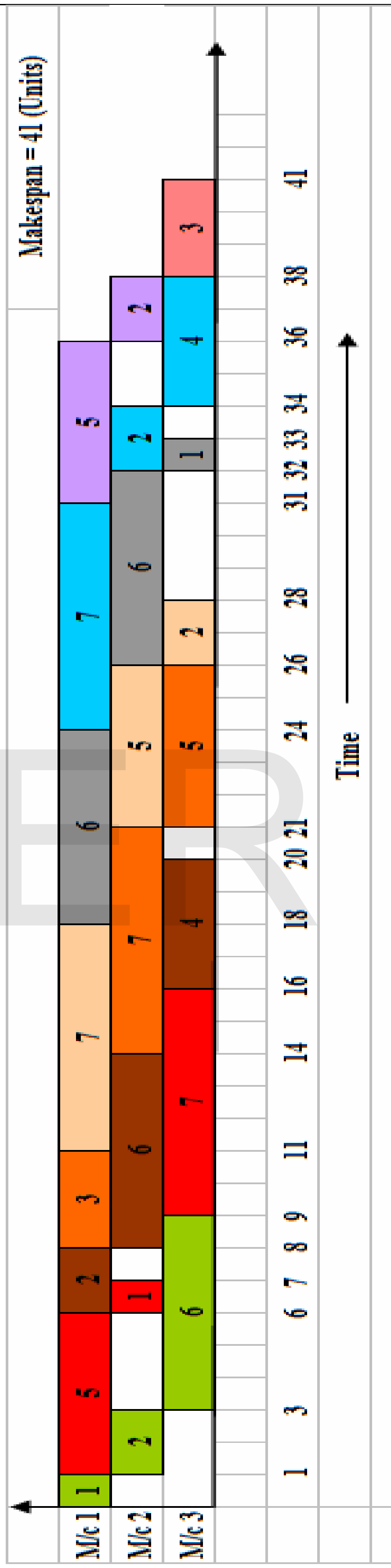


Fig. 4.4. Gantt Chart for 8-Jobs, 3-Machines Problem by Gupta's Heuristic Model

***4.10.3. CDS's Heuristic Rule:***

Algorithm: CDS's Heuristic
Procedure: CDS's Heuristic
Input: job list i, machine m;
Output: schedule S;

begin
    for i = 1 to n
    for j = 1 to m-1
        Calculate $t_{i1}' = t_{i1} + t_{ij}$;
    for j= m-1 to m
        Calculate $t_{i2}' = t_{i2} + t_{ij}$;
end
        calculate $U = \{i \mid t_{i1}' < t_{i2}'\}$ and
            $V = \{i \mid t_{i1}' \geq t_{i2}'\}$; // step 1
        sort jobs in U with non-decreasing order of $t_{i1}'$; //step 2
        sort jobs in V with non-increasing order of $t_{i2}'$; //step 3
    Output optimal sequence is obtained as schedule S by U and V //step 4
end



***Consider 3-Machines, 8-Jobs problem:***

**Table 4.8: General 3-Machines and 8- Jobs Problems:**

| **Job i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| $t_{i1}$ | 5 | 2 | 1 | 7 | 6 | 3 | 7 | 5 |
| $t_{i2}$ | 2 | 6 | 2 | 5 | 6 | 7 | 2 | 1 |
| $t_{i3}$ | 3 | 4 | 6 | 2 | 1 | 5 | 4 | 7 |

The solution constructed as follows:

Step 1:

| **Job i** | $t_{i1}'$ | $t_{i2}'$ |
|---|---|---|
| **1** | $t_{11} + t_{12} = 5+2 = 7$ | $t_{12} + t_{13} = 2+3 = 5$ |
| **2** | $t_{21} + t_{22} = 2+6 = 8$ | $t_{22} + t_{23} = 6+4 = 10$ |
| **3** | $t_{31} + t_{32} = 1+2 = 3$ | $t_{32} + t_{33} = 2+6 = 8$ |
| **4** | $t_{41} + t_{42} = 7+5 = 12$ | $t_{42} + t_{43} = 5+2 = 7$ |
| **5** | $t_{51} + t_{52} = 6+6 = 12$ | $t_{52} + t_{53} = 6+1 = 7$ |
| **6** | $t_{61} + t_{62} = 3+7 = 10$ | $t_{62} + t_{63} = 7+5 = 12$ |
| **7** | $t_{71} + t_{72} = 7+2 = 9$ | $t_{72} + t_{73} = 2+4 = 6$ |
| **8** | $t_{81} + t_{82} = 5+1 = 6$ | $t_{82} + t_{83} = 1+7 = 8$ |

Jobs sets are: U = {2, 3, 6, 8} and V = {1, 4, 5, 7}

Step 2: Sort jobs in U as follows:

| | | | | |
|---|---|---|---|---|
| Job i : | 3 | 8 | 2 | 6 |
| $t_{i1}$' : | 3 | 6 | 8 | 10 |

Step 3: Sort jobs in V as follows:

| | | | | |
|---|---|---|---|---|
| Job i : | 4 | 5 | 7 | 1 |
| $t_{i1}$' : | 7 | 7 | 6 | 5 |

Step 4: **Output optimal sequence is {3, 8, 2, 6, 4, 5, 7, 1}**

Thus total processing time can be calculated as:

**Table 4.9: Total Processing Time for 3-Machines, 8-Jobs by CDS's Heuristic Model**

| Job i | M/c 1 | | M/c 2 | | M/c 3 | |
|---|---|---|---|---|---|---|
| | Time | | Time | | Time | |
| | In | Out | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 | 3 | 9 |
| **8** | 1 | 6 | 6 | 7 | 9 | 16 |
| **2** | 6 | 8 | 8 | 14 | 16 | 20 |
| **6** | 8 | 11 | 14 | 21 | 21 | 26 |
| **4** | 11 | 18 | 21 | 26 | 26 | 28 |
| **5** | 18 | 24 | 26 | 32 | 32 | 33 |
| **7** | 24 | 31 | 32 | 34 | 34 | 38 |
| **1** | 31 | 36 | 36 | 38 | 38 | 41 |

Therefore, total processing time = **41 (Units)**
Total Idle Time for M/c 1 = 41-36 = 5 (Units)
Total Idle Time for M/c 2 = 1+3+1+2+(41-38) = 10 (Units)
Total Idle Time for M/c 3 = 3+1+4+1 = 9 (Units)

The **Gantt chart** according to **Table 4.9** is shown in **Figure 4.5**

### *4.10.4. RA's Heuristic Rule:*

Algorithm: RA's Heuristic
Procedure: RA's Heuristic
Input: job list i, machine m;
Output: schedule *S*;

begin

for i = 1 to n

for j =1 to m-1

$w_{j1} = m-(j-1)$, $w_{j2} = j$;

$$t_{i1}' = \sum_{j=1}^{m} w_{j1}.t_{ij} \text{ and } t_{i2}' = \sum_{j=1}^{m} w_{j2}.t_{ij}$$

where weights are defined as follows:

$W_1 = \{w_{j1} \mid j=1, 2,\ldots,m\}=\{m,m-1,\ldots,2,1\}$

$W_2 = \{w_{j2} \mid j=1, 2,\ldots,m\}=\{1,2,\ldots,m-1,m\}$

Calculate $U = \{i \mid t_{i1}' < t_{i2}'\}$ and $V = \{i \mid t_{i1}' \geq t_{i2}'\}$; //step 1

sort jobs in U with non-decreasing order of $t_{i1}'$; //step 2

sort jobs in V with non-increasing order of $t_{i2}'$; //step 3

output: optimal sequence is obtained as schedule S by U and V // step 4

end

***Consider the above 3-Machines and 8-Jobs problem:***

The solution constructed as follows:

Step 1:

| **Job i** | **$t_{i1}'$** | **$t_{i2}'$** |
|---|---|---|
| **1** | $(w_{11}.t_{11})+(w_{21}.t_{12})+(w_{31}.t_{13}) = 22$ | $(w_{12}.t_{11})+(w_{22}.t_{12})+(w_{32}.t_{13}) = 18$ |
| **2** | 22 | 26 |
| **3** | 13 | 23 |
| **4** | 33 | 23 |
| **5** | 31 | 21 |
| **6** | 28 | 32 |
| **7** | 29 | 23 |
| **8** | 24 | 28 |

Jobs sets are: U = {2, 3, 6, 8} and V = {1, 4, 5, 7}

Step 2: Sort jobs in U as follows:

Job i : 3 2 8 6

$t_{i1}'$ : 13 22 24 28

Step 3: Sort jobs in V as follows:

Job i : 4 7 5 1

$t_{i1}'$ : 23 23 21 18

Step 4: **Output optimal sequence is {3, 2, 8, 6, 4, 5, 7, 1}**

Thus total processing time can be calculated as:

**Table 4.10: Total Processing Time for 3-Machines, 8-Jobs by RA's Heuristic Model**

| Job i | M/c 1 | | M/c 2 | | M/c 3 | |
|---|---|---|---|---|---|---|
| | Time | | Time | | Time | |
| | In | Out | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 | 3 | 9 |
| **2** | 1 | 3 | 3 | 9 | 9 | 13 |
| **8** | 3 | 8 | 9 | 10 | 13 | 20 |
| **6** | 8 | 11 | 11 | 18 | 20 | 25 |
| **4** | 11 | 18 | 18 | 23 | 25 | 27 |
| **5** | 18 | 24 | 24 | 30 | 30 | 31 |
| **7** | 24 | 31 | 31 | 33 | 33 | 37 |
| **1** | 31 | 36 | 36 | 38 | 38 | 41 |

Therefore, total processing time = **41 (Units)**
Total Idle Time for M/c 1 = 41-36 = 5 (Units)
Total Idle Time for M/c 2 = 1+1+1+1+3+(41-38) = 10 (Units)
Total Idle Time for M/c 3 = 3+3+2+1 = 9 (Units)

