A New Proposed Heuristic for Job Shop and Flow Shop Scheduling Problem

Author

MUHAMMAD HAMID

Regn Number

00000172533

Supervisor

MUHAMMAD HAMID

DEPARTMENT OF DESIGN AND MANUFACTURING ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

JANUARY 2018

A New Proposed Heuristic for Job Shop and Flow Shop Scheduling Problem

Author

MUHAMMAD HAMID

Regn Number

00000172533

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Design and Manufacturing Engineering

Thesis Supervisor:

DR MUSHTAQ KHAN

Thesis Supervisor's Signature:

____

DEPARTMENT OF DESIGN AND MANUFACTURING ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

JANUARY, 2018

Declaration

I certify that this research work titled "*A New Proposed Heuristic for Job Shop and Flow Shop*

Scheduling Problem" **is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.**

Signature of Student

MUHAMMAD HAMID

00000172533

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

MUHAMMAD HAMID

Registration Number

00000172533

Signature of Supervisor

Copyright Statement

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed, I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor DR MUSHTAQ KHAN for his help throughout my thesis and for Laser Material Processing (LMP) and Rapid Prototyping, Tooling and Manufacturing (RP, T&M) courses which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to ENGR ALTAF UR REHMAN for his tremendous support and cooperation. Each time I got stuck in something, he came up with the solution. Without his help, I wouldn’t have been able to complete my thesis. I appreciate his patience and guidance throughout the whole thesis.

I would also like to thank DR LIAQAT ALI, DR SHAHID IKRAMULLAH BUTT and DR SYED HUSAIN IMRAN for being on my thesis guidance and evaluation committee and express my special Thanks to Mr. Shahzaib Iqbal for his help. I am also thankful to Muhammad Luqman, Nabeel Mushtaq, Muhammad Athar Ali, Ahmad Faraz and Saad Ali for their support and cooperation.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

Scheduling is an essential and critical component of all manufacturing processes that has a major impact on productivity and efficiency of a firm. It plays a vital role in the optimization of the manufacturing times and costs that ultimately results in energy efficient processes. The optimization of job-shop and flow shop process scheduling problem is still a challenge to researchers and is far from being completely solved due to its combinatorial nature. It has been estimated that more than 75% of manufacturing processes occur in small batches. In such environments, processes must be able to perform a variety of operations on a mix of different batches. Job shop and flow shop scheduling optimization is the response to such low batch manufacturing problems. In this research, a novel proposed heuristic (P.H) solution approach for job shop and flow shop scheduling problem is presented with the objective of optimizing the overall Make span (C_{max}). The proposed P.H is the combination of Longest Processing Time (LPT) and Shortest Processing Time (SPT) process schedule. It is composed of eight sequences, eight Makespan values are calculated and then minimum of these is selected as a final Makespan. The proposed P.H is explained with the help of a detailed example. Comparative analysis tables are constructed for distinct set of benchmark problems from the literature to check the validity and effectiveness of the proposed heuristic. The presented P.H has achieved batch-job process schedules that have outperformed the traditional heuristics. The results are encouraging and show that the proposed heuristic is a valid methodology for batch process and job shop scheduling optimization.

Key Words: ***Process Scheduling, Optimization, Job Shop, Flow Shop, Makespan, Proposed Heuristic (P.H), Shortest Processing Time (SPT), Longest Processing Time (LPT)***

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

The customer demand for various products with shorter product life, lead time is the need of every organization. The competitive pressure of producing products with high quality, low cost now becomes principal factor to capture the market. To obtain the above objectives there is a need of zero inventory philosophy. The market share will help the manufacturing enterprises to be fast enough to respond quickly to changing demand and requirement.

The engineer must develop a production schedule to minimize the product completion times and on-time delivery. The significance of this issue today has turned out to be extremely noteworthy because of the undeniably focused nature of worldwide economies. All these can be obtained efficiently with the method and techniques of scheduling.

The scheduling process frequently emerges in a circumstance where resources availability is basically and essentially fixed by long term commitments of a prior planning decision. In the view of this circumstance, production scheduling and sequencing is important activities in production planning and control.

The research work in this dissertation has been presented. It is related to the detailed study of traditional heuristic i.e. CR (Critical Ratio), EDD (Earliest Due Date), SPT (Shortest Processing Time), LPT (Longest Processing Time), WSPT (Weighted Shortest Processing Time) etc. The objective of this study is to understand all the traditional heuristics and develop a new combined heuristic for flow and job shop scheduling problem to minimize the make span of these problems.

1.1 Scheduling in Manufacturing Industries

A present-day complex manufacturing setting comprises of multiple production lines, each requiring various resources such as steps and machines for completion. Subsequently, the manufacturing engineer must manage the resource in most economical and optimal manner to produce the product efficiently [1]. In all manufacturing industries scheduling become one of the most important and difficult problem because time is taken

by each activity to accomplish it. Scheduling is often failed due to the lack of addressing problems of scheduling in an questionable domain, for example a long queue time. To find the solution of scheduling problems industries use a combination of different heuristics and simulation to find the optimal or near optimal solution.

1.2 Scheduling vs Sequencing

Another terminology which is normally comes with scheduling is sequencing. Sequencing is based on the operation and assembly precedence with no involvement of time but it only tells about the order of activities that is to be carried out. On the other hand, scheduling is the sequencing of task based on involvement of time. The time consists of setup time and machining time.

1.3 Scheduling Definition

Scheduling consists of choosing, organizing, and timing available for resources to carry out all the activities, which are important to produce desired part in the desired time while satisfying many relationship and time constraints among the available resources, and activities. Scheduling is necessarily concerned with the constrained optimization problem and in the context of manufacturing dealing with finding a sequential allowance of competing resources that optimize an objective function. The use of scheduling in manufacturing environment is to minimize the total production time, cost by asking the question of when to make, with which machine or equipment. Hence, scheduling is necessary to maximize the efficiency of operations and reducing cost.

1.4 Types of Scheduling

There are distinct types of scheduling, which are as follow.

1.4.1 Open Shop Scheduling

In an open shop scheduling, jobs have no fixed sequence of operations and the job can follow any of the sequence. Generally, in an open shop is one, which fabricates the final product other than directly to demands. When there is much demand for the same product

form many customers, then it is necessary to maintain a large inventory in the warehouse to change the jobs meant for one customer to other of high priority.

1.4.2 Flow Shop Scheduling

In flow shop scheduling, jobs have the same sequence of the operations and every job in the sequence can move to every machine only one time e.g. car assembly operation. In flow shop all the jobs have the same processing times on all the machines and the jobs passes between the machines in the same order. There is a predefine constraints for every job in the flow shop which differs it from the job shop.

1.4.3 Job Shop Scheduling

Job shop scheduling is a classical combinational optimization problem [2]. In the job shop, scheduling every job has its own sequence of operation that may or may not be different from other job e.g. Cement manufacturing.

According to Jain and Meeran, there are several techniques to solve the scheduling problem [3, 4, 5, 6, 7, 8, 9, 10, 11]. Some of the most common listed are genetic algorithms techniques, artificial intelligence and heuristic rules etc. Heuristics are the techniques that are developed to find the problem nature, to find the optimal solution but not guarantee about its optimality. Heuristic rules are also called dispatching rules or priority rules used to find the next job in the queue that is to be processed on the machine. Some of the most common dispatching rules are First in First out (FIFO), Shortest Processing Time (SPT), Earliest Due Date (EDD), Longest Processing Time (LPT), Earliest Completion Time (ECT) etc.

1.5 The Research Problem

In a Job Shop Scheduling Problem (JSSP), there exists n x m jobs means that we have n number of jobs that is to be processed on m number of machines. The job shop scheduling problem tries to find the order of different operation on the job that minimize the total completion time.

For a job shop scheduling problem, the following assumption is to be made:

i) At the start of the operation on the machines i.e.at time zero (0) a set of n number of jobs are to be present.

ii) One job is to be processed on machine at a time.

iii) Every job must have a well-known operation time.

iv) The running operations are to be continues processed until it is to be completed.

v) There are no precedence constraints between different jobs.

vi) There is no description of due dates and release times.

For every job in a JSSP there is a specific set of operation and a predetermined set of operations on machines. Each job operation must be fixed which is different from other job and the machine sequence. In a manufacturing system, a JSSP deals with several other traditional optimization methods and has proven to be the hardest optimization problem. The reason is that its scope is very large and wide. For example, for 20 x20 problem (20 jobs and 20 machines) there are "n" power "m" alternative solution means that there are a trillion-different solution exists, now the objective is to find the most optimal solution among all them. Fast computer could take a thousand of year to find the optimal solution that's why the job shop scheduling problem is considered as one of the hardest optimizing approach. Though in the existing methods and techniques there are chances of further improvement.

1.6 The Research Objective

There are many objectives of this research but the main objective is to develop a new heuristic using different dispatching/priority rules for solving the job shop scheduling problem to find the near optimal result that minimize the total completion time (Makespan). Mostly the objectives of this research are

i) To obtain the contemporary knowledge about scheduling, a detailed study of scheduling problem and solution approaches is carried out.

ii) To study the existing techniques and method that was already developed and how it can be solved e.g. Palmer heuristics, shifting bottleneck, branch and bound heuristics etc. to find the satisfactory approaches after comparing their effectiveness

and efficiency with the new developed heuristics. The literature review will also help to find the research area to be focused upon.

iii) To develop a new heuristic, which minimizes the Makespan

iv) To find the strength of the new proposed heuristics and compare it with the traditional heuristics.

v) To determine the future work.

1.7 Organization of Thesis

This thesis consists of five chapters. In chapter 1, introduction to scheduling, problem of the research, objective of the research is presented. In chapter 2 a detailed study of scheduling is carried out including different scheduling rules and scheduling environment. The literature review of different scheduling methods is studied. The main goal of this chapter is to review different scheduling heuristics in literature. Chapter 3 covers the complete detail of the proposed heuristics and its methodology. Chapter 4 consists of results and discussion and it consists of comparison of proposed heuristics with the traditional heuristics to find its strength. In the closing chapter 5 concludes the whole thesis and gives the recommendation and future work.

In this chapter, a brief introduction of scheduling is presented also with the research problem statement and research objective. The next chapter includes a detailed introduction to scheduling, scheduling rules, and different scheduling environments.

CHAPTER 2: LITERATURE REVIEW

Scheduling is a process of identifying the time when an activity is to be start and when it is to be finished. As every activity in the process is somewhat in competition with the other activities due to the limitation of resources like time and capacity. So, scheduling has a very importance in the management of resources. Proper scheduling is required to complete the orders on time, maximize the use of resources, and minimizing idle time [12].

Scheduling is a task of assigning resources over a period to the process to perform a collection of different activities. From the above definitions, it is understood that scheduling process consists of two things that is first scheduling is a decision-making process which means how decision regarding each activity in the process is to be schedule and the second thing about scheduling is that it is a complex process which consists of rules, principles, models, and techniques.