The **Gantt chart** according to **Table 4.10** is shown in **Fig. 4.6**

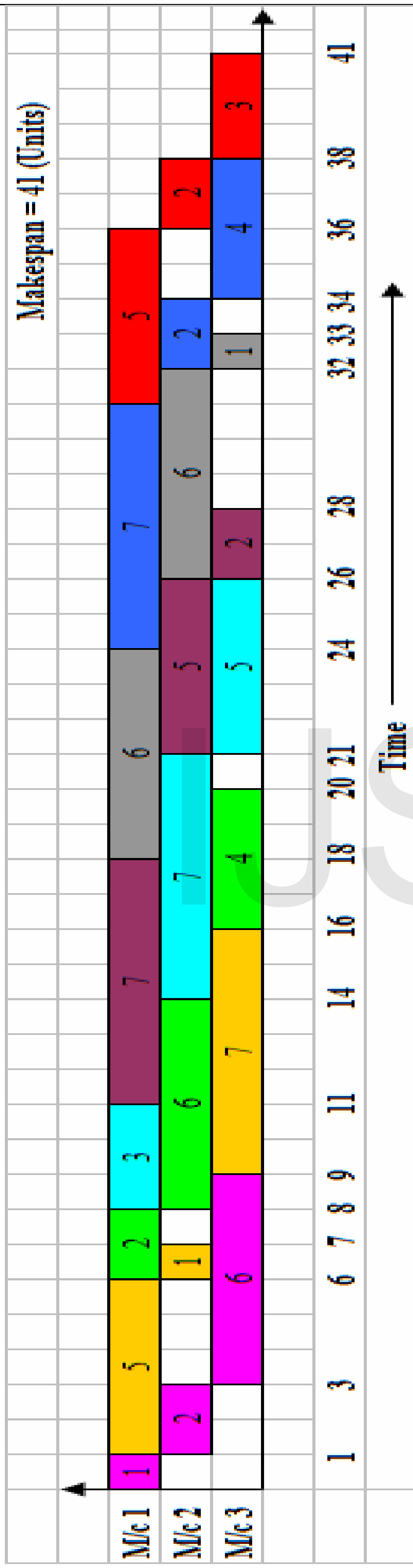


Fig. 4.5: Gantt Chart for 8-Jobs, 3-Machines Problem by CDS's Heuristic Model

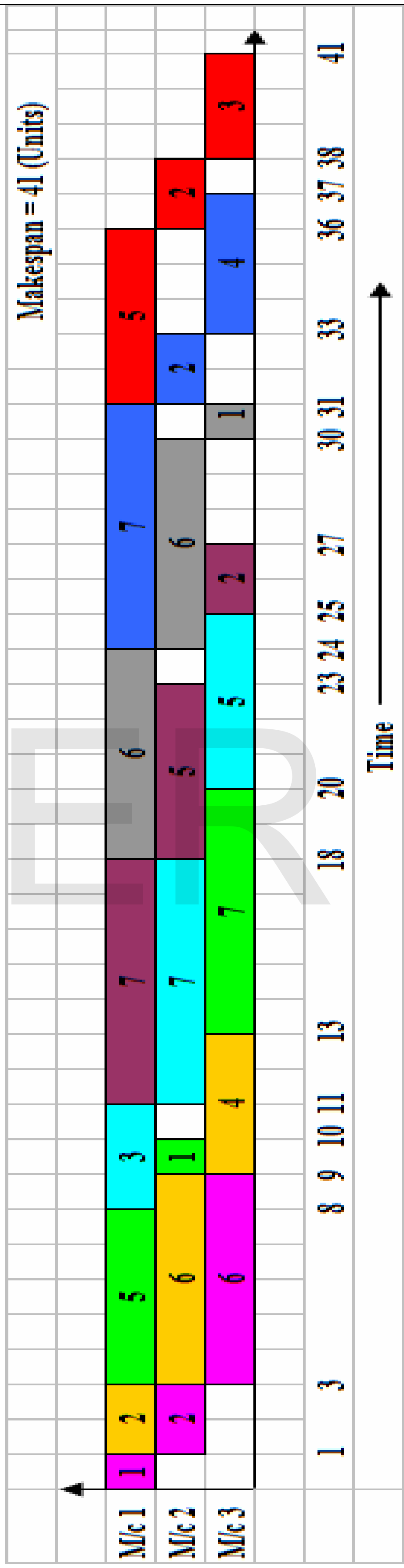


Fig. 4.6: Gantt Chart for 8-Jobs, 3-Machines Problem by RA's Heuristic Model

### 4.11 Heuristics for General 10-Machine and 10-Jobs Problems

Let us consider the case of 10-machines and 10-jobs to compare the various heuristics procedure: The problem in terms of processing times is as follows:

| Table shows the processing time of 10 -jobs on 10 –machines where<br>**Processing time = Operation Time (OT) and Re-fixturing Time (RT)** |
|---|

**Table 4.11: 10-Machines and 10-Jobs Flowshop Problem**

| **Job ⇨ / M/c ⇩** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 5 | 2 | 1 | 7 | 6 | 3 | 7 | 5 | 7 | 4 |
| **2** | 2 | 6 | 2 | 5 | 6 | 7 | 2 | 1 | 8 | 3 |
| **3** | 3 | 4 | 2 | 6 | 1 | 5 | 4 | 7 | 6 | 5 |
| **4** | 5 | 2 | 1 | 3 | 8 | 2 | 6 | 1 | 9 | 8 |
| **5** | 7 | 6 | 3 | 2 | 6 | 2 | 5 | 7 | 1 | 3 |
| **6** | 9 | 2 | 7 | 3 | 4 | 1 | 5 | 3 | 8 | 1 |
| **7** | 7 | 5 | 2 | 2 | 3 | 5 | 1 | 6 | 2 | 3 |
| **8** | 8 | 2 | 5 | 4 | 9 | 3 | 2 | 6 | 1 | 8 |
| **9** | 2 | 6 | 4 | 2 | 6 | 2 | 5 | 2 | 6 | 3 |
| **10** | 7 | 1 | 4 | 2 | 4 | 6 | 2 | 2 | 6 | 7 |



#### *4.11.1. For Gupta's Heuristics:*

The solution is constructed as follows:

Step 1: Set the slope index $s_i$ for job $i$ as:

Calculate $s_i = e_i / \min\{t_{i,k} + t_{i,k+1}\}$;

if $t_{i1} < t_{im}$ then

$e_i = 1$;

else

$e_i = -1$;

Optimal sequence is constructed on the basic of decreasing order of slope values:

$s_1$ = 1/min (48, 50) = 0.0208

$s_2$ = -1/min (35, 34) = -0.0294

$s_3$ = 1/min (27, 30) = 0.0370

$s_4$ = -1/min (34, 29) = -0.0345

$s_5$ = -1/min (49, 47) = -0.0213

$s_6$ = 1/min (30, 33) = 0.0333

$s_7$ = -1/min (37, 32) = -0.0313

$s_8$ = -1/min (38, 35) = -0.0286

$s_9$ = -1/min (48, 47) = -0.0213

$s_{10}$ = 1/min (38, 41) = 0.0263

**Optimal sequence:**

**0.037 ≥ 0.033 ≥ 0.026 ≥ 0.021 ≥ -0.021 ≥ -0.021 ≥-0.029 ≥ -0.029 ≥ -0.031 ≥ -0.034**

**$s_3$ ≥ $s_6$ ≥ $s_{10}$ ≥ $s_1$ ≥ $s_5$ ≥ $s_9$ ≥ $s_8$ ≥ $s_2$ ≥ $s_7$ ≥ $s_4$**

Thus total processing time can be calculated as:

**Table 4.12: Total Processing Time for 10-Jobs, 10-Machines by Gupta's Heuristic Model**

| Job i | M/c 1 Time | | M/c 2 Time | | M/c 3 Time | | M/c 4 Time | | M/c 5 Time | | M/c 6 Time | | M/c 7 Time | | M/c 8 Time | | M/c 9 Time | | M/c 10 Time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 | 3 | 5 | 5 | 6 | 6 | 9 | 9 | 16 | 16 | 18 | 18 | 23 | 23 | 27 | 27 | 31 |
| **6** | 1 | 4 | 4 | 11 | 11 | 16 | 16 | 18 | 18 | 20 | 20 | 21 | 21 | 26 | 26 | 29 | 29 | 31 | 31 | 37 |
| **10** | 4 | 8 | 11 | 14 | 16 | 21 | 21 | 29 | 29 | 32 | 32 | 33 | 33 | 36 | 36 | 44 | 44 | 47 | 47 | 54 |
| **1** | 8 | 13 | 14 | 16 | 21 | 24 | 29 | 34 | 34 | 41 | 41 | 50 | 50 | 57 | 57 | 65 | 65 | 67 | 67 | 74 |
| **5** | 13 | 19 | 19 | 25 | 25 | 26 | 34 | 42 | 42 | 48 | 50 | 54 | 57 | 60 | 65 | 74 | 74 | 80 | 80 | 88 |
| **9** | 19 | 26 | 26 | 34 | 34 | 40 | 42 | 51 | 51 | 52 | 54 | 62 | 62 | 64 | 74 | 75 | 80 | 86 | 88 | 94 |
| **8** | 26 | 31 | 34 | 35 | 40 | 47 | 51 | 52 | 52 | 59 | 62 | 65 | 65 | 71 | 75 | 81 | 86 | 88 | 94 | 96 |
| **2** | 31 | 33 | 35 | 41 | 47 | 51 | 52 | 54 | 59 | 65 | 65 | 67 | 71 | 76 | 81 | 83 | 88 | 94 | 96 | 97 |
| **7** | 33 | 40 | 41 | 43 | 51 | 55 | 55 | 61 | 65 | 70 | 70 | 75 | 76 | 77 | 83 | 85 | 94 | 99 | 99 | 101 |
| **4** | 40 | 47 | 47 | 52 | 55 | 61 | 61 | 64 | 70 | 72 | 75 | 78 | 78 | 80 | 85 | 89 | 99 | 101 | 101 | 103 |



Therefore, total processing time = **103 (Units)**

Total Idle Time for M/c 1 = 103-47 = 56 (Units)

Total Idle Time for M/c 2 = 1+1+3+1+4+(103-52) = 61 (Units)

Total Idle Time for M/c 3 = 3+6+1+8+(103-61) = 60 (Units)

Total Idle Time for M/c 4 = 5+10+3+1+(103-64) = 58 (Units)

Total Idle Time for M/c 5 = 6+9+9+2+1+3+(103-72) = 61 (Units)

Total Idle Time for M/c 6 = 9+4+11+8+3+(103-78) = 60 (Units)

Total Idle Time for M/c 7 = 16+3+7+14+2+1+2+(103-80) = 68 (Units)

Total Idle Time for M/c 8 = 18+3+7+13+(103-89) = 55 (Units)

Total Idle Time for M/c 9 = 23+2+13+18+7+(103-101) = 65 (Units)

Total Idle Time for M/c 10 = 27+10+13+6+2 = 58 (Units)

The **Gantt chart** according to **Table 4.12** is shown in **Fig. 4.7**

### *4.11.2. For CDS Heuristics:*

The solution for the above problem is constructed as follows:

| Job | $t_{i1}$' | $t_{i2}$' |
|---|---|---|
| **1** | 48 | 50 |
| **2** | 35 | 34 |
| **3** | 27 | 30 |
| **4** | 34 | 29 |
| **5** | 49 | 47 |
| **6** | 30 | 33 |
| **7** | 37 | 32 |
| **8** | 38 | 35 |
| **9** | 48 | 47 |
| **10** | 38 | 41 |

Optimal sequence is constructed on the basic of increasing order of $t_{i1}$'and $t_{i2}$'

**Optimal sequence: 3-6-10-1-9-5-8-2-7-4**

Thus total processing time can be calculated as:

**Table 4.13: Total Processing Time for 10-Jobs, 10-Machines by CDS's Heuristic Model**

| Job i | M/c 1 Time | | M/c 2 Time | | M/c 3 Time | | M/c 4 Time | | M/c 5 Time | | M/c 6 Time | | M/c 7 Time | | M/c 8 Time | | M/c 9 Time | | M/c 10 Time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 | 3 | 5 | 5 | 6 | 6 | 9 | 9 | 16 | 16 | 18 | 18 | 23 | 23 | 27 | 27 | 31 |
| **6** | 1 | 4 | 4 | 11 | 11 | 16 | 16 | 18 | 18 | 20 | 20 | 21 | 21 | 26 | 26 | 29 | 29 | 31 | 31 | 37 |
| **10** | 4 | 8 | 11 | 14 | 16 | 21 | 21 | 29 | 29 | 32 | 32 | 33 | 33 | 36 | 36 | 44 | 44 | 47 | 47 | 54 |
| **1** | 8 | 13 | 14 | 16 | 21 | 24 | 29 | 34 | 34 | 41 | 41 | 50 | 50 | 57 | 57 | 65 | 65 | 67 | 67 | 74 |
| **9** | 13 | 20 | 20 | 28 | 28 | 34 | 34 | 43 | 43 | 44 | 50 | 58 | 58 | 60 | 65 | 66 | 67 | 73 | 74 | 80 |
| **5** | 20 | 26 | 28 | 34 | 34 | 35 | 43 | 51 | 51 | 57 | 58 | 62 | 62 | 65 | 66 | 75 | 75 | 81 | 81 | 89 |
| **8** | 26 | 31 | 34 | 35 | 35 | 42 | 51 | 52 | 57 | 64 | 64 | 67 | 67 | 73 | 75 | 81 | 81 | 83 | 89 | 91 |
| **2** | 31 | 33 | 35 | 41 | 42 | 46 | 52 | 54 | 64 | 70 | 70 | 72 | 73 | 78 | 81 | 83 | 83 | 89 | 91 | 92 |
| **7** | 33 | 40 | 41 | 43 | 46 | 50 | 54 | 60 | 70 | 75 | 75 | 80 | 80 | 81 | 83 | 85 | 89 | 94 | 94 | 96 |
| **4** | 40 | 47 | 47 | 52 | 52 | 58 | 60 | 63 | 75 | 77 | 80 | 83 | 83 | 85 | 85 | 89 | 94 | 96 | 96 | 98 |

Therefore, total processing time = **98 (Units)**

Total Idle Time for M/c 1 = 98-47 = 51 (Units)

Total Idle Time for M/c 2 = 1+1+4+4+(98-52) = 56 (Units)

Total Idle Time for M/c 3 = 3+6+4+2+(98-58) = 55 (Units)

Total Idle Time for M/c 4 = 5+10+3+(98-63) = 53 (Units)

Total Idle Time for M/c 5 = 6+9+9+2+2+7+(98-77) = 56 (Units)

Total Idle Time for M/c 6 = 9+4+11+8+2+3+3+(98-83) = 55 (Units)

Total Idle Time for M/c 7 = 16+3+7+14+1+2+2+2+2+(98-85) = 62 (Units)

Total Idle Time for M/c 8 = 18+3+7+13+(98-89) = 50 (Units)

Total Idle Time for M/c 9 = 23+2+13+18+2+(98-96) = 60 (Units)
Total Idle Time for M/c 10 = 27+10+13+1+2 = 53 (Units)

The **Gantt chart** according to **Table 4.13** is shown in **Fig. 4.8**

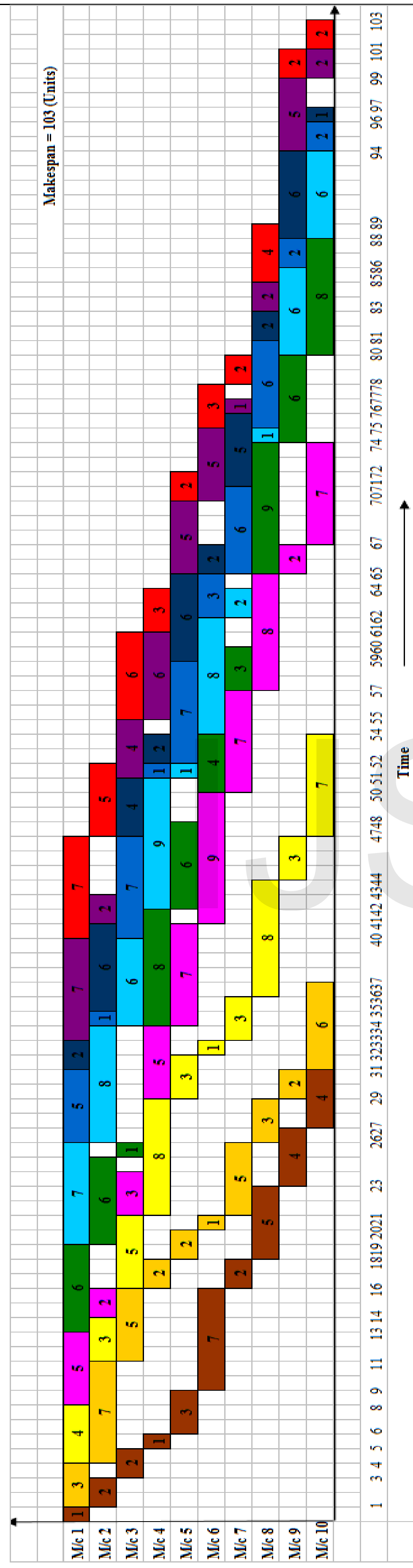


Fig. 4.7: Gantt Chart for 10-Jobs, 10-Machines Problem by Gupta's Heuristic Model

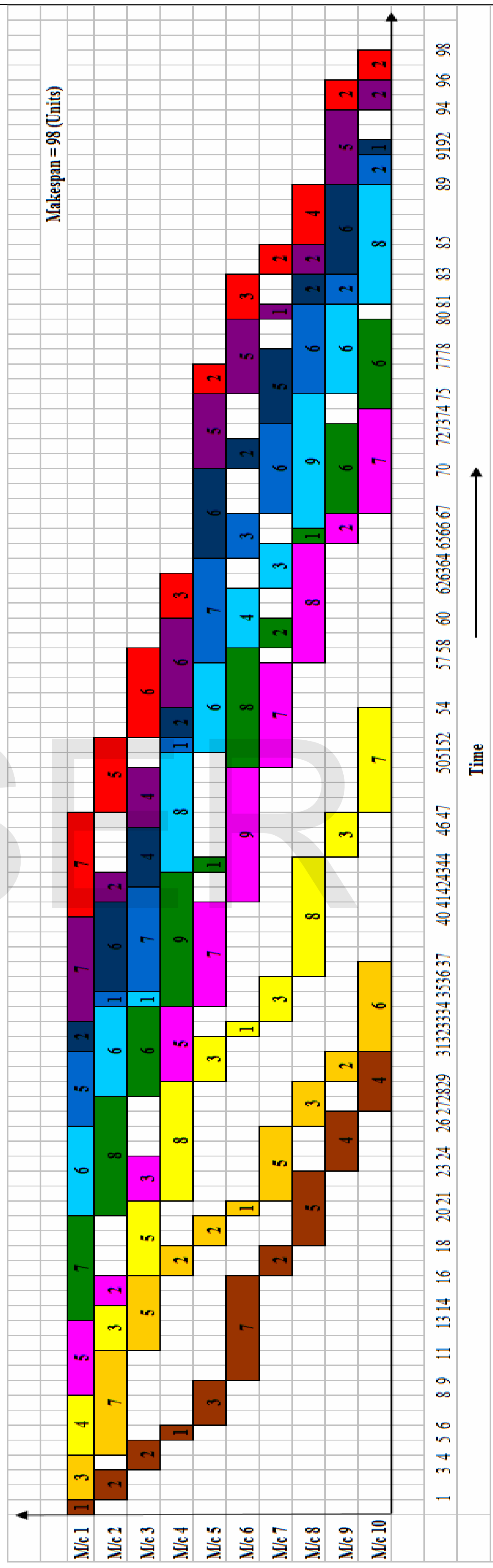


Fig. 4.8 Gantt Chart for 10-Jobs, 10-Machines Problem by CDS's Heuristic Model

### *4.11.3. For RA Heuristics:*

The solution for the above problem is constructed as follows:

| Job | $t_{i1}$' | $t_{i2}$' |
|---|---|---|
| **1** | 277 | 328 |
| **2** | 205 | 191 |
| **3** | 139 | 202 |
| **4** | 237 | 159 |
| **5** | 289 | 294 |
| **6** | 203 | 193 |
| **7** | 239 | 190 |
| **8** | 227 | 213 |
| **9** | 328 | 266 |
| **10** | 235 | 260 |

Optimal sequence is constructed on the basic of increasing order of $t_{i1}$'and $t_{i2}$'

| **Optimal sequence: 3-10-1-5-9-8-6-2-7-4** |
|---|

Thus total processing time can be calculated as:



**Table 4.14: Total Processing Time for 10-Jobs, 10-Machines by RA's Heuristic Model**

| Job i | M/c 1 Time | | M/c 2 Time | | M/c 3 Time | | M/c 4 Time | | M/c 5 Time | | M/c 6 Time | | M/c 7 Time | | M/c 8 Time | | M/c 9 Time | | M/c 10 Time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 | 3 | 5 | 5 | 6 | 6 | 9 | 9 | 16 | 16 | 18 | 18 | 23 | 23 | 27 | 27 | 31 |
| **10** | 1 | 5 | 5 | 8 | 8 | 13 | 13 | 21 | 21 | 24 | 24 | 25 | 25 | 28 | 28 | 36 | 36 | 39 | 39 | 46 |
| **1** | 5 | 10 | 10 | 12 | 13 | 16 | 21 | 26 | 26 | 33 | 33 | 42 | 42 | 49 | 49 | 57 | 57 | 59 | 59 | 66 |
| **5** | 10 | 16 | 16 | 22 | 22 | 23 | 26 | 34 | 34 | 40 | 42 | 46 | 49 | 52 | 57 | 66 | 66 | 72 | 72 | 80 |
| **9** | 16 | 23 | 23 | 31 | 31 | 37 | 37 | 46 | 46 | 47 | 47 | 55 | 55 | 57 | 66 | 67 | 72 | 78 | 80 | 86 |
| **8** | 23 | 28 | 31 | 32 | 37 | 44 | 46 | 47 | 47 | 54 | 55 | 58 | 58 | 64 | 67 | 73 | 78 | 80 | 86 | 88 |
| **6** | 28 | 31 | 32 | 39 | 44 | 49 | 49 | 51 | 54 | 56 | 58 | 59 | 64 | 69 | 73 | 76 | 80 | 82 | 88 | 94 |
| **2** | 31 | 33 | 39 | 45 | 49 | 53 | 53 | 55 | 56 | 62 | 62 | 64 | 69 | 74 | 76 | 78 | 82 | 88 | 94 | 95 |
| **7** | 33 | 40 | 45 | 47 | 53 | 57 | 57 | 63 | 63 | 68 | 68 | 73 | 74 | 75 | 78 | 80 | 88 | 93 | 95 | 97 |
| **4** | 40 | 47 | 47 | 52 | 57 | 63 | 63 | 66 | 68 | 70 | 73 | 76 | 76 | 78 | 80 | 84 | 93 | 95 | 97 | 99 |