When the word scheduling is used the word, sequencing comes in mind so, it is important to differentiate the difference between the two. According to Ibrahim M. Alharkan "Scheduling is defined as assigning each activity operation of each job a start and a completion time on the time scale of machine within the precedence relation" [13]. While on the other hand, sequencing means for each machine in the shop it needs to define the order in which each job in the shop is to be waiting in front of required machine.

The need that energizes the scheduling concepts is that let we have m number of jobs that is to be processed on n machines. Every job in process has it given sequence of operation which needs to be processed using the required sequence of machines. Every operation in the process needs a particular machine at a given time. While every machine in the order can be processed once per operation at a given time so, the orders of operations on each machine that result in minimizing the total cost need to be clear.

The scheduling technique's mostly used use in the cases where resource availability is especially constant by long term commitment before the planning decision process. The scheduling decision reaches to system approaches by describing the following steps. These steps are formulation, analysis, synthesis, and evaluation. In the first step, problem is identified and according to be criteria being defined the problem is formulated. In the next step, a detail analysis of the problem is carried out. Analysis is a more detail process of

examining the highlighted features of the problem. In the synthesis stage, alternative solution to the problem is built while at the last all the alternatives are evaluated with the alternative with best results are selected.

2.1 Definitions

Sequencing and scheduling is a standout amongst the most imperative strategies in manufacturing planning and control. Morton and Pentico discussed the importance of sequencing and scheduling, stating that "it pervades all economic activity" [14].

According to Baker, "Sequencing and scheduling are forms of decision-making which play a crucial role in manufacturing as well as in service industries" [15]. In the current focused and competitive condition, effective and efficient sequencing and scheduling has turned into a need for survival in the commercial and global marketplace. Organization must meet transportation dates committed to the clients, as inability to do as such may bring about a significant loss of good will, cooperative attitude. They also have to schedule activities in such a way as to use the resources available in an efficient manner and its full utilization of resources.

The definition of sequencing among researchers is common. Sequencing is defined as the order in which the jobs (tasks) are processed through the machines (resources).

According to Baker, "Scheduling is a decision-making function and the allocation of resources over time to perform a collection of tasks. It determines the process of schedule and body of theory. In addition to this, it is the collection of principles, models, techniques, and logical conclusions that provide insight into the scheduling function" [15].

According to David Pentico, "Scheduling is the process of organizing, choosing, and timing resource usage to carry out all the activities necessary to produce the desired outputs at the desired times, while satisfying many time and relationship constraints among the activities and the resources" [14].

The general definition given by Alharkan to define scheduling is, "There are M machines {M1, M2... M_m} available and N jobs {J1, J2... J_n} to be processed. To complete the processing of each job a subset of these machines is required [13]". Process plan or flow pattern for some or all jobs is not necessarily to be fixed. Each job must be processed via the machines in a sequence that satisfies the job's technological and precedence constraints.

If job i is processed on machine j then it is called an operation denoted by O_{ij}. It is associated with a processing time denoted by P_{ij}, and a setup time denoted by S_{ij}. Moreover, it is associated with a weight w_i, a ready (release or arrival) time, and a due date, di. Finally, each job has an allowance time to be in the shop.

2.2 Level of Scheduling

There are distinct levels of scheduling and sequencing according to Baker [15]. These are as follows.

2.2.1 Long Term Scheduling

The long scheduling has a long-time span which comprise of 2 to 5 years of time horizon. This kind of planning is better for planning process such as layout planning, plant design planning etc.

2.2.2 Middle Term Scheduling

The middle term scheduling planning has the time span of 1 to 2 years. This kind of planning is better for production smoothing, logistics etc.

2.2.3 Short Term Scheduling

The short terms scheduling planning has a time span of 3 to 6 months. This kind of planning is better for requirement planning, shop bidding, and due date setting etc.

Table 2-1: Classification of Scheduling Levels

Level	Examples	Time span
Long range planning	Plant layout, plant design, plant expansion	2-5 years
Middle range planning	Production smoothing and logistics	1-2 years
Short range planning	Shop bidding, due date setting	3-6 months
Predictive planning	Job shop routing, process batch sizing	2-6 weeks
Reactive planning	Late Material, down machines	1-3 days

2.2.4 Predictive Scheduling

This kind of planning is done for the period up to 2 to 6 weeks and is effective for Job shop routing, assembly line balancing, and process batch sizing.

2.2.5 Reactive Scheduling

The reactive scheduling is done on day-to-day basis or every 3 days and are effective for jobs, down machines, and late material.

2.3 Scheduling Environment

Conway, Maxwell, and Miller divided the scheduling and sequencing environment into four types [16]. These are,

i) The work part and its operation to be work on.

ii) The number and types of machines that contains the shop.

iii) The disciplines that restrict the way job is to be loaded can be made.

iv) The processing criteria by which a schedule will be selected.

The different scheduling environments are discussed below.

2.3.1 Open Shop

In the open shop environment, there is no restriction on the flow of jobs on the n number of machines. Simply there is no well defining flow path for the jobs. Generally, in an open shop is one in which fabricates the final product other than directly to demands. When there is much demand for the same product form many customers, then it is necessary to maintain a large inventory in the warehouse to change the jobs meant for one customer to other of high priority.

2.3.2 Batch Shop

A batch shop is an open shop for which duplication in work in process and final production between customers becomes so large such that a high batch processing is take advantage of economies of scale in processing similar parts. In a batch shop the flow of parts are necessarily in a straight line. An example of discrete batch shop is the garments industries and for a continuous batch shop is the oil refinery, chemical industries.

2.3.3 Single Machine Shop

In a single machine shop, there is m number of jobs and that is to be processed on a single machine. The following assumption must be made to develop a single machine shop models.

- **i) In all the scheduling time, the machine will be continuously available.**
- **ii) The machine will process the part one at a time.**
- **iii) Processing time of all the jobs on the machines will be well defined.**
- **iv) Setup and machining time are included in the process time.**
- **v) The jobs would be processed without an interruption in a non-preemptive case.**
- **vi) Information related to other jobs including due date, release time will be available timely.**

2.3.4 Job Shop

In a job shop environment, every job must follow its flow sequence (process plan) and a subset of these jobs can go through each machine two or more times. In simple, there are multiple inputs and outputs. The job shop composed of "M" machines and "N" jobs and the jobs have a known processing order and time. All the possible sequencing for a job shop consists of n power m. The job shop problems are complex and were unsolved for years.

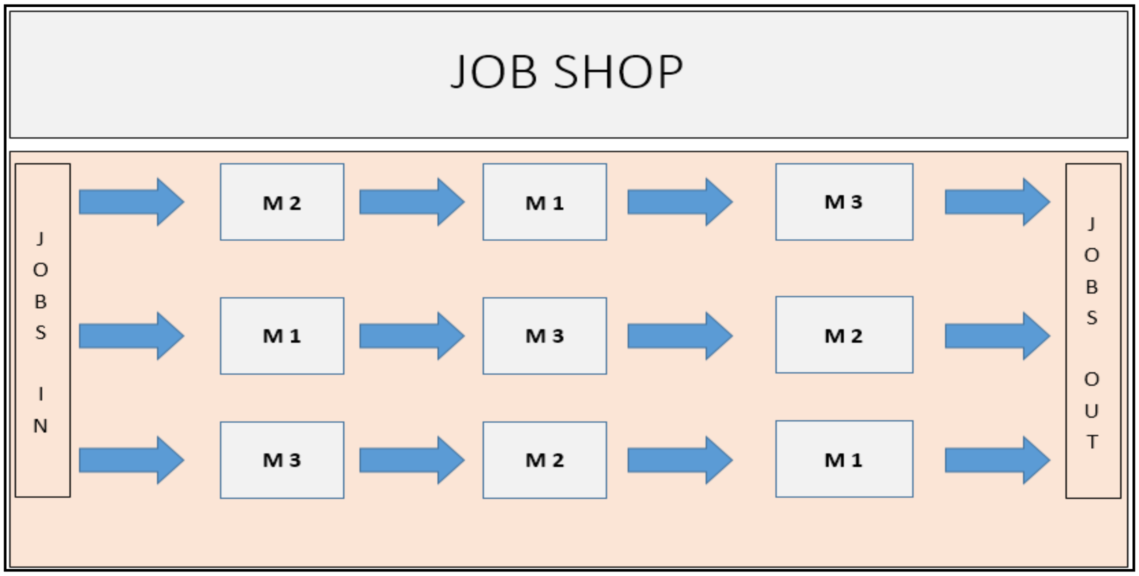

Figure 2.1: Job Shop Environment

2.3.5 Closed Shop

A closed shop resembles a job shop in which the jobs are restricted to its own flow pattern and production is not change with customer demand. It is a job shop with all the jobs has its specified flow pattern which may or may not be different from other jobs in the sequence of its flow.

2.3.6 Flow Shop

In flow shop environment, there is "N" number of machines in series and every job can follow the same sequence of operation. The jobs must be processed in the following way. All the jobs have the same processing plans in flow shop, on all the machines and the jobs passes between the machines in the same order. There is a predefine constraints for every job in the flow shop which different it from the job shop.

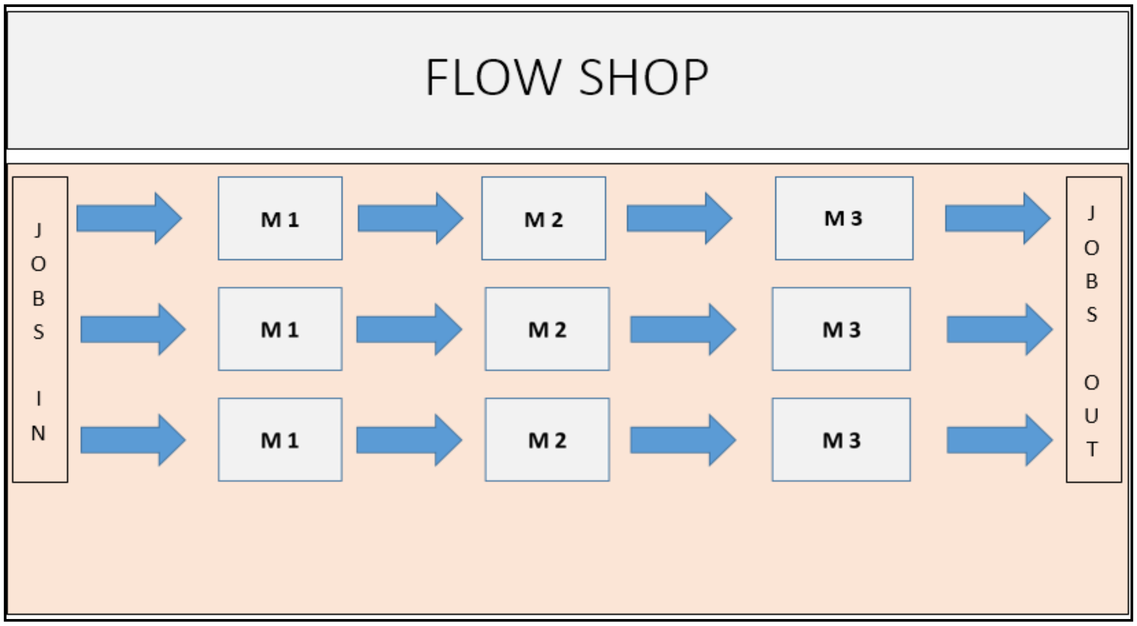

Figure 2.2: Flow shop environment

2.3.6.1 Permutation

In a permutation flow-shop environment, jobs are processed on n number of machines. The machines are arranged in serial sequence. Permutation is the case in which jobs is to be processed on the machines in the order of sequence in which it is predefined from the start with no job to be processed other than that sequence.