Therefore, total processing time = **99 (Units)**

Total Idle Time for M/c 1 = 99-47 = 52 (Units)

Total Idle Time for M/c 2 = 1+2+2+4+1+(99-52) = 57 (Units)

Total Idle Time for M/c 3 = 3+3+6+8+(99-63) = 56 (Units)

Total Idle Time for M/c 4 = 5+7+3+2+2+2+(99-66) = 54 (Units)

Total Idle Time for M/c 5 = 6+12+2+1+6+1+(99-70) = 57 (Units)

Total Idle Time for M/c 6 = 9+8+8+1+3+4+(99-76) = 56 (Units)

Total Idle Time for M/c 7 = 16+7+14+3+1+1+(99-78) = 63 (Units)

Total Idle Time for M/c 8 = 18+5+13+(99-84) = 51 (Units)
Total Idle Time for M/c 9 = 23+9+18+7+(99-95) = 61 (Units)
Total Idle Time for M/c 10 = 27+8+13+6 = 54 (Units)
The **Gantt chart** according to **Table 4.14** is shown in **Fig. 4.9**

***4.11.4. For Palmers Heuristics:***

The solution for the above problem is constructed as follows:
Slope = $(m-1)t_{j,m} + (m-3)t_{j,(m-1)} + (m-5)t_{j,(m-2)}$……………………………….

For 10-jobs and 10- machines:
$s_1 = (m-1)t_{1,10} + (m-3)t_{1,9}$………………………………………+$(m-19)t_{1,1}$
For 10 machines (m=10)
= (10-1)*7+(10-3)*2+(10-5)*8+(10-7)*7+(10-9)*9+(10-11)*7+(10-13)*5
+(10-15)*3+(10-17)*2+(10-19)*5
= 51
Similarly
$s_2 = -14$
$s_3 = 63$
$s_4 = -78$
$s_5 = 5$
$s_6 = -10$
$s_7 = -49$
$s_8 = -14$
$s_9 = -62$
$s_{10} = 25$

Optimal sequence is constructed on the basic of decreasing order of slope values

**Optimal sequence:** $s_3 \geq s_1 \geq s_{10} \geq s_5 \geq s_6 \geq s_2 \geq s_8 \geq s_7 \geq s_9 \geq s_4$

Thus total processing time can be calculated as:

International Journal of Scientific & Engineering Research (IJSER), May 2013
ISSN 2229-5518

**Table 4.15: Total Processing Time for 10-Jobs, 10-Machines by Palmer's Heuristic Model**

| Job i | M/c 1 Time | | M/c 2 Time | | M/c 3 Time | | M/c 4 Time | | M/c 5 Time | | M/c 6 Time | | M/c 7 Time | | M/c 8 Time | | M/c 9 Time | | M/c 10 Time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **3** | 0 | 1 | 1 | 3 | 3 | 5 | 5 | 6 | 6 | 9 | 9 | 16 | 16 | 18 | 18 | 23 | 23 | 27 | 27 | 31 |
| **1** | 1 | 6 | 6 | 8 | 8 | 11 | 11 | 16 | 16 | 23 | 23 | 32 | 32 | 39 | 39 | 47 | 47 | 49 | 49 | 56 |
| **10** | 6 | 10 | 10 | 13 | 13 | 18 | 18 | 26 | 26 | 29 | 32 | 33 | 39 | 42 | 47 | 55 | 55 | 58 | 58 | 65 |
| **5** | 10 | 16 | 16 | 22 | 22 | 23 | 26 | 34 | 34 | 40 | 40 | 44 | 44 | 47 | 55 | 64 | 64 | 70 | 70 | 78 |
| **6** | 16 | 19 | 22 | 29 | 29 | 34 | 34 | 36 | 40 | 42 | 44 | 45 | 47 | 52 | 64 | 67 | 70 | 72 | 78 | 84 |
| **2** | 19 | 21 | 29 | 35 | 35 | 39 | 39 | 41 | 42 | 48 | 48 | 50 | 52 | 57 | 67 | 69 | 72 | 78 | 84 | 85 |
| **8** | 21 | 26 | 35 | 36 | 39 | 46 | 46 | 47 | 48 | 55 | 55 | 58 | 58 | 64 | 69 | 75 | 78 | 80 | 85 | 87 |
| **7** | 26 | 33 | 36 | 38 | 46 | 50 | 50 | 56 | 56 | 61 | 61 | 66 | 66 | 67 | 75 | 77 | 80 | 85 | 87 | 89 |
| **9** | 33 | 40 | 40 | 48 | 50 | 56 | 56 | 65 | 65 | 66 | 66 | 74 | 74 | 76 | 77 | 78 | 85 | 91 | 91 | 97 |
| **4** | 40 | 47 | 48 | 53 | 56 | 62 | 65 | 68 | 68 | 70 | 74 | 77 | 77 | 79 | 79 | 83 | 91 | 93 | 97 | 99 |

Therefore, total processing time = **99**
Total Idle Time for M/c 1 = 99-47 = 52 (Units)
Total Idle Time for M/c 2 = 1+3+2+3+2+(99-53) = 57 (Units)
Total Idle Time for M/c 3 = 3+3+2+4+6+1+(99-62) = 56 (Units)
Total Idle Time for M/c 4 = 5+5+2+3+5+3+(99-68) = 54 (Units)
Total Idle Time for M/c 5 = 6+6+3+5+1+4+2+(99-70) = 56 (Units)
Total Idle Time for M/c 6 = 12+4+7+3+5+3+(99-77) = 56 (Units)
Total Idle Time for M/c 7 = 19+11+2+1+2+7+1+(99-79) = 63 (Units)
Total Idle Time for M/c 8 = 21+13+1+(99-83) = 51 (Units)
Total Idle Time for M/c 9 = 26+17+6+8+(99-93) = 63 (Units)
Total Idle Time for M/c 10 = 30+15+2+5+2 = 54 (Units)

The **Gantt chart** according to **Table 4.15** is shown in **Fig. 4.10**

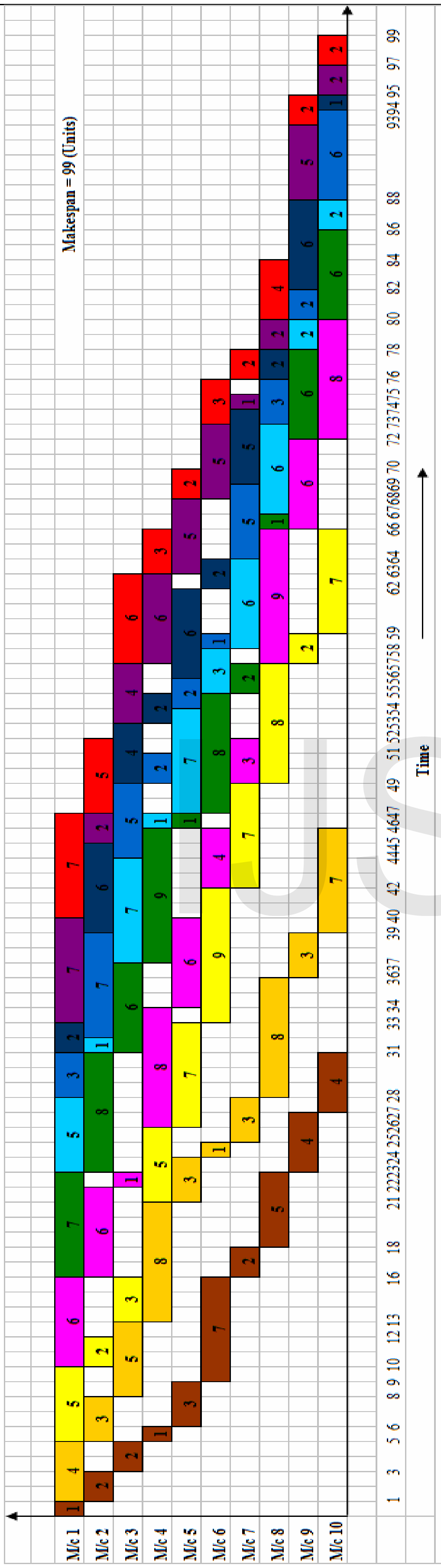


Fig. 4.9: Gantt Chart for 10-Jobs, 10-Machines Problem by RA's Heuristic Model

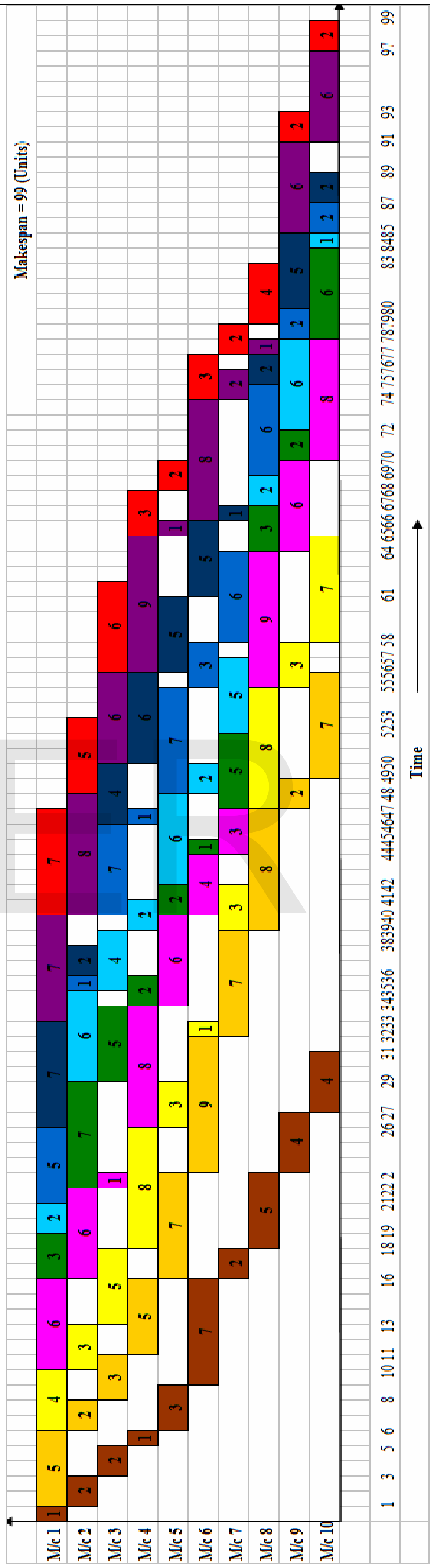


Fig. 4.10: Gantt Chart for 10-Jobs, 10-Machines Problem by Palmer's Heuristic Model

### 4.12 Heuristics for General 8-Machines and 10-Jobs Problems

Again let us consider the case of 8-machines and 10-jobs to compare the various heuristics procedure: The problem in terms of processing times is as

| Table shows the processing time of 10-jobs on 8-machines where<br>**Processing time = Operation time (OT) and Re-fixturing time (RT)** |
|---|

**Table 4.16: 8-Machines and 10-Jobs Flowshop Problem**

| Job ⇨ / M/c ⇩ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 6 | 3 | 8 | 4 | 9 | 3 | 5 | 2 | 1 | 6 |
| **2** | 5 | 9 | 1 | 6 | 8 | 2 | 9 | 8 | 4 | 3 |
| **3** | 1 | 5 | 6 | 3 | 2 | 4 | 4 | 9 | 6 | 5 |
| **4** | 7 | 7 | 4 | 1 | 9 | 3 | 2 | 1 | 2 | 5 |
| **5** | 9 | 2 | 3 | 5 | 2 | 7 | 4 | 6 | 5 | 2 |
| **6** | 3 | 5 | 9 | 6 | 5 | 2 | 8 | 3 | 4 | 7 |
| **7** | 4 | 6 | 5 | 7 | 9 | 3 | 6 | 4 | 3 | 1 |
| **8** | 2 | 1 | 9 | 7 | 6 | 5 | 6 | 8 | 9 | 9 |

#### *4.12.1. For Gupta's Heuristics:*

The solution is constructed as follows:

Step 1: Set the slope index $s_i$ for job $i$ as:

Calculate $s_i = e_i / \min\{t_{i,k} + t_{i,k+1}\}$;

if $t_{i1} < t_{im}$ then

$e_i = 1$;

else

$e_i = -1$;

$s_1$ = -1/min (35, 31) = -0.0323
$s_2$ = -1/min (37, 35) = -0.0286
$s_3$ = 1/min (36, 37) = 0.0278
$s_4$ = 1/min (32, 35) = 0.0313
$s_5$ = -1/min (44, 41) = -0.0244
$s_6$ = 1/min (24, 26) = 0.0417
$s_7$ = 1/min (38, 39) = 0.0263
$s_8$ = 1/min (33, 39) = 0.0303
$s_9$ = 1/min (25, 33) = 0.0400
$s_{10}$ = 1/min (29, 32) = 0.0345

International Journal of Scientific & Engineering Research (IJSER), May 2013
ISSN 2229-5518

**Optimal sequence:**

**0.042 ≥ 0.040 ≥ 0.034 ≥ 0.031 ≥ 0.030 ≥ 0.028 ≥0.026 ≥ -0.024 ≥ -0.029 ≥ -0.032**

$s_6 \geq s_9 \geq s_{10} \geq s_4 \geq s_8 \geq s_3 \geq s_7 \geq s_5 \geq s_2 \geq s_1$

Thus total processing time can be calculated as:

**Table 4.17: Total Processing Time for 8-Machines, 10-Jobs by Gupta's Heuristic Model**

| Job i | M/c 1 | | M/c 2 | | M/c 3 | | M/c 4 | | M/c 5 | | M/c 6 | | M/c 7 | | M/c 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | | Time | | Time | | Time | | Time | | Time | | Time | | Time | |
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **6** | 0 | 3 | 3 | 5 | 5 | 9 | 9 | 12 | 12 | 19 | 19 | 21 | 21 | 24 | 24 | 29 |
| **9** | 3 | 4 | 5 | 9 | 9 | 15 | 15 | 17 | 19 | 24 | 24 | 28 | 28 | 31 | 31 | 40 |
| **10** | 4 | 10 | 10 | 13 | 15 | 20 | 20 | 25 | 25 | 27 | 28 | 35 | 35 | 36 | 40 | 49 |
| **4** | 10 | 14 | 14 | 20 | 20 | 23 | 25 | 26 | 27 | 32 | 35 | 41 | 41 | 48 | 49 | 56 |
| **8** | 14 | 16 | 20 | 28 | 28 | 37 | 37 | 38 | 38 | 44 | 44 | 47 | 48 | 52 | 56 | 64 |
| **3** | 16 | 24 | 28 | 29 | 37 | 43 | 43 | 47 | 47 | 50 | 50 | 59 | 59 | 64 | 64 | 73 |
| **7** | 24 | 29 | 29 | 38 | 43 | 47 | 47 | 49 | 50 | 54 | 59 | 67 | 67 | 73 | 73 | 79 |
| **5** | 29 | 38 | 38 | 46 | 47 | 49 | 49 | 58 | 58 | 60 | 67 | 72 | 73 | 82 | 82 | 88 |
| **2** | 38 | 41 | 46 | 55 | 55 | 60 | 60 | 67 | 67 | 69 | 72 | 77 | 82 | 88 | 88 | 89 |
| **1** | 41 | 47 | 55 | 60 | 60 | 61 | 67 | 74 | 74 | 83 | 83 | 86 | 88 | 92 | 92 | 94 |



Therefore, total processing time = **94 (Units)**
Total Idle Time for M/c 1 = 94-47 = 47 (Units)
Total Idle Time for M/c 2 = 3+1+1+(94-60) = 39 (Units)
Total Idle Time for M/c 3 = 5+5+6+(94-61) = 49 (Units)
Total Idle Time for M/c 4 = 9+3+3+11+5+2+(94-74) = 53 (Units)
Total Idle Time for M/c 5 = 12+1+6+3+4+7+5+(94-83) = 49 (Units)
Total Idle Time for M/c 6 = 19+3+3+3+6+(94-86) = 42 (Units)
Total Idle Time for M/c 7 = 21+4+4+5+7+3+(94-92) = 46 (Units)
Total Idle Time for M/c 8 = 24+2+3+3 = 32 (Units)

The **Gantt chart** according to **Table 4.17** is shown in **Fig. 4.11**

### *4.12.2. For CDS Heuristics:*

The solution for the above problem is constructed as follows:

| Job | $t_{i1}$’ | $t_{i2}$’ |
|---|---|---|
| **1** | 35 | 31 |
| **2** | 37 | 35 |
| **3** | 36 | 37 |
| **4** | 32 | 35 |
| **5** | 44 | 41 |
| **6** | 24 | 26 |
| **7** | 38 | 39 |
| **8** | 33 | 39 |
| **9** | 25 | 33 |
| **10** | 29 | 32 |

Optimal sequence is constructed on the basic of increasing order of $t_{i1}$’ and $t_{i2}$’

| **Optimal sequence: 6-9-10-4-8-3-7-5-2-1** |
|---|

Thus total processing time can be calculated as:

**Table 4.18: Total Processing Time for 8-Machines, 10-Jobs by CDS’s Heuristic Model**

| Job i | M/c 1 | | M/c 2 | | M/c 3 | | M/c 4 | | M/c 5 | | M/c 6 | | M/c 7 | | M/c 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | | Time | | Time | | Time | | Time | | Time | | Time | | Time | |
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **6** | 0 | 3 | 3 | 5 | 5 | 9 | 9 | 12 | 12 | 19 | 19 | 21 | 21 | 24 | 24 | 29 |
| **9** | 3 | 4 | 5 | 9 | 9 | 15 | 15 | 17 | 19 | 24 | 24 | 28 | 28 | 31 | 31 | 40 |
| **10** | 4 | 10 | 10 | 13 | 15 | 20 | 20 | 25 | 25 | 27 | 28 | 35 | 35 | 36 | 40 | 49 |
| **4** | 10 | 14 | 14 | 20 | 20 | 23 | 25 | 26 | 27 | 32 | 35 | 41 | 41 | 48 | 49 | 56 |
| **8** | 14 | 16 | 20 | 28 | 28 | 37 | 37 | 38 | 38 | 44 | 44 | 47 | 48 | 52 | 56 | 64 |
| **3** | 16 | 24 | 28 | 29 | 37 | 43 | 43 | 47 | 47 | 50 | 50 | 59 | 59 | 64 | 64 | 73 |
| **7** | 24 | 29 | 29 | 38 | 43 | 47 | 47 | 49 | 50 | 54 | 59 | 67 | 67 | 73 | 73 | 79 |
| **5** | 29 | 38 | 38 | 46 | 47 | 49 | 49 | 58 | 58 | 60 | 67 | 72 | 73 | 82 | 82 | 88 |
| **2** | 38 | 41 | 46 | 55 | 55 | 60 | 60 | 67 | 67 | 69 | 72 | 77 | 82 | 88 | 88 | 89 |
| **1** | 41 | 47 | 55 | 60 | 60 | 61 | 67 | 74 | 74 | 83 | 83 | 86 | 88 | 92 | 92 | 94 |

Therefore, total processing time = **94 (Units)**

Total Idle Time for M/c 1 = 94-47 = 47 (Units)

Total Idle Time for M/c 2 = 3+1+1+(94-60) = 39 (Units)

Total Idle Time for M/c 3 = 5+5+6+(94-61) = 49 (Units)

Total Idle Time for M/c 4 = 9+3+3+11+5+2+(94-74) = 53 (Units)
Total Idle Time for M/c 5 = 12+1+6+3+4+7+5+(94-83) = 49 (Units)
Total Idle Time for M/c 6 = 19+3+3+3+6+(94-86) = 42 (Units)
Total Idle Time for M/c 7 = 21+4+4+5+7+3+(94-92) = 46 (Units)
Total Idle Time for M/c 8 = 24+2+3+3 = 32 (Units)