2.3.6.2 Non-Permutation

In non-permutation flow-shop environment, Jobs are not processed in the same order. The job with high priority in the above order is to be selected from the queue and is to be process first from the other jobs in the queue. The reason may be the due date of some jobs or some other importance that is why it must be worked out earlier.

2.3.7 Assembly Job Shop

In assembly shop is a job shop environment with a least two parts and that is be assemble. An assembly job shop is like an open shop or batch shop, which assembles to subassemblies which in turns assembles another subassembly and finally produces the final product. The assembly job shop is mostly occurring in form of long straight lines in which the entry of parts comes from the sides of the lines. Some other assembly shops are not

linear but rather they are very complex. An example of the assembly job shop is the production of airbuses and car manufacturing.

2.3.8 Hybrid Job Shop

In hybrid job shop environment, it involves the precedence ordering of operations of some work parts is same. Sometimes there exist some jobs which has operations which is common in many jobs that operations are to be noted because that can save a lot amount of time, energy and ultimately cost. Hybrid shop mostly occurs in the industries, which can update its process with new available information the passage of time. Example of hybrid job shop is the enterprise, which produces different automobiles.

2.3.9 Hybrid Assembly Job Shop

This shop environment, the characteristics of both the assembly job shop and hybrid job shop are combined. This job shop takes the advantage of both. It is a compact from of assembly and hybrid shop which is very efficient in terms of machines utilization and cost effective.

2.3.10 Transfer Line

A transfer line consists of a large volume and a low variety production line, which is generally in a straight line. The transfer line itself is often automated. The transfer line is used for carrying parts from station to station. There is no buffer zone along the line rather than at the beginning. Sometimes the flow sides of some lines consist of two to three other transfers, which end at a single line at last. Transfer lines are generally used for carrying parts from one station to other station.

2.3.11 Flexible Transfer Line

A flexible transfer line is used for the same function as the transfer line but instead of a linear the line can be changed to a variety of shapes and thus the jobs can follow several ways. This line provides a greater diversity of product to be accomplished at a smaller volume for each. Simply it's very flexible in terms of any changes in the line.

Table 2-2: Scheduling environments

S. No	Type	Characteristics

1	**job shop**	**Discrete, complex flow, unique jobs, no multiuse parts.**
2	**Open shop**	**Discrete, complex flow, some repetitive jobs, multiuse parts.**
3	**Batch shop**	**Discrete or continuous, less complex flow, many repetitive and multiple use parts.**
4	**Flow shop**	**Discrete or continuous, linear flow, large continuous batch process.**
5	**Assembly shop**	**Assembly version of open job shop or batch shop.**
6	**Assembly line**	**High volume, low variety, transfer line of assembly shop.**
7	**Transfer line**	**Very high volume and low variety linear production facility with automated operations.**
8	**Flexible transfer line**	**Modern version of cells and transfer lines.**

2.4 Dispatching Rules

The dispatching rules (also called sequencing, scheduling, priority, and decision rule) are the rules that determine the priority of each job according to some traditional rules. This priority of job is determined as a function of machine parameter, job parameter, or shop characteristics. When the sequence each job is determined, jobs are sorted and then the job with the highest priority is selected that is to be processed first [17]. Scheduling rules are mainly classified as:

i) Local rules deal with the dispatching information which are mostly available locally in specific areas likes banking, PTCL offices in which the dispatching rule first come first serve is applied.

ii) Global rules consist of scheduling knowledge that are mostly available on shop floor in manufacturing sectors like sequencing of jobs on machines according to its demand and priority.

iii) Static rules are used when the arrival of jobs is constant over time means that a specified number of jobs is to be processed every time during a period. Moreover, do not change over time and do not see what happens at the shop floor.

iv) Dynamic rules include the scheduling techniques for the job that change over a period, it is time dependent and changes with the status of the job that is arriving to the system.

v) Forecast rules gives priority to the jobs based upon it future prediction which the job to be come across and according to the status of machines.

There are more than a 100 of different scheduling rules that has been developed over the period. Most commonly used dispatching rules as follows.

2.5 Shortest Processing Time

Shortest Expected Processing Time (SEPT) or Shortest Processing Time (SPT) is the dispatching rule which prioritize the job based on shortest processing time. The job with the shortest processing time is to be processed first according to this rule. The SPT have several sub-kind or versions like,

i) SPRT: Total shortest remaining processing time. The job with the short-time remaining is processed first in the queue. This rule is best applicable to semi-finished jobs. It is related to work remaining on a job.

ii) TSPT: Truncated shortest processing time that means the job with shortest operation time is processed first. This rule is related with processing time and the jobs with less processing time remaining are processed first.

iii) WST: Weighted shortest processing time. Each shortest process time is given weight according to some criteria and the job with the smallest ratio is processed first. This rule is related to high priority jobs in which the jobs having more weight in terms of importance are processed first under this rule.

iv) LWR: Least work remaining in terms of its operations. Under this rule the jobs with less work remaining as well as less processing time remaining are processed first.

2.6 Longest Processing Time

Longest Expected Processing Time (LEPT) or Longest Processing Time (LPT) is the dispatching rule which prioritize the job base on the longest processing time. The job with

the longest processing time is to be processed first according to this rule. The other sub-kind versions of LPT are,

i) **TLPT: Total longest processing time. The jobs with the highest total longest processing time are processed first on machines. This rule is the opposite of shortest processing time. In terms of Makespan the shortest processing time is better than longest processing time, and in cases where jobs having high priority and have large processing time then longest processing time rule is better.**

ii) **LRPT: Total longest remaining processing time. This rule is best applicable to semi-finished products with largest processing time remaining jobs having high priority.**

iii) **MWR: Most work remaining according to the operations required. The semi-finished jobs on which more work is remaining in terms of processing times this rule can be used. Very important when the most work remaining jobs are of high priority.**

2.7 Earliest Due Date

The Earliest Due Date (EDD) arranges the jobs based on the smallest due date or which job is needed earlier to be processed first. The due date for a job is that date at which the job is to be completed and to be received by the supplier. It is very important among all the dispatching rules. The other versions are,

i) **ODD: Operation Due Date. That operation which has the short due date is processed earlier. Under this policy the jobs that have short due dates are loaded first on machines to complete all jobs on or before due dates.**

ii) **MDD: Modified Due Date. Sometimes jobs are assigned with a new due date from the set of jobs waiting in the front of machine and then that are prioritize based on earliest due date. The job with a positive slack is considered to be on the original date while the job with negative slackness is considered to be equal to available time and processing time.**

iii) **MODD: Modified Operation Due Date. Sometimes jobs are assigned with a new due date from the set of jobs waiting in the front of machine and then that are prioritize based on operation due date.**

2.8 Job Slack Time

The Job Slack Time (JST) prioritize job based on minimum slackness. The job slack time is computed as the difference between the job due date, the work remaining, and the current time. Simply, Job slack time = (job due date –work remaining-current time) Operation slack time. The job with the smallest operation slackness is to be prioritizing first.

Operation slack time = job slack time / number of remaining job operation.

2.9 Critical Ratio

The job with the smallest critical ratio is processed first. The critical ratio can be calculated as the ratio between job allowances divided by remaining work time.

Critical ratio = job allowances / remaining work time

2.10 First Come First Served

In First Come First Served (FCFS), the job that arrives first on the machines will be processed first. This is the simplest of all the dispatching rule with the job that comes first to the system will be processed first on the machine and then followed by all the other jobs. The other version is FASFS or SRT First at the shop first served or smallest release time means that a job comes first to the shop will be given preference to be loaded on the machine.

2.11 Last Come First Served

The Last Come First Served (LCFS) select the job which come in the end will be processed first. In this priority rule, the jobs, which join the system at the last, will be loaded on the machines first and followed by several other through that sequence of operation. It is categorized in the local dispatching rule. The scheduling process is usually takes less time than other dispatching rule due to its simplicity.

2.12 Least Flexible Job

The Least Flexible Job (LFJ) prioritizes the job base on the least flexibility. Flexibility means how much it is able to the change to process.

2.13 First off First On

In First Off First on (FOFO), the job with the operation that could be completed earliest will be processed first even if the job is not in queue. The machine will be free until job arrives.

2.14 Random

In random case, the jobs are randomly selected and then processed first. There is no specified rule to process the job and any of the job is to be loaded on the machine first. The other version of random is, BIASED-RANDOM: Service in biased random order. In this rule jobs are not to be comparatively selected as that in the random case but from a random area. The selection process is biased according to some other dispatching rule like, SPT and EDD.

2.15 Categories of Scheduling Problems

The sequencing and scheduling problem are categorized as.

2.15.1 Deterministic

Deterministic are the type of problem which do not include stochastic factors which means that all the elements of the scheduling problems such as arrival time, processing time, due date, ordering do not make use of any probabilistic approaches. And all the elements of problems have the constant values which do not change over time.

2.15.2 Static

In the static case, the set of job over time does not change and it is available beforehand. It is like the deterministic except that nature of job arrival is different. Over a period, a specified number of jobs are to be entering to the system and to be processed which is predefined from the start.

2.15.3 Dynamic

In dynamic problem, the set of jobs changes repeatedly, arrivals are at different time. The dynamic environment is a real scheduling environment in which there are no assumptions.

2.15.4 Stochastic

The stochastic models involved the probability factor. In a stochastic model, the jobs are processed on the machines based on probability e.g.: -there are 20% of chances that 100 jobs are to process on the machines in an hour. The stochastic reflected the real-world environment that why that is mostly used. The subject, scheduling mostly includes stochastic problem.

2.16 Performance of Dispatching Rules under Perfect Sequencing Flexibility

According to Baker, the number of different process plans for completing a job can measure the sequencing flexibility [15]. Though this is an initial step towards measuring sequencing flexibility, it could be improved upon by relating to the total number of process plans to the total number of operations needed to be performed for completing the job. The need exists to precisely measure sequencing flexibility before its impact on scheduling methods can be evaluated.