The **Gantt chart** according to **Table 4.18** is shown in **Fig. 4.12**

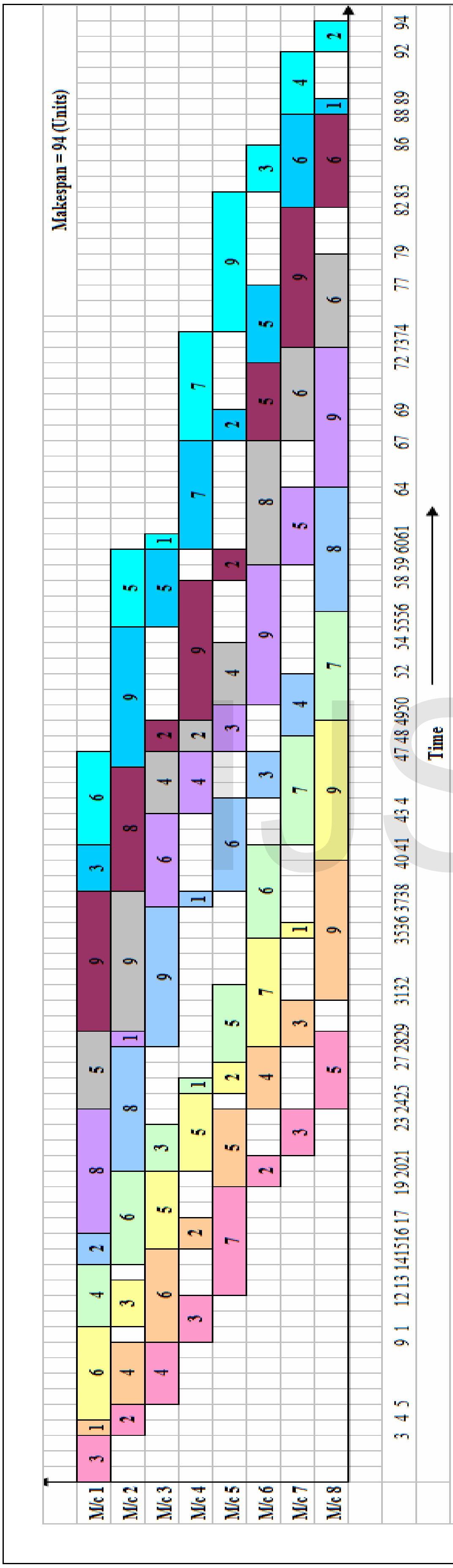


Fig. 4.11: Gantt Chart for 10-Jobs, 8-Machines Problem by Gupta's Heuristic Model

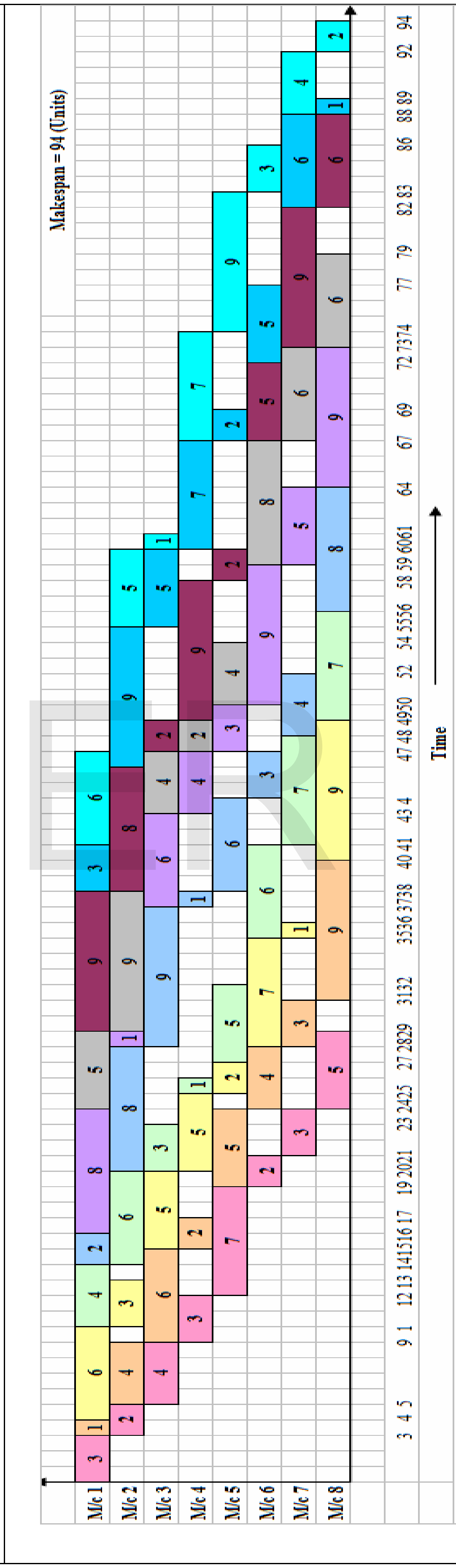


Fig. 4.12: Gantt Chart for 10-Jobs, 8-Machines Problem by CDS's Heuristic Model

### *4.12.3. For RA Heuristics:*

The solution for the above problem is constructed as follows:

| Job | $t_{i1}$' | $t_{i2}$' |
|---|---|---|
| **1** | 179 | 154 |
| **2** | 188 | 154 |
| **3** | 185 | 220 |
| **4** | 156 | 195 |
| **5** | 232 | 218 |
| **6** | 122 | 139 |
| **7** | 195 | 201 |
| **8** | 180 | 189 |
| **9** | 129 | 177 |
| **10** | 164 | 178 |

Optimal sequence is constructed on the basic of increasing order of $t_{i1}$' and $t_{i2}$'

| **Optimal sequence: 6-9-4-10-8-3-7-5-2-1** |
|---|

Thus total processing time can be calculated as:

**Table 4.19: Total Processing Time for 8-Machines, 10-Jobs by RA's Heuristic Model**

| Job i | M/c 1 | | M/c 2 | | M/c 3 | | M/c 4 | | M/c 5 | | M/c 6 | | M/c 7 | | M/c 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | | Time | | Time | | Time | | Time | | Time | | Time | | Time | |
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **6** | 0 | 3 | 3 | 5 | 5 | 9 | 9 | 12 | 12 | 19 | 19 | 21 | 21 | 24 | 24 | 29 |
| **9** | 3 | 4 | 5 | 9 | 9 | 15 | 15 | 17 | 19 | 24 | 24 | 28 | 28 | 31 | 31 | 40 |
| **4** | 4 | 8 | 9 | 15 | 15 | 18 | 18 | 19 | 24 | 29 | 29 | 35 | 35 | 42 | 42 | 49 |
| **10** | 8 | 14 | 15 | 18 | 18 | 23 | 23 | 28 | 29 | 31 | 35 | 42 | 42 | 43 | 49 | 58 |
| **8** | 14 | 16 | 18 | 26 | 26 | 35 | 35 | 36 | 36 | 42 | 42 | 45 | 45 | 49 | 58 | 66 |
| **3** | 16 | 24 | 26 | 27 | 35 | 41 | 41 | 45 | 45 | 48 | 48 | 57 | 57 | 62 | 66 | 75 |
| **7** | 24 | 29 | 29 | 38 | 41 | 45 | 45 | 47 | 48 | 52 | 57 | 65 | 65 | 71 | 75 | 81 |
| **5** | 29 | 38 | 38 | 46 | 46 | 48 | 48 | 57 | 57 | 59 | 65 | 70 | 71 | 80 | 81 | 87 |
| **2** | 38 | 41 | 46 | 55 | 55 | 60 | 60 | 67 | 67 | 69 | 70 | 75 | 80 | 86 | 87 | 88 |
| **1** | 41 | 47 | 55 | 60 | 60 | 61 | 67 | 74 | 74 | 83 | 83 | 86 | 86 | 90 | 90 | 92 |

Therefore, total processing time = **92 (Units)**

Total Idle Time for M/c 1 = 92-47 = 45 (Units)

Total Idle Time for M/c 2 = 3+2+(92-60) = 37 (Units)

Total Idle Time for M/c 3 = 5+3+1+7+(92-61) = 47 (Units)

Total Idle Time for M/c 4 = 9+3+1+4+7+5+1+3+(92-74) = 51 (Units)
Total Idle Time for M/c 5 = 12+5+3+5+8+5+(92-83) = 47 (Units)
Total Idle Time for M/c 6 = 19+3+1+3+8+(92-86) = 40 (Units)
Total Idle Time for M/c 7 = 21+4+4+2+8+3+(92-90) = 44 (Units)
Total Idle Time for M/c 8 = 24+2+2+2 = 30 (Units)

The **Gantt chart** according to **Table 4.19** is shown in **Fig. 4.13**

### *4.12.4. For Palmers Heuristics:*

The solution for the above problem is constructed as follows:

Slope = $(m-1)t_{j,m} + (m-3)t_{j,(m-1)} + (m-5)t_{j,(m-2)}$……………………………….

For 10-jobs and 8- machines:

$s_1 = (m-1)t_{1,8} + (m-3)t_{1,7}$…………………………………………+$(m-15)t_{1,1}$

For 8 machines (m=8)

= (8-1)*2+(8-3)*4+(8-5)*3+(8-7)*9+(8-9)*7+(8-11)*1+(8-13)*5 +(8-15)*6

= -25

Similarly

$s_2$ = -34
$s_3$ = 35
$s_4$ = 39
$s_5$ = -14
$s_6$ = 17
$s_7$ = 6
$s_8$ = 9
$s_9$ = 48
$s_{10}$ = 14

Optimal sequence is constructed on the basic of decreasing order of slope values

**Optimal sequence:**

$\mathbf{s_9 \geq s_4 \geq s_3 \geq s_6 \geq s_{10} \geq s_8 \geq s_7 \geq s_5 \geq s_1 \geq s_2}$

Thus total processing time can be calculated as:

**Table 4.20: Total Processing Time for 8-Machines, 10-Jobs by Palmer's Heuristic Model**

| Job i | M/c 1 | | M/c 2 | | M/c 3 | | M/c 4 | | M/c 5 | | M/c 6 | | M/c 7 | | M/c 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | | Time | | Time | | Time | | Time | | Time | | Time | | Time | |
| | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out | In | Out |
| **9** | 0 | 1 | 1 | 5 | 5 | 11 | 11 | 13 | 13 | 18 | 18 | 22 | 22 | 25 | 25 | 34 |
| **4** | 1 | 5 | 5 | 11 | 11 | 14 | 14 | 15 | 18 | 23 | 23 | 29 | 29 | 36 | 36 | 43 |
| **3** | 5 | 13 | 13 | 14 | 14 | 20 | 20 | 24 | 24 | 27 | 29 | 38 | 38 | 43 | 43 | 52 |
| **6** | 13 | 16 | 16 | 18 | 20 | 24 | 24 | 27 | 27 | 34 | 38 | 40 | 43 | 46 | 52 | 57 |
| **10** | 16 | 22 | 22 | 25 | 25 | 30 | 30 | 35 | 35 | 37 | 40 | 47 | 47 | 48 | 57 | 66 |
| **8** | 22 | 24 | 25 | 33 | 33 | 42 | 42 | 43 | 43 | 49 | 49 | 52 | 52 | 56 | 66 | 74 |
| **7** | 24 | 29 | 33 | 42 | 42 | 46 | 46 | 48 | 49 | 53 | 53 | 61 | 61 | 67 | 74 | 80 |
| **5** | 29 | 38 | 42 | 50 | 50 | 52 | 52 | 61 | 61 | 63 | 63 | 68 | 68 | 77 | 80 | 86 |
| **1** | 38 | 44 | 50 | 55 | 55 | 56 | 61 | 68 | 68 | 77 | 77 | 80 | 80 | 84 | 86 | 88 |
| **2** | 44 | 47 | 55 | 64 | 64 | 69 | 69 | 76 | 77 | 79 | 80 | 85 | 85 | 91 | 91 | 92 |

Therefore, total processing time = **92 (Units)**
Total Idle Time for M/c 1 = 92-47 = 45 (Units)
Total Idle Time for M/c 2 = 1+2+2+4+(92-64) = 37 (Units)
Total Idle Time for M/c 3 = 5+1+3+4+3+8+(92-69) = 47 (Units)
Total Idle Time for M/c 4 = 11+1+5+3+7+3+4+1+(92-76) = 51 (Units)
Total Idle Time for M/c 5 = 13+1+1+6+8+5+(92-79) = 47 (Units)
Total Idle Time for M/c 6 = 18+1+2+1+2+9+(92-85) = 40 (Units)
Total Idle Time for M/c 7 = 22+4+2+1+4+5+1+3+1+(92-91) = 44 (Units)
Total Idle Time for M/c 8 = 25+2+3 = 30 (Units)

The **Gantt chart** according to **Table 4.20** is shown in **Fig. 4.14**

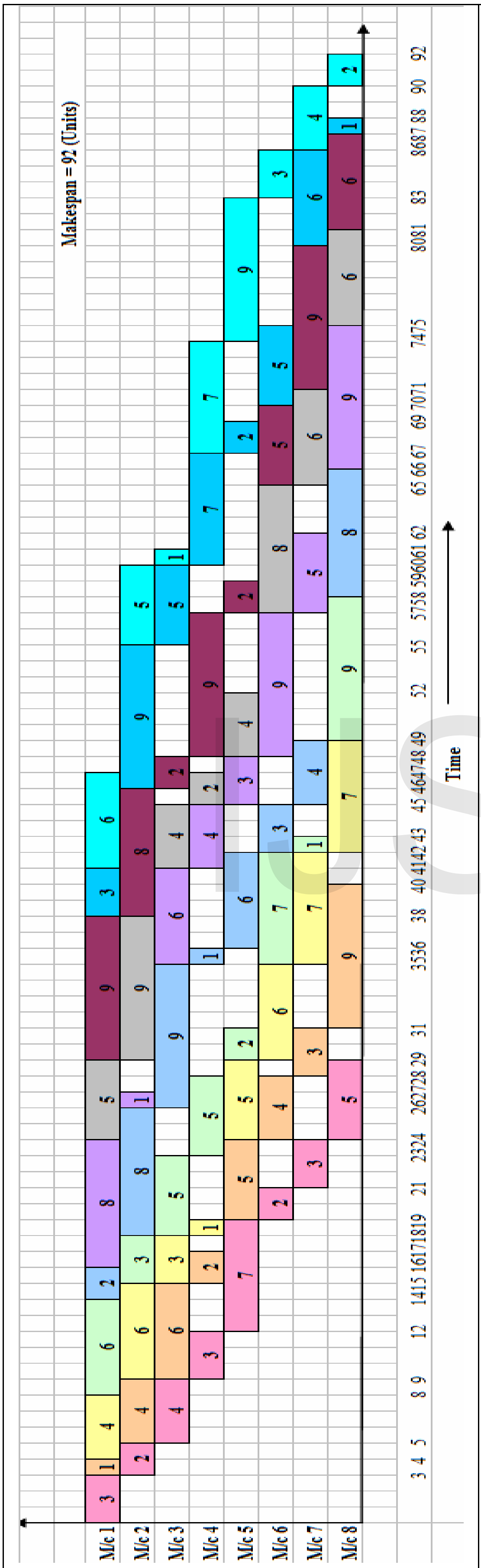


Fig..4.13: Gantt Chart for 10-Jobs, 8-Machines Problem by RA's Heuristic Model

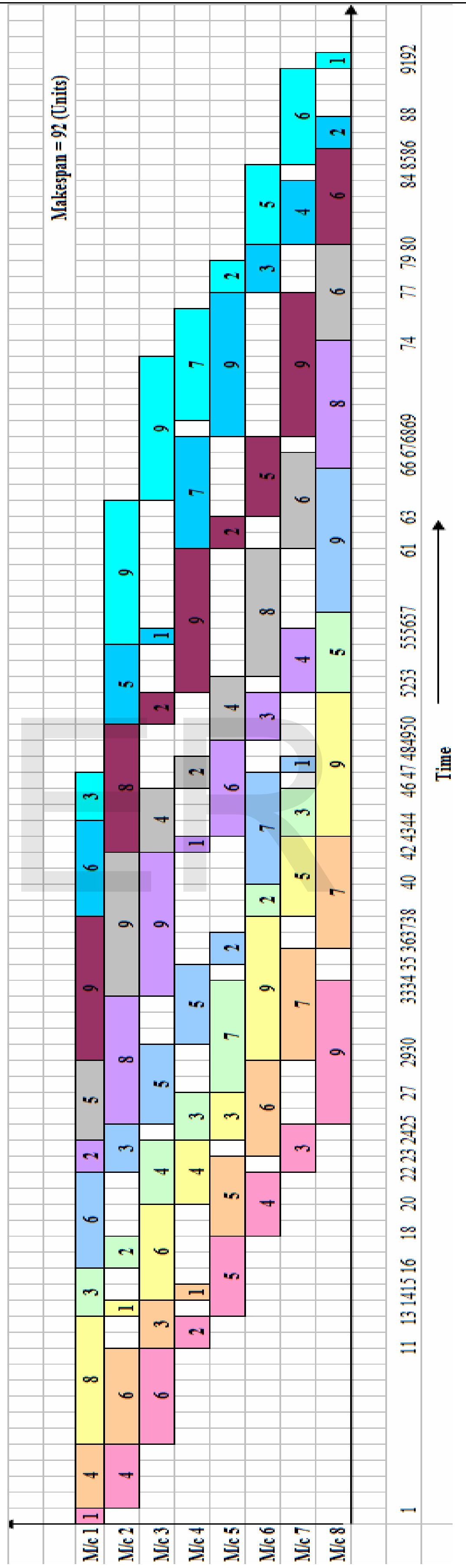


Fig..4.14: Gantt Chart for 10-Jobs, 8-Machines Problem by Palmer's Heuristic Model

**4.13 Scope of the Present Work**

Another important consideration is the choice of appropriate criteria for scheduling. Although the ultimate objective of any enterprise is to maximize the net present value of the shareholder wealth, this criterion does not easily lend itself to operational decision-making in scheduling. Some researchers are developing methodologies which take revenue and cost effects of schedules into consideration. Researchers and practitioners have so far used operational surrogates that influence costs and revenues. These include: number of parts tardy, average tardiness, weighted tardiness, throughput (this is a revenue influencing surrogate), as well as average number of parts in the system, machine utilization, and work-in- process inventory, for example. Analyses of these surrogates indicate that a scheduling procedure which does well for one criterion is not necessarily the best for some other. For example, attempts to reduce mean tardiness can lead to an increase in mean flow time. Minimizing makespan can result in higher mean flow time.

Further, a criterion which is appropriate at one level of decision-making may be unsuitable at another level. These raise further complications in the context of FMS because of the additional decision variables involved in including, for example, routing and sequencing alternatives. Scheduling may likely be a more complicated function when each part needs to visit several machines and when several operations have a choice of machines. The availability of alternative routing can improve system performance as well as increase scheduling complexities. In general, queuing network models have been used to address FMS design problems quantitatively and FMS planning problems qualitatively.



Queuing network models generally do not have sufficient Modelling capability to address detailed scheduling problems. Further research is required into using queuing models to address some scheduling problems.

**4.14 Summary**

Conventional methods of solving scheduling problems based on priority rules (FIFO, SPT, EDD…) determined the corresponding schedule but usually, still having idle times. To reduce these and improving CIM productivity optimization is necessary. Single factory production in traditional manufacturing has been gradually replaced by multi-factory production due to the trend of globalization. These factories may be geographically distributed in different locations, which allow them to be closer to their customers, to comply with the local laws, to focus on a few product types, to produce and market their products more effectively, and to be responsive to market changes more quickly.

Each factory is usually capable of manufacturing a variety of product types. Some may be unique in a particular factory, while some may not. In addition, they may have different production efficiency and various constraints depending on the machines, labor skills and education levels, labor cost, government policy, tax, nearby suppliers, transportation facilities, etc. This induces different operating costs, production lead time, customer service levels, etc. in different factories. The objective of this approach is to maximize the system efficiency by finding an optimal planning for a better collaboration among various processes.

# CHAPTER 5
## RESULTS AND DISCUSSION

### 5.1 Results of the Above Applied Heuristics Rules to Flowshop Scheduling

#### *5.1.1. For 2 X 8 Problems (2-Machines and 8-Jobs Problems):*

**Table 5.1: 2 X 8 Flowshop Problems**

| **Job *i*** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| $t_{i1}$ | 5 | 2 | 1 | 7 | 6 | 3 | 7 | 5 |
| $t_{i2}$ | 2 | 6 | 2 | 5 | 6 | 7 | 2 | 1 |

Makespan for the applied heuristics rules are:

| **Rule** | **Johnson's** | **Kusiak's** |
|---|---|---|
| **Makespan** | 37 Units | 37 Units |

#### *5.1.2. For 3 X 8 Problems (3-Machines and 8-Jobs Problems):*

**Table 5.1: 3 X 8 Flowshop Problems**

| **Job ⇨ / M/c ⇩** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| **1** | 5 | 2 | 1 | 7 | 6 | 3 | 7 | 5 |
| **2** | 2 | 6 | 2 | 5 | 6 | 7 | 2 | 1 |
| **3** | 3 | 4 | 6 | 2 | 1 | 5 | 4 | 7 |

Makespan for the applied heuristics rules are:

| **Rule** | **Gupta's** | **CDS** | **RA** | **Palmer's** |
|---|---|---|---|---|
| **Makespan** | 41 Units | 42 Units | 42 Units | 43 Units |

#### *5.1.3. For 10 X 10 Problems (10-Machines and 10-Jobs Problems):*

**Table 5.2: 10 X 10 Flowshop Problems**

| **Job ⇨ / M/c ⇩** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 5 | 2 | 1 | 7 | 6 | 3 | 7 | 5 | 7 | 4 |
| **2** | 2 | 6 | 2 | 5 | 6 | 7 | 2 | 1 | 8 | 3 |
| **3** | 3 | 4 | 2 | 6 | 1 | 5 | 4 | 7 | 6 | 5 |
| **4** | 5 | 2 | 1 | 3 | 8 | 2 | 6 | 1 | 9 | 8 |
| **5** | 7 | 6 | 3 | 2 | 6 | 2 | 5 | 7 | 1 | 3 |
| **6** | 9 | 2 | 7 | 3 | 4 | 1 | 5 | 3 | 8 | 1 |
| **7** | 7 | 5 | 2 | 2 | 3 | 5 | 1 | 6 | 2 | 3 |
| **8** | 8 | 2 | 5 | 4 | 9 | 3 | 2 | 6 | 1 | 8 |
| **9** | 2 | 6 | 4 | 2 | 6 | 2 | 5 | 2 | 6 | 3 |
| **10** | 7 | 1 | 4 | 2 | 4 | 6 | 2 | 2 | 6 | 7 |