For a practical situation when perfect sequencing flexibility occurs and the job visits a subset of machines. It has been observed that performance of dispatching rules such as the Shortest Processing Time rule (SPT), First Come First Served (FCFS), First in System (FIS), Least Work Remaining (LWR) and Fewest Remaining Operations (FRS) on important criteria such as the mean flow time and work-in-process inventory. SPT assigns highest priority for machine use to the task with the smallest processing time on the machine. FCFS assigns highest priority to the task that listed itself earliest at the machine. FIS assigns highest priority to the job that arrived earliest in the shop. LWR assigns highest priority to the job that has the least amount of work yet to be completed on it in the shop. FRS assigns highest priority to the job that has the fewest remaining tasks to be completed. To analyze these dispatching rules a simulation model is carried out to study the relative performance of scheduling rule. In the simulation, study of these rules a

Method of Batch Means (Schmeiser 1982) was selected as the statistical methodology used in the experimentation [18]. For each dispatching rule, an expected overall machine utilization, and Sequencing Flexibility Measures (SFM) of 0.0 and 1.0 are calculated. It is noted that all rules improve their performance when the SFM increases from 0 to 1. The improvement is very large for rules that perform poorly when the SFM value is 0. Thus, the differences between the rules decrease when sequencing flexibility is introduced. The benefits of sequencing flexibility in terms of reduced flow times and inventories are sufficiently great; this has implications for designing products in such a way as to maximize the potential sequencing flexibility in manufacturing the products. Therefore, product design has largely emphasized process and product compatibility with less emphasis, if any, on the sequencing flexibility.

2.17 Solution Approaches to Manufacturing Scheduling Problem

In the mid nineteen century, the researcher started to study the scheduling concept and theory. The literature review present different solution approaches to scheduling problems. These approaches have been divided into distinct categories. Jain and Meeran has divided the solution techniques of job shop scheduling problem into optimization and approximation techniques [3, 19]. The sub classes of approximation technique are based on general algorithm and tailor algorithm. The tailor algorithms are divided into heuristic and priority rules.

Genetic algorithms are classified into artificial intelligence techniques and local scheduling techniques. Furthermore, optimization techniques are dividing into enumerative techniques and efficient methods. The enumerative techniques are categorized into branch and bound algorithm and mathematical optimization techniques.

Another class of solution approaches to scheduling is the benchmark method in which the result of problems is compared with some standard ones to find the strength of the results. One common of them is the shifting bottleneck techniques. Bottleneck refers to the case in the transfer or assembly line in which a station which takes the maximum processing time among the all stations. The objective is to reduce the bottleneck station time by the method of shifting bottleneck, which optimizes the station time.

2.18 Optimization and Approximation Techniques

Optimizations techniques are very powerful technique that are used to manage the resources efficiently that results in maximize the profit. This technique is used for finding the optimal solution it is most common used techniques because of its optimality.

A method that guarantees that the result is an optimal solution is called exact solution otherwise it is known as approximation method, which are near to the optimal solution but not the exact solution. In manufacturing scheduling system, heuristics and dispatching rules are commonly used to find appropriate solution to scheduling problems.

2.19 Priority Dispatching Rules

Priority dispatching rules are called with several names in the literature such as heuristics rules, dispatching rules, scheduling rules, priority rules. These rules are used to process all the jobs on the machines that is waiting to be processed. Under the last six decades, there is a significant increase in the field of scheduling research. Due to the global competitiveness, the priority dispatching rules has a scope in the complexity and difficulty in the manufacturing. The scheduling research is categorized into two main classifications:

2.19.1 Theoretical Research

The theoretical research problems are based on assumptions like, no machine breakdown, all jobs are available at time zero etc. The theoretical research problems are restricted to literature only and cannot be applied to real scheduling environments.

2.19.2 Experimental Research

The experimental research problems are based on real industrial data, which includes machine breakdowns, transfer time, loading/unloading time etc. The experimental research problems can be applied to real scheduling environments.

2.20 Dispatching Rules and Their Classification

According to Panwalker, the scheduling rules are divided into several classes [20]. He made a survey of different dispatching rules and concludes that Priority dispatching rules are the techniques, which assign a number to every job according to some criteria, the

job that is to be processed is either to be static or dynamic or it to be local or global. Static rule is the rule in which the value of job priority does not change with the passage of time such as Earliest Due Date (EDD) while in the dynamic rule the job priority change with time like, Least Slack Rule (LSR). The local scheduling rule is limited up to the information about the job waiting at a machine while the information that is necessary for a global rule are the state of the machine and jobs at other machines.

Panwalker has divided the scheduling rules into the following classification globally [20].

Simple Priority Rules refers to that information which is needed for a specific job about its due date, processing time, and remaining number of processing. It also includes the information about the length of queue for a machine as well as random selection rules, which do not need any information about specific job e.g. shortest remaining processing time, Longest remaining processing time, earliest due date etc.

Combination of Simple Priority Rule includes the combination of two or more simple priority rules. In many instances, two rules apply to the same queue under different circumstances e.g. First come first serve, critical ratio rules etc.

Weighted Priority Rules includes the simple priority rules with the weighted values. It may be either the simple priority rule or combination of priority rules.

Heuristic Scheduling Rules is a more detailed rule, which includes several elements, scheduling alternative operation machine loading, and alternative routing effects etc.

Other Rules consists of rules designed for specific shop, combination of priority indexes base on mathematical function of job parameter, or those rules, which are not mentioned earlier, e.g. Work in Next Queue (WINQ).

2.21 Deterministic Job Shop Scheduling: Past, Present and Future

The work of Jain and Meeran in 1999 in the field of scheduling helps in the further improvement of job shop scheduling problems over the last forty years [3]. Their work does not help us in understanding the job shop problems but also in the benchmark problems. The objective of scheduling is to solve constraints based problems. In the production environment, scheduling concerns with the management of resources that are limited in number.

For a job shop, there are 'S' sets of 'N' number of jobs that is to be processed on 'M' number of machine along with its own complicated operations 'O'. These operations are in some predetermine constraints. The objective is to reduce the total completion time.

Roy and Sussman (1964) presented the disjunctive graph method [5]. Friedman (1955) developed Boolean algebraic method for representation of different scheduling processes. Thomson and Gifler give the idea of the template of dispatching rule. Gantt (1919) developed Gantt chart for scheduling purpose to make the scheduling processes easier.

In the preliminary stages, Johnson (1954) developed algorithm for two machines flow shop problems and it is to be consider one of the earliest work in the field of scheduling and sequencing. After that, several algorithms are developed in late 1960s after Johnson work. One of the important in them is the Branch and Bound algorithm which is made in tree like structure used for the solution space for all those scheduling which is supposed to be feasible.

Conway had presented a scheme of classification in the scheduling problems which is used for basic problems, denoted by four letters as (A/B/C/D) [16]. In this classification, A represents the number of jobs to be processed on machines. B represents the number of machines on which jobs to be processed. The letter C shows the sequential flow of the jobs. While D represents shows the criteria on which the scheduling is to be selected. For non-basic problems, used another notation which is denoted as (α/β/ϒ) [21]. In this notation, α is used to represent number of machines, β represents the precedence constraints while ϒ shows the criteria for jobs.

Nowadays there is a considerable number of different heuristics and algorithms being developed and more than two hundred heuristics is presented in literature which make scheduling process to be more optimize and efficient.

This chapter gives an overview of different scheduling theory and concepts and it also includes optimization and approximation techniques, priority rules and their classification. The next chapter will cover the complete methodology for the development of new proposed heuristics. Scheduling is a very broad concept. There are different environments of scheduling. Dealing with scheduling in one environment is completely different from another environment. The deterministic environment is based on some

assumptions while the dynamic environment is a real scheduling environment. For scheduling problems, there are many different techniques ranging from simple dispatching rules to complex heuristics and genetic algorithms.

CHAPTER 3: DEVELOPMENT OF PROPOSED HEURISTICS

In the theory of scheduling, the terms such as heuristic rules, scheduling rules, priority rules, or dispatching rules are used by different authors but all these terms refer to the same concept Panwalker [20]. The research in the field of scheduling and sequencing has been increased from the last six decades. Researchers are continuously working hard for finding out heuristics and algorithms for solving FSSP and JSSP. These heuristics and algorithms are used to sequence jobs on machines in complex manufacturing environment to achieve certain criteria.

According to Panwalker, scheduling research can be classified into two main classes, Theoretical research and experimental research [20]. The theoretical research deals with the optimization of static problems. The experimental research deals with the optimization of static and dynamic scheduling problems. The static scheduling problems are based on assumptions in which some important real-life scenario is missing such as machine breakdown, maintenance time etc. The experimental scheduling problems deals with real life scenario and considers all the dynamic data as well. The experimental research problems are also called practical scheduling Pinedo [8, 22].

In this chapter, an approach to the development of proposed heuristic is explained in detail. An example problem is solved using the proposed heuristic. The proposed heuristic is coded in Matlab. The strength of proposed heuristic is tested against different FSSP and JSSP benchmark problems. A comparative analysis of proposed heuristic with other heuristics and dispatching rules is carried out to gauge its strength.

3.1 Development of Proposed Heuristic

Some common characteristics exist for optimization problems to evaluate their quality. According to S. Maqsood, these characteristics should be formulated in efficient manner so that they can be applicable to any kind of new or existing scheduling problems to achieve efficiency [19]. The following are the most important quality characteristics of optimization models.

i) Validity and reliability

ii) Ease of Interpretation and comparison iii) Cost effectiveness

iv) Ease of testing

The development of proposed heuristic is the result of studying heuristics rules and algorithms in the literature review. The development process is divided into three main phases.

i) Need identification phase

ii) Analysis and synthesis phase

iii) Proposed heuristic phase.

The need identification phase deals with literature review of scheduling problems in which different heuristic rules and their solution techniques are studied in detail. The second phase is the analysis and synthesis phase in which the existing heuristics are modified. The results of existing and modified heuristics are compared; decision is made based on their results. The third phase is to develop a new heuristic by modifying the existing one or using a completely new idea.

3.2 Proposed Heuristic

The proposed heuristic is a processing time based heuristic and the performance criterion is Makespan. Makespan is the total completion time of all jobs. The main objective in Makespan criterion is to minimize the total completion time of jobs by sequencing jobs on machines. The proposed utilizes three sequences to find the Makespan. The sequence is applied to operation 1 and then the rest of the operations are completed through early finished job based. Three different Makespan values are obtained based on three different sequences and the minimum of the three Makespan values is selected as a final Makespan.

3.3 Logic of the Proposed Heuristic

The proposed heuristic uses three sequences. The decision module decides whether operations 1 is completed or not, if operation finishes then jobs are loaded on machines based on early finish time, there is another decide module which decides if operations are completed. When all operations are completed then three different Makespan values are displayed based on three different sequences. Final the minimum of the three Makespan

value is selected. The complete logic of the proposed heuristic is shown in the form of a flow chart in figure 3.1.

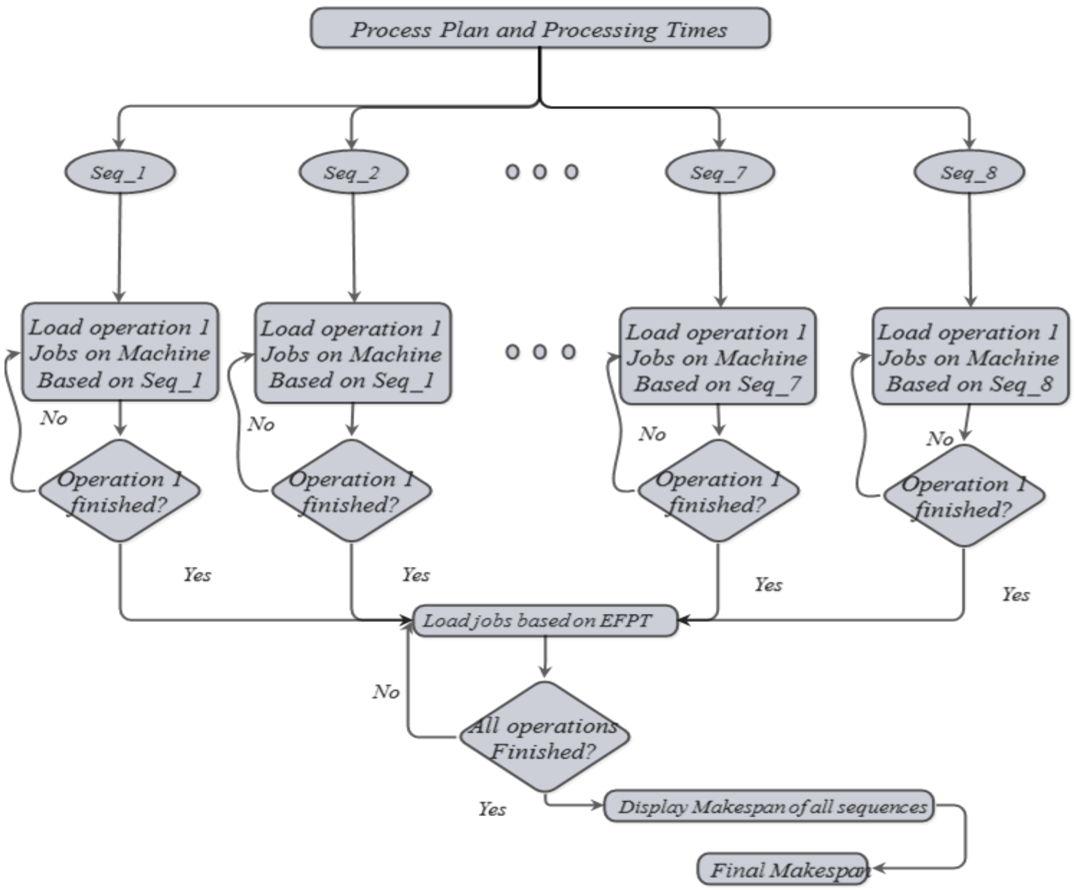

Figure 3.1: Proposed heuristic for Scheduling Problem

3.4 Stepwise Procedure of Proposed Heuristic

The procedure of proposed heuristic is simple and the stepwise procedure is explained with the help of example problem as shown in table 3.1.

Table 3-1: Example problem 4-jobs, 3-machines

JOBS	Operation 1		Operation 2		Operation 3	
	PP	PT	PP	PT	PP	PT
J1	1	5	2	8	3	2
J2	3	7	1	3	2	9
J3	1	1	3	7	2	10

J4	**2**	**4**	**3**	**11**	**1**	**7**

The example problem is a P (4X3) problem which means 4 jobs and 3 machines. Process plan for jobs is represented by PP, and processing times by PT. when a job completes its processing times on a machine, operation 1 is completed for that job i.e. Operation 1 for job 1 is on machine 1 with a processing time of 5 unit, for job 2 is on machine 3 with a processing time of 7 unit, for job 3 is on machine 1 with a processing time of 1 unit, for job 4 is on machine 2 with a processing time of 4 unit and so on. The table 3.1 below shows the example problem.

3.4.1 STEP # 1: Makespan through sequence 1

Find the total processing time of each job.

This is calculated as

For job 1, [5+8+2] = 15,

For job 2, [7+3+9] = 19,

For job 3, [1+7+10] = 18,

For job 4, [4+11+7] = 22.

All these values are shown in the last column under the heading sum of PT in table 3.2.

Table 3-2: Last column shows sum of all processing times

JOBS	Operation 1		Operation 2		Operation 3		Sum of PT
	PP	PT	PP	PT	PP	PT	
J1	1	5	2	8	3	2	15
J2	3	7	1	3	2	9	19
J3	1	1	3	7	2	10	18
J4	2	4	3	11	1	7	22

The last column which represent the total processing times is ranked from highest to lowest as shown in table 3.3. Job 4 has the highest total processing time so this job will go first in the sequence followed by job 2, job 3, and job 1. The operation 1 is completed through sequence 1 while operations 2 and 3 are completed through early finished

processing time based rule i.e. the job that has completed its processing time early will seize the next required machine.

Table 3-3: Shows sequence # 1

Ranked from highest to lowest	**22**	**19**	**18**	**15**
Sequence 1	**J4**	**J2**	**J3**	**J1**

This procedure is shown graphically through a Gantt chart developed in excel in figure 3.2. The x axis represents time and y axis represent machines, and jobs are sequenced on machines and are differentiated through colors. The colors of job1, job2, job3, and job4 are Yellow, Green, Red, and Black respectively.

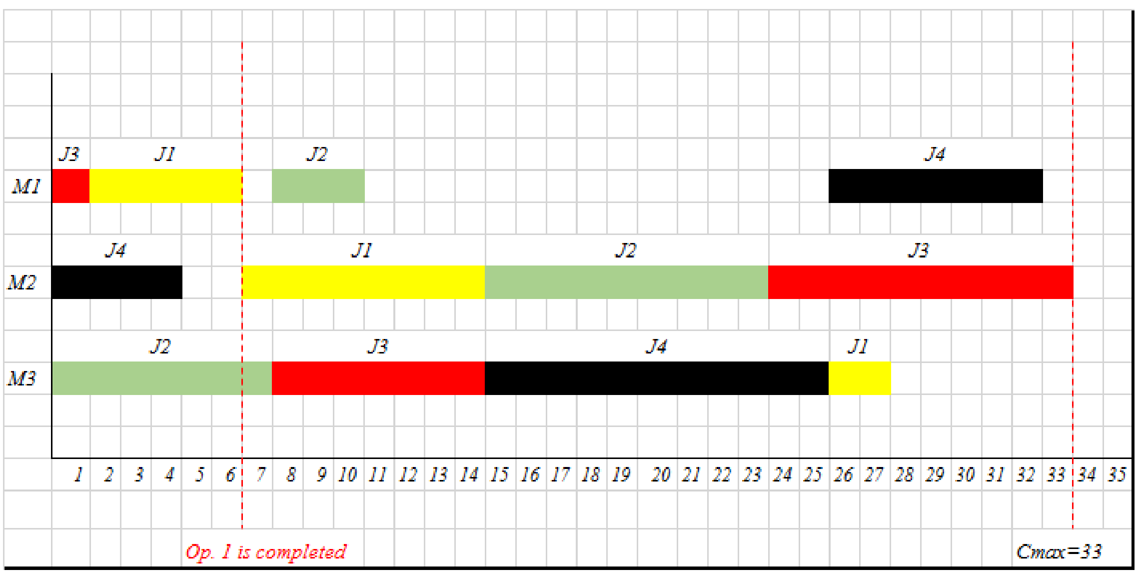

Figure 3.2: Shows the result of sequence # 1

Job 4 is loaded first on machine 2 followed by job 2 on machine 3, job 3 on machine 1 and finally job1 on machine 1. Operation 1 is completed. Job3 has completed its processing times early, as compared to other jobs that are still under processing after time 1, so job 3 will seize its next required machine 3, followed by job 4 on machine 3, then job 1 on machine 1 and so on. In this way, all operations are completed and Makespan value is obtained which is 33 as shown in figure 3.2. Result of sequence 1 is a Makespan value 33.

3.4.2 STEP # 2: Makespan through sequence 2

The sequence 2 is obtained by arranging all jobs in ascending order i.e. ranking from lowest to highest as opposed to sequence1 based on the sum of their process times. So, sequence 2 is formed as shown in table 3.4. The first job to seize a machine is job 1 followed by job 3, job2 and finally job 4 on machines 1, 1, 3and 2 respectively.

Table 3-4: Shows sequence # 2

Ranked from lowest to highest	15	18	19	22
Sequence 2	J1	J3	J2	J4

This procedure is shown graphically through a Gantt chart developed in excel in figure 3.3. The x axis represents time and y axis represent machines, and jobs are sequenced on machines and are differentiated through colors. The colors of job1, job2, job3, and job4 are Yellow, Green, Red, and Black respectively.

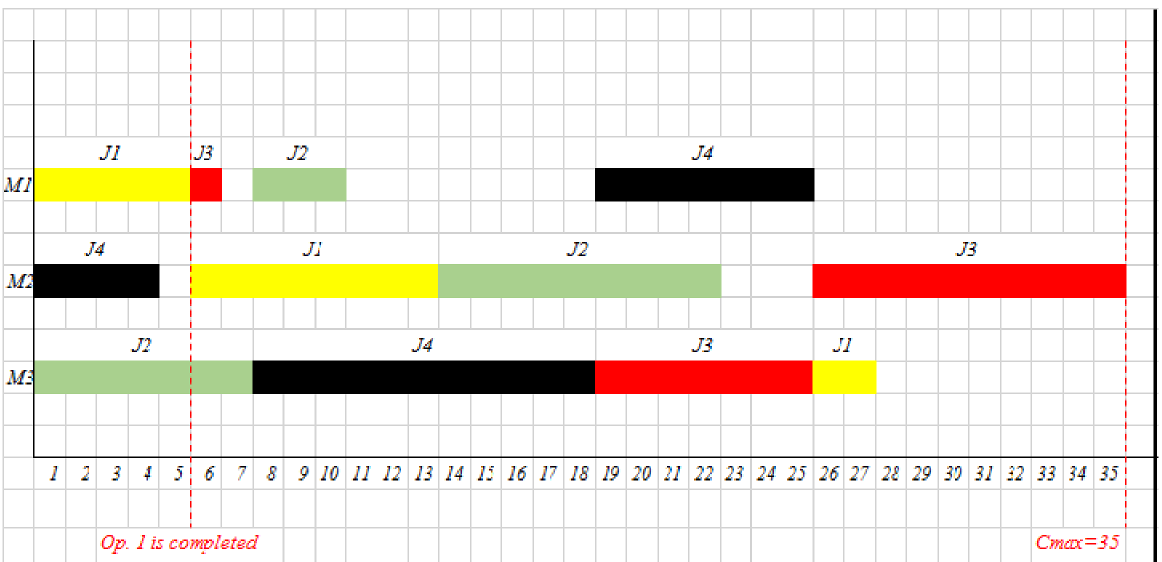

Figure 3.3: Shows the result of sequence # 2

Job 1 is loaded first on machine 1 followed by job 3 on machine 1, job 2 on machine 3 and finally job 4 on machine 2. Operation 1 is completed. Job4 has completed its processing times early, as compared to other jobs that are still under processing after time 4, so job 4 will seize its next required machine 3, followed by job 1 on machine 2, then job 3 on machine 3 and so on. In this way, all operations are completed and Makespan value is obtained which is 35 as shown in figure 3.3. Result of sequence 2 is a Makespan value 35.