Makespan for the applied heuristics rules are:

| Rule | Gupta's | CDS | RA | Palmer's |
|---|---|---|---|---|
| **Makespan** | 103 Units | 98 Units | 99 units | 99 Units |

### *5.1.4. For 8 X 10 Problems (8-Machines and 10-Jobs Problems):*

**Table 5.3: 8 X 10 Flowshop Problems**

| Job ⇨ M/c⇩ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 6 | 3 | 8 | 4 | 9 | 3 | 5 | 2 | 1 | 6 |
| **2** | 5 | 9 | 1 | 6 | 8 | 2 | 9 | 8 | 4 | 3 |
| **3** | 1 | 5 | 6 | 3 | 2 | 4 | 4 | 9 | 6 | 5 |
| **4** | 7 | 7 | 4 | 1 | 9 | 3 | 2 | 1 | 2 | 5 |
| **5** | 9 | 2 | 3 | 5 | 2 | 7 | 4 | 6 | 5 | 2 |
| **6** | 3 | 5 | 9 | 6 | 5 | 2 | 8 | 3 | 4 | 7 |
| **7** | 4 | 6 | 5 | 7 | 9 | 3 | 6 | 4 | 3 | 1 |
| **8** | 2 | 1 | 9 | 7 | 6 | 5 | 6 | 8 | 9 | 9 |

Makespan for the applied heuristics rules are:

| Rule | Gupta's | CDS | RA | Palmer's |
|---|---|---|---|---|
| **Makespan** | 94 Units | 94 Units | 92 units | 92 Units |

"**Makespan** is the time length from the starting of the first operation of the first demand to the finishing of the last operation of the last demand."

# CHAPTER 6
## CONCLUSION AND FUTURE SCOPE

### 6.1 Conclusion

By Scheduling, we assign a particular time for completing a particular job. The main objective of scheduling is to arrive at a position where we will get minimum processing time. The problem examined here is the n-job, m-machine problem in a flowshop. This work arranges the jobs in a particular order and gets many combinations and chooses that combination where we get the minimum makespan. This study try to solve the problem of a flowshop scheduling with the objective of minimizing the makes pan. Here the objective is to minimize the makespan of batch-processing machines in a flowshop. Comparison based on Gupta's heuristics, RA heuristic's, Palmer's heuristics, CDS heuristics are proposed here. Analytic solutions in all the heuristics are investigated. Gantt chart is generated to verify the effectiveness of the proposed approaches. As a result of the work proposed here the researcher found that out of the four proposed algorithms Gupta, CDS and RA heuristics yields more efficient results but comparatively and practically RA heuristics yields most efficient and better results because here the processing times are determined from a weighting scheme. The main advantage of RA heuristics for yielding the better result as compared to others is that, it is the combination of two heuristics approaches (Palmers slope index + CDS method) and is least biased and best operated of the heuristics and the CDS, Gupta, Palmer algorithm comes next. Here in this work generally RA heuristics yields efficient result, these have been explained with the help of numerical examples and their performances are examined with the help of Gantt charts.

The Gantt schedule can illustrate the relationship between work activities having duration, events without duration that indicate a significant completion, that represent major achievements or decision points. The researchers proposed the comparison between four heuristics approaches and tell which is more efficient. For a given task, the order in which tasks are considered and the criteria by which machine centers are selected play major roles for optimizing the scheduling policies. The model introduced here proposed a new approach for planning scheduling problems - providing a way to optimize the makespan. Scheduling is an activity to select the right future operational program or diagram of an actual time plan for allocating competitive different demands of different products, delivery dates, by sequencing through different machines, operations, and routings for the combination of the high flexibility of Jobshop type with high productivity of flow-shop type and meeting delivery dates. The researcher has provided a technique for processing information that is at once elegant and versatile.

### 6.2 Scope for Future Work

Further research may be conducted to investigate the applications of other meta-heuristics to the lot-streaming flowshop problem. It is also worthwhile to design other versions of RA heuristics to continue pursuing the best performance of RA heuristics. Future research should address problems with different shop environments, including parallel machines flowshop, Jobshop, and open shop. Problems with other performance measures, such as minimum due dates, maximum lateness, and multi-criteria measures should also be studied. Future research should be directed to generalize the method to multipart, multi machine group cases.

# REFERENCES

1) I.Drstvensek, I. Pahole, J. Balic, "A model of data flow in lower CIM levels", Journal of Materials Processing Technology 157-158 (2004)
2) Ezedeen Kodeekha," a new method of FMS scheduling using optimization and simulation", Department of Production, Informatics, Management and Control Faculty of Mechanical Engineering Budapest University of Technology and Economics
3) Clarence H Martin, "A hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming", Omega 37 (2009) 126 – 137
4) Chin-Chia Wu, Wen-Chiung Lee, "A note on the total completion time problem in a permutation flowshop with a learning effect", European Journal of Operational Research 192 (2009) 343–347
5) Christos Koulamas, George J. Kyparisis, "Single-machine scheduling with waiting-time-dependent due dates", European Journal of Operational Research 191 (2008) 577–581
6) Felix T.S. Chan, S.H. Chung, L.Y. Chan, G. Finke, M.K. Tiwari ,"Solving distributed FMS scheduling problems subject to maintenance: Genetic algorithms approach", Robotics and Computer-Integrated Manufacturing 22 (2006) 493-504
7) Safia Kedad-Sidhoum, Yasmin Rios Solis, Francis Sourd, "Lower bounds for the earliness–tardiness scheduling problem on parallel machines with distinct due dates", European Journal of Operational Research 189 (2008) 1305-1316
8) Moosa Sharafali, Henry C. Co, Mark Goh, "Production scheduling in a flexible manufacturing system under random demand", European Journal of Operational Research 158 (2004) 89-102
9) K.H. Ecker, J.N.D. Gupta, "Scheduling tasks on a flexible manufacturing machine to minimize tool change delays" , European Journal of Operational Research 164 (2005) 627-638
10) Xiaoping Li, QianWang, ChengWu, "Efficient composite heuristics for total flow time minimization in permutation flowshops", Omega 37 (2009) 155- 164
11) Kanchan Das, M.F. Baki, Xiangyong Li, "Optimization of operation and changeover time for production planning and scheduling in a flexible manufacturing system", Computers & Industrial Engineering xxx (2008) xxx-xxx
12) Bo Chen, Chung-Yee Lee, "Logistics scheduling with batching and transportation", European Journal of Operational Research 189 (2008) 871-876
13) Konstantinos N. Androutsopoulos, Konstantinos G. Zografos, "Solving the multi-criteria time-dependent routing and scheduling problem in a multimodal fixed scheduled network", European Journal of Operational Research 192 (2009) 18-28
14) Nadia Hamani**,** Nathalie Dangoumau, Etienne Craye, "Reactive mode handling of flexible manufacturing systems", Mathematics and Computers in Simulation xxx (2008) xxx-xxx

15) Tiente Hsu, Ouajdi Korbaa, Re´my Dupas, Gilles Goncalves, "Cyclic scheduling for FMS: Modelling and evolutionary solving approach", European Journal of Operational Research 191 (2008) 464-484
16) T.C.E. Cheng, C.T. Ng, J.J. Yuan, "Single-machine scheduling of multi-operation jobs without missing operations to minimize the total completion time", European Journal of Operational Research 191 (2008) 320-331
17) Ruud Teunter, Konstantinos Kaparis, Ou Tang, "Multi-product economic lot scheduling problem with separate production lines for manufacturing and remanufacturing", European Journal of Operational Research 191 (2008) 1241-1253
18) Lixin Tang, Hua Gong, "A hybrid two-stage transportation and batch scheduling problem", Applied Mathematical Modelling 32 (2008) 2467-2479
19) Chao-Tang Tseng, Ching-Jong Liao, "A discrete particle swarm optimization for lot-streaming flowshop scheduling problem", European Journal of Operational Research 191 (2008) 360-373
20) Pei Chann Chang, Shih Hsin Chen, V. Mani, "A hybrid genetic algorithm with dominance properties for single machine scheduling with dependent penalties" , Applied Mathematical Modelling 33 (2009) 579-596
21) T.C.E. Cheng, B.M.T. Lin, "Johnson's rule, composite jobs and therelocation problem", European Journal of Operational Research 192 (2009) 1008-1013
22) Seong-Jong Joo, "Scheduling preventive maintenance for modular designed components: A dynamic approach", European Journal of Operational Research 192 (2009) 512-520
23) Ruslan Sadykov, "A branch-and-check algorithm for minimizing the weighted number of late jobs on a single machine with release dates", European Journal of Operational Research 189 (2008) 1284-1304
24) Xianyi Wu, Xian Zhou, "Stochastic scheduling to minimize expected maximum lateness", European Journal of Operational Research 190 (2008) 103-115
25) Andrea Rossi, Gino Dini, "Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimization method" Robotics and Computer-Integrated Manufacturing 23 (2007) 503-516
26) Shi-jin Wang, Li-feng Xi, Bing-hai Zhou, "FBS-enhanced agent-based dynamic scheduling in FMS", Engineering Applications of Artificial Intelligence 21 (2008) 644-657
27) J.F. Goncalves, J.J.M. Mendes, M.G.C. Resende, "A genetic algorithm for the resource constrained multi-project scheduling problem", European Journal of Operational Research 189 (2008) 1171-1190
28) Yaohua He, Chi-Wai Hui, "A rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units", Computers and Chemical Engineering 32 (2008) 3067-3083

29) Jiaqiong Chen, Ronald G. Askin, “Project selection, scheduling and resource allocation with time dependent returns”, European Journal of Operational Research 193 (2009) 23-34
30) Adam Janiak, Tomasz Krysiak, Costas P. Pappis, Theodore G. Voutsinas, “A scheduling problem with job values given as a power function of their completion times”, European Journal of Operational Research 193 (2009) 836-848
31) Tamer Eren, Ertan Gu¨ner, “A bicriteria flowshop scheduling with a learning effect” , Applied Mathematical Modelling 32 (2008) 1719-1733
32) Xianyi Wu, Xian Zhou, “Stochastic scheduling to minimize expected maximum lateness”, European Journal of Operational Research 190 (2008) 103-115
33) Bibo Yang, Joseph Geunes, “Predictive–reactive scheduling on a single resource with uncertain future jobs”, European Journal of Operational Research 189 (2008) 1267-1283
34) Gur Mosheiov, Assaf Sarig, “Due-date assignment on uniform machines”, European Journal of Operational Research 193 (2009) 49-58