3.4.3 STEP # 3: Makespan through sequence 3

The sequence 3 is obtained by arranging half or 50% jobs first according to LPT rule and then arranging the remaining jobs according to SPT rule based on the sum of their process times. So, sequence 3 is formed as shown in table 3.5. The first job to seize a machine is job 4 followed by job 2, job 1 and finally job 3 on machines 2, 3, 1, and 1 respectively.

Job 4 is loaded on machine 2 followed in the sequence by job 2 on 3, job 1 on machine 1, and finally job 3 on machine 1. In this way operation 1 is completed. Then jobs are loaded on machines based on early finished processing time. So, job 4 has completed its processing time earlier as compared to other job so job 4 will be loaded on next required machine which is machine 3.

Table 3-5: Shows sequence # 3

LPT	**22**	**19**		
SPT			**15**	**18**
Sequence 3	**J4**	**J2**	**J1**	**J3**

Then job 1 will be loaded on machine 2, job 3 on machine 3, and job 2 on machine 1 and so on. In this way, all are operation are competed and Makespan value is obtained which is shown in the figure 3.4. Result of Sequence is a Makespan value 35.

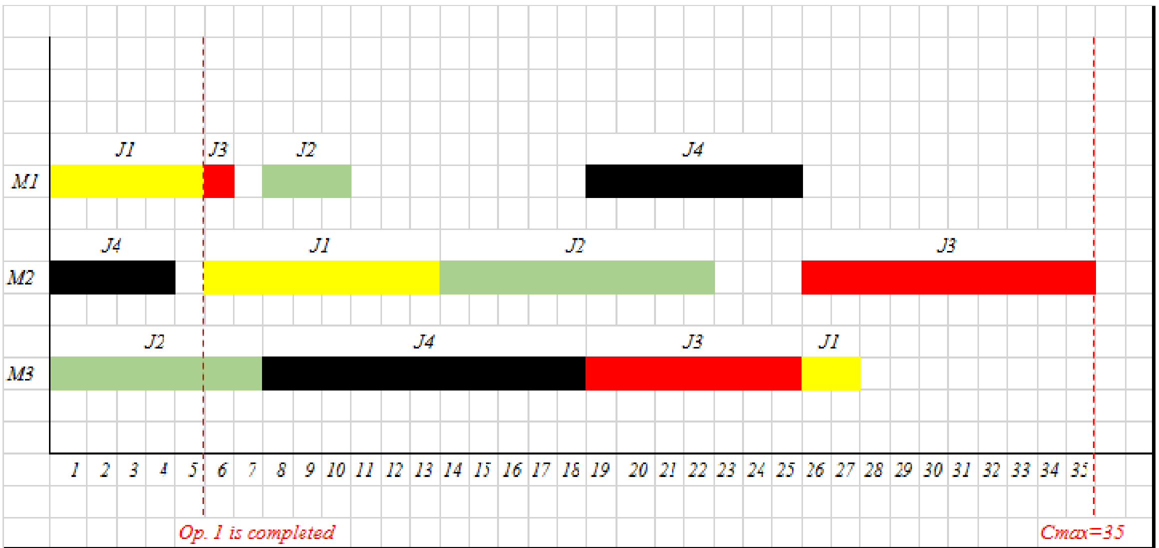

Figure 3.4: Shows the result of sequence # 3

3.4.4 STEP # 4: Makespan through sequence 4

The sequence 4 is obtained by arranging jobs first according to ascending order. Then first half or 50% of the jobs will be sequenced based on SPT rule and remaining jobs on LPT rule. So, sequence 3 is formed as shown in table 3.6. The first job to seize a machine is j1 followed by job 3, job 4 and finally job 2 on machines 1, 1, 2 and 3 respectively.

Table 3-6: Shows sequence # 4

SPT	15	18		
LPT			22	19
Sequence 4	J1	J3	J4	J2

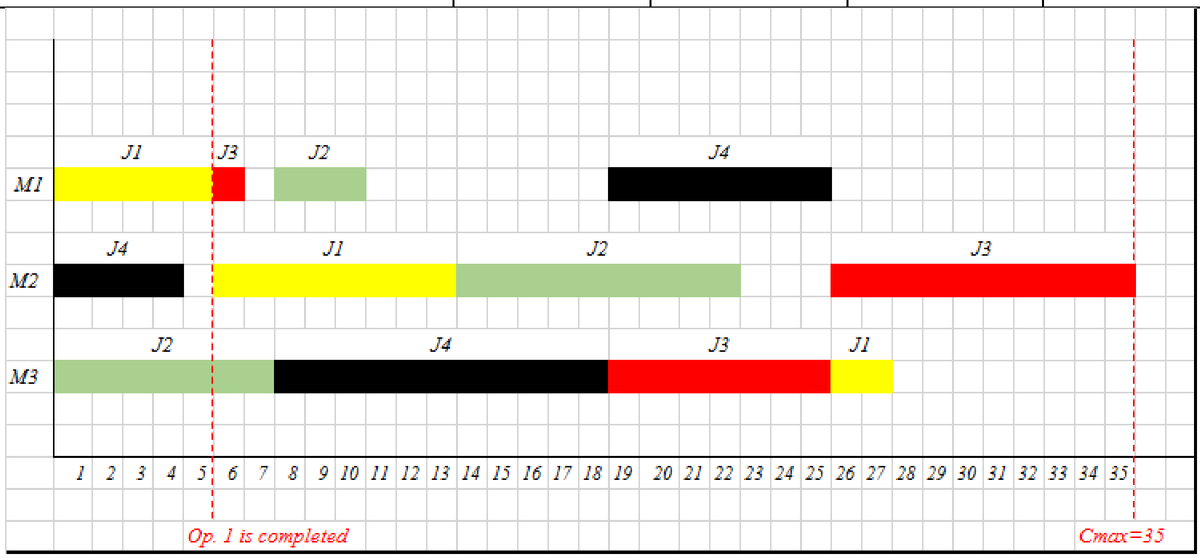

Figure 3.5: Shows the result of sequence # 4

Job 1 is loaded on machine 1 followed in the sequence by job 3 on 1, job 4 on machine 2, and finally job 2 on machine 3. In this way operation 1 is completed. Then jobs are loaded on machines based on early finished processing time. So, job 4 has completed its processing time earlier as compared to other job so job 4 will be loaded on next required machine which is machine 3. Then job 1 will be loaded on machine 2, job 3 on machine 3, and job 2 on machine 1 and so on. In this way, all are operation are competed and Makespan value is obtained which is shown in the figure 3.5. Result of Sequence is a Makespan value 35.

3.4.5 STEP # 5: Makespan through sequence 5

The sequence 4 is obtained by arranging 25% of jobs first according to LPT rule and then arranging the next 25% of jobs according to SPT rule and so on. So, sequence is formed as shown in table 3.7. The first job to seize a machine is j4 followed by job 1, job 2 and finally job 3 on machines 2, 1, 3, and 1 respectively.

Table 3-7: Shows sequence # 5

LPT	**22**		**19**	
SPT		**15**		**18**
Sequence 5	**J4**	**J1**	**J2**	**J3**

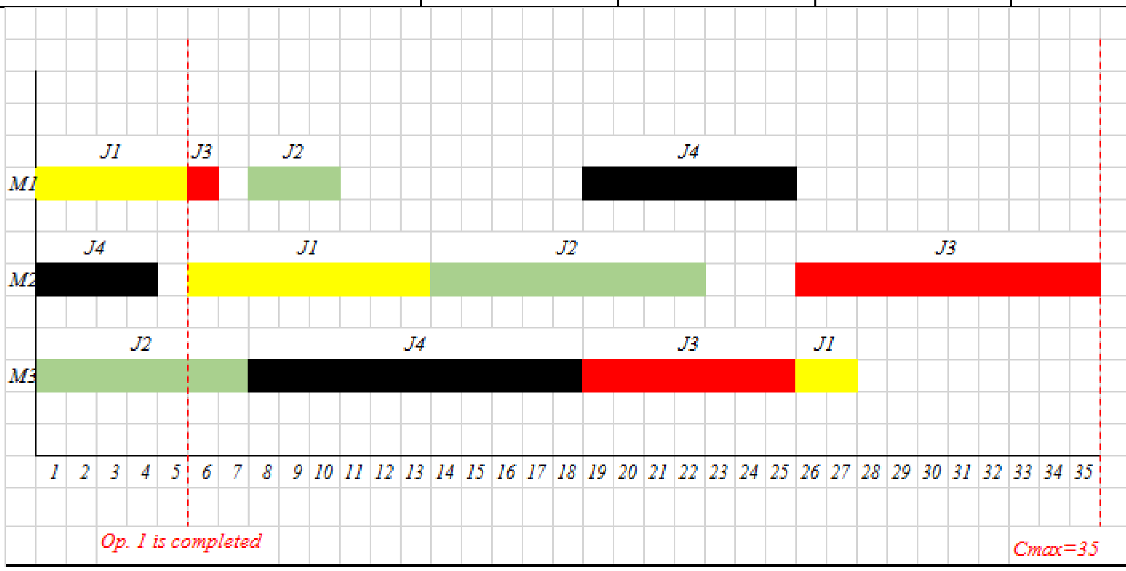

Figure 3.6: Shows the result of sequence # 5

Job 4 is loaded on machine 2 followed in the sequence by job 1 on 1, job 2 on machine 3, and finally job 3 on machine 1. In this way operation 1 is completed. Then jobs are loaded on machines based on early finished processing time. So, job 4 has completed its processing time earlier as compared to other jobs so job 4 will be loaded on next required machine which is machine 3. Then job 1 will be loaded on machine 2, job 3 on machine 3, and job 2 on machine 1 and so on. In this way, all are operation are competed and Makespan value is obtained which is shown in the figure 3.6. Result of Sequence is a Makespan value 35.

3.4.6 STEP # 6: Makespan through sequence 6

The sequence 6 is obtained by arranging 25% of jobs first according to SPT rule and then arranging the next 25% of jobs according to LPT rule and so on. So, sequence is

formed as shown in table 3.8. The first job to seize a machine is job 1 followed by job 4, job 3 and finally job 2 on machines 2, 1, 3, and 1 respectively.

Table 3-8: Shows sequence # 6

SPT	15		18	
LPT		22		19
Sequence 6	J1	J4	J3	J2

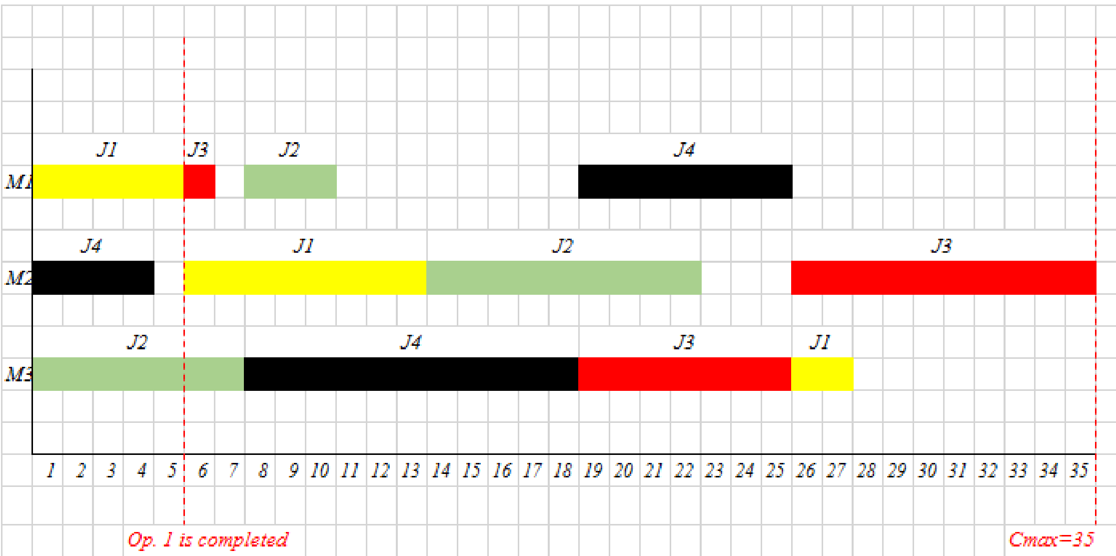

Figure 3.7: Shows the result of sequence # 6

Job 1 is loaded on machine 1 followed in the sequence by job 4 on 2, job 3 on machine 1, and finally job 2 on machine 3. In this way operation 1 is completed. Then jobs are loaded on machines based on early finished processing time. So, job 4 has completed its processing time earlier as compared to other jobs so job 4 will be loaded on next required machine which is machine 3. Then job 1 will be loaded on machine 2, job 3 on machine 3, and job 2 on machine 1 and so on. In this way, all are operation are completed and Makespan value is obtained which is shown in the figure 3.7. Result of Sequence is a Makespan value 35.

3.4.7 STEP # 7: Makespan through sequence 7

The sequence 7 is obtained by arranging jobs first according to SPT rule and then arranging the jobs according to LPT rule every time, the format for arrangement would be like SPT, LPT, SPT… and so on. So, sequence is formed as shown in table 3.9. The first job to seize a machine is job 1 followed by job 4, job 3 and finally job 2 on machines 1, 2, 1, and 3 respectively.

Table 3-9: Shows sequence # 7

SPT	**15**		**18**	
LPT		**22**		**19**
Sequence 7	**J1**	**J4**	**J3**	**J2**

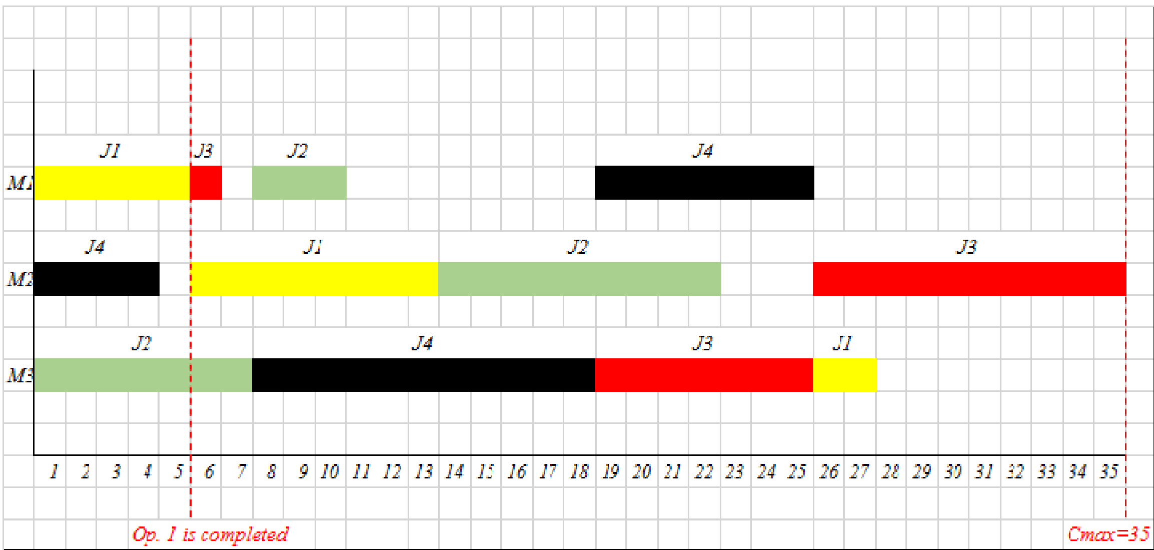

Figure 3.8: Shows the result of sequence # 7

Job 1 is loaded on machine 1 followed in the sequence by job 4 on 2, job 3 on machine 1, and finally job 2 on machine 3. In this way operation 1 is completed. Then jobs are loaded on machines based on early finished processing time. So, job 4 has completed its processing time earlier as compared to other jobs so job 4 will be loaded on next required machine which is machine 3. Then job 1 will be loaded on machine 2, job 3 on machine 3, and job 2 on machine 1 and so on. In this way, all are operation are completed and Makespan value is obtained which is shown in the figure 3.8. Result of Sequence is a Makespan value 35.

3.4.8 STEP # 8: Makespan through sequence 8

The sequence 8 is obtained by arranging jobs first according to LPT rule and then arranging the jobs according to LPT rule, the format for arrangement would be like LPT, SPT, LPT… and so on. So, sequence is formed as shown in table 3.10. The first job to seize

a machine is job 4 followed by job 1, job 2 and finally job 3 on machines 2, 1, 3, and 1 respectively.

Table 3-10: Shows sequence # 8

LPT	**22**		**19**	
SPT		**15**		**18**
Sequence 4	**J4**	**J1**	**J2**	**J3**

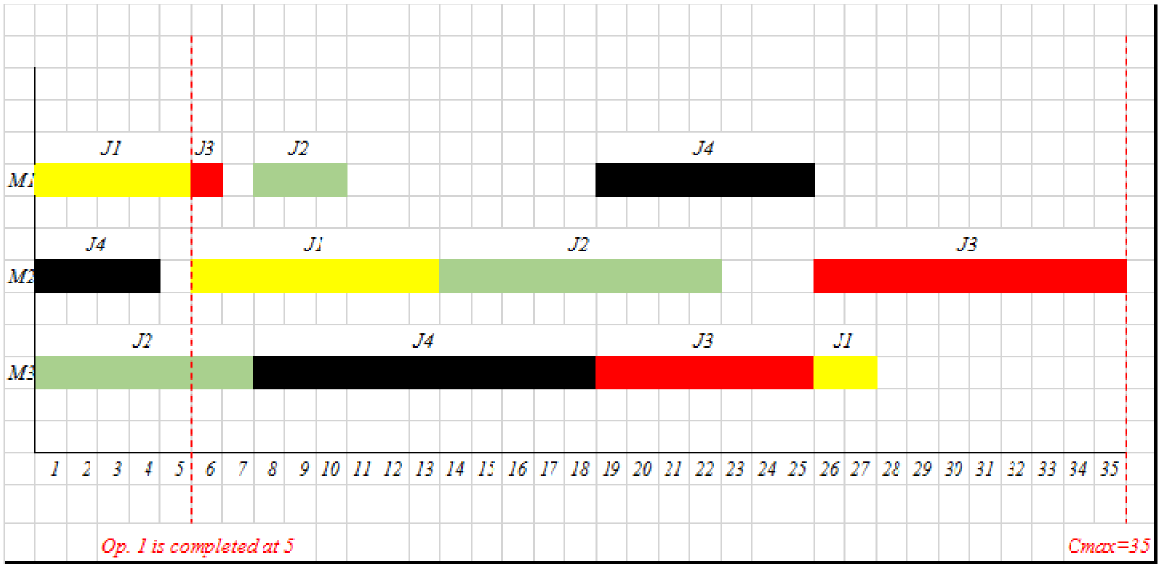

Figure 3.9: Shows the result of sequence # 8

Job 4 is loaded on machine 2 followed in the sequence by job 1 on 1, job 2 on machine 3, and finally job 3 on machine 1. In this way operation 1 is completed. Then jobs are loaded on machines based on early finished processing time. So, job 4 has completed its processing time earlier as compared to other jobs so job 4 will be loaded on next required machine which is machine 3. Then job 1 will be loaded on machine 2, job 3 on machine 3, and job 2 on machine 1 and so on. In this way, all are operation are competed and Makespan value is obtained which is shown in the figure 3.9. Result of Sequence is a Makespan value 35.

The results of sequence 1, sequence 2… and sequence 8 are 33, 35… and 35 respectively. The minimum of the eight is 33, which is the final Makespan value. The Makespan of the example problem above according to proposed heuristic is 33.

This chapter explains the phases that lead to the development of proposed heuristic. The proposed heuristic has the same quality characteristics as that of other heuristics. It

uses eight sequences to find the Makespan. Stepwise procedure is explained and an example problem is solved through it to check its validity and effectiveness. The next chapter deals with the results of proposed heuristic in comparison with other heuristics.

CHAPTER 4: RESULTS AND DISCUSSIONS

In this chapter, benchmark problems are solved by using the proposed heuristic. The benchmark problems are of different variety, ranging from smaller 6 jobs 6 machines to larger 20 jobs 20 machines. The proposed heuristic is tested against different dimensions of problems to check its effectiveness. The results are compared with well-known traditional heuristics like First come first serve, longest processing time, shortest processing time earliest due date etc. The results are tested on benchmark problems of flow shop and job shop. The list of traditional heuristic with which comparison of proposed heuristic will take place is shown below in table 4.1.

Table 4-1: Shows different dispatching rules

S. No	RULES	DESCRIPTION
1	FCFS	First Come First Serve
2	SPT	Shortest Processing Time
3	LPT	Longest Processing Time
4	MS	Minimum Slack
5	WSPT	Weighted Shortest Processing Time
6	CR	Critical Ratio
7	EDD	Earliest Due Date

The proposed heuristic is coded in MATLAB. The process plan and processing times are stored in excel sheet. From excel these data are imported into MATLAB as an input to the proposed heuristic.

4.1 Comparative Analysis of Job Shop Problems

In this section, comparative analysis of proposed heuristic is done with the traditional heuristics and is discussed below.

4.1.1 Fisher and Thompson Job Shop Benchmark Problems

The fisher and Thompson represented by FT are benchmark problems developed in 1963 [23]. The problems selected from this test-bed are FT06 (6 jobs, 6 machines) and FT10

(10 jobs, 10 machines). The results are compared with traditional heuristics. The decision is made on overall mean % GAP. The % GAP is defined and calculated as the deviation of the Makespan value obtained by a particular heuristic from the global Makespan is the % GAP, and is calculated as

GAP% = 100*(Makespan found – Global Makespan) / Global Makespan

In this case the Makespan found refers to the result of proposed heuristic while Global Makespan refers to the best known lower bound.

4.1.1.1 For FT-06

The best result is given by a heuristic rule EDD with a Makespan value of 63, and a % GAP of 14.5%. The worst result is given by CR with a Makespan value of 81, and a % GAP of 47.3%. The average of all 7 heuristics is 70 with a % GAP of 27.3%. The proposed heuristic performs better against the same set of benchmark problem. The Makespan obtained by proposed heuristic is 59, with a % GAP of 7.3%.

4.1.1.2 For FT-10

The best results are given by heuristics LPT and MS with a Makespan value of 1168, and a GAP of 25.6%. The worst results are given by SPT and WSPT with a Makespan value of 1338,

Table 4-2: Results of Fisher and Thompson problems for P.H vs Dispatching Rules

Test Bed	*Fisher and Thompson (1963)*				Overall
Problem	FT-06 (OPT=55)		FT-10 (OPT=930)		Mean
Instances	6x6	GAP%	10X10	GAP%	Gap %
SPT	73	32.7	1338	43.9	38.3
LPT	67	21.8	1168	25.6	23.7
CR	81	47.3	1181	27.0	37.1
MS	67	21.8	1168	25.6	23.7
FCFS	65	18.2	1184	27.3	22.7
WSPT	73	32.7	1338	43.9	38.3
EDD	63	14.5	1246	34.0	24.3
Minimum	63	14.5	1168	25.6	20.1
Average	70	27.3	1232	32.5	29.9
Maximum	81	47.3	1338	43.9	45.6
P. Heuristic	59	7.3	1136	22.2	14.7

and a GAP of 43.9%. The average of all 7 heuristics is 1232 with a GAP of 32.5%. The proposed heuristic performs better against the same set of benchmark problem. The Makespan obtained by proposed heuristic is 1130, with a GAP of 22.2%. The results of fisher and Thompson benchmark problems are given in the table 4.2 above.

4.1.1.3 Over-All Mean Gap

The over-all mean gap for the best performance heuristics is 20.1%. The over-all mean gap for the worst performance heuristics is 45.6%. The over-all mean gap of proposed heuristic is 14.7% which is better even than the best performance. Thus, the proposed heuristic out performs all other traditional heuristics in terms of Makespan and GAP.

4.1.2 Lawrence Job Shop Benchmark Problems

The Lawrence benchmark problems are represented by LA, developed in 1984 [24]. The problems selected from this test-bed are LA-01 (10 jobs, 5 machines), LA-06 (15 jobs, 5 machines), LA-11 (20 jobs, 5 machines), LA-12 (20 jobs, 5 machines), and LA-26 (20 jobs, 10 machines). The results are compared with traditional heuristics. The decision is made on overall mean GAP.

4.1.2.1 For La-01

The best results are given by heuristics rules LPT and MS with a Makespan value of 752, and a GAP of 12.9%. The worst results are given by SPT and WSPT with a Makespan value of 1122, and a GAP of 68.5%. The average Makespan of all 7 heuristics is 909 with a GAP of 36.5%. The proposed heuristic performs better against the same set of benchmark problem. The Makespan obtained by proposed heuristic is 682, with a GAP of 2.4%.

4.1.2.2 For La-06

The best results are given by heuristics rules LPT, FCFS, and MS with a Makespan value of 926, and a GAP of 0%. The worst results are given by SPT and WSPT with a Makespan value of 1475, and a GAP of 59.3%. The average Makespan of all 7 heuristics is

1127 with a GAP of 21.7%. The Makespan obtained by proposed heuristic is 930, with a GAP of 0.43%.

4.1.2.3 For La-11

The best results are given by heuristics rules FCFS, and EDD with a Makespan value of 1272, and a GAP of 4.1%. The worst results are given by SPT and WSPT with a Makespan value of 1802, and a GAP of 47.5%. The average Makespan of all 7 heuristics is 1506 with a GAP of 23.2%. The proposed heuristic performs better against the same set of benchmark problem. The Makespan obtained by proposed heuristic is 1228, with a GAP of 0.49%.

4.1.2.4 For La-12

The best results are given by heuristics rules FCFS, and EDD with a Makespan value of 1039, and a GAP of 0%. The worst results are given by SPT and WSPT with a Makespan value of 1439, and a GAP of 38.5%. The average Makespan of all 7 heuristics is 1242 with a GAP of 19.5%. The Makespan obtained by proposed heuristic is 1052, with a GAP of 1.25%.

4.1.2.5 For La-26

The best results are given by heuristics rules LPT and MS with a Makespan value of 1394, and a GAP of 14.4%. The worst result is given by CR with a Makespan value of 2069, and a GAP of 69.9%. The average Makespan of all 7 heuristics is 1683 with a GAP of 38.1%. The proposed heuristic performs better against the same set of benchmark problem. The Makespan obtained by proposed heuristic is 1422, with a GAP of 16.75%.

4.1.2.6 Over-All Mean Gap

The over-all mean gap for the best performance heuristics is 6.28%. The over-all mean gap for the worst performance heuristics is 56.74%. The over-all mean gap of proposed heuristic is 4.26% which is better even than the best performance.

Table 4-3: Results of Lawrence benchmark problems P.H vs Dispatching Rules

test bed	LAWRENCE (LA) - 1984										Overall Mean Gap
Problem	La01	(Opt=666)	La06	(Opt=926)	La11	(Opt=1222)	La12	(Opt=1039)	La26	(Opt=1218)	
Instances	10x5	GAP%	15x5	GAP%	20x5	GAP%	20x5	GAP%	20x10	GAP%	
FIFO	772	15.90%	926	0.00%	1272	4.10%	1039	0.00%	1505	23.60%	8.72%
LPT	752	12.90%	926	0.00%	1300	6.40%	1167	12.30%	1394	14.40%	9.20%
SPT	1122	68.50%	1475	59.30%	1802	47.50%	1439	38.50%	1993	63.60%	55.48%
CR	979	47.00%	1140	23.10%	1792	46.60%	1401	34.80%	2069	69.90%	44.28%
EDD	865	29.90%	1024	10.60%	1272	4.10%	1039	0.00%	1430	17.40%	12.40%
MS	752	12.90%	926	0.00%	1300	6.40%	1167	12.30%	1394	14.40%	9.20%
WSPT	1122	68.50%	1475	59.30%	1802	47.50%	1439	38.50%	1993	63.60%	55.48%
Average	909	36.50%	1127	21.70%	1506	23.20%	1242	19.50%	1683	38.10%	27.80%
Minimum	752	12.90%	926	0.00%	1272	4.10%	1039	0.00%	1394	14.40%	6.28%
Maximum	1122	68.50%	1475	59.30%	1802	47.50%	1439	38.50%	2069	69.90%	56.74%
Proposed Heuristic	682	2.4	930	0.43	1228	0.49	1052	1.25	1422	16.75	4.264

4.1 Comparative Analysis (Proposed Heuristic Vs Hybrid Heuristics)

In this section, the proposed heuristic is compared with the Hybrid heuristic.

4.1.1 Fisher and Thompson Job Shop Benchmark Problems

The fisher and Thompson represented by FT are benchmark problems developed in 1963. The problems selected from this test-bed are FT06 (6 jobs, 6 machines), FT10 (10 jobs, 10 machines), and FT20 (20 jobs, 5 machines). The results are compared with traditional heuristics. The decision is made on overall mean GAP.

Table 4-4: Results of Fisher and Thompson P.H vs Hybrid Heuristic

Test Bed	Fisher and Thompson (1963)						Overall Mean Gap %
Problem	FT-06 (OPT=55)		FT-10 (OPT=930)		FT-20 (OPT=1165)		
Instances	6x6	GAP%	10X10	GAP%	20X5	GAP%	
HYB.H	61	10.9	1175	26.3	1613	34.8	24.0
P.H	<u>59</u>	7.3	<u>1136</u>	22.2	1670	38.5	22.6

The Makespan given by Hybrid heuristic for FT06 is 61 with a GAP from the optimum Makespan is 10.9%. The proposed heuristic gives a Makespan value 59 with a GAP of 7.3%. The Makespan given by Hybrid heuristic for FT10 is 1175 with a GAP from the optimum Makespan is 26.3%. The proposed heuristic gives a Makespan value 1136 with a GAP of 22.2%. The Makespan given by Hybrid heuristic for FT20 is 1670 with a GAP from the optimum Makespan is 34.8%. The proposed heuristic gives a Makespan value 1613 with a GAP of 38.5%. The over-all mean gap of hybrid heuristic from optimum values is 24%, and this gap for the proposed heuristic is 22.6%, which means the proposed heuristic performs better than the hybrid heuristic for FT benchmark problems.

4.1.2 Lawrence Job Shop Benchmark Problems

The Lawrence benchmark problems are represented by LA and are developed in 1984 [24]. The problems selected are LA-01 (10 jobs, 5 machines), LA-06 LA-11, LA-12 and LA-26 and many more. The results are compared with Hybrid heuristics shown in table 5.5. The decision is made on overall mean GAP.

It is concluded from the comparative analysis that in some problems like LA 6, 7, 21, 25 etc. the hybrid heuristic performs better, while in other problems like LA 1, 2, 3 etc. the performance of proposed heuristic is better. In some cases, like LA 19 and 23 the results of both heuristics ties, while in some cases like LA 5, 10 and 14 it achieves the global Makespan. The over-all mean gap% for Hybrid heuristic is 11.45%, while for proposed heuristic it is 10.73%, which means the proposed heuristic performs better than hybrid heuristic.

Table 4-5: Results of Lawrence problems, proposed heuristic vs hybrid heuristic

Lawrence Benchmark Problems						
Problems code	**Problem size**	**Optimum**	**Hyb. H**	**%GAP**	**P. Heur**	**%GAP**
1	10x5	666	700	5.11	<u>682</u>	2.40
2	10x5	655	808	23.36	<u>774</u>	18.17
3	10x5	597	726	21.61	<u>699</u>	17.09
5	10x5	593	593	0.00	<u>593</u>	0.00
6	15x5	926	926	0.00	930	0.43
7	15x5	890	976	9.66	994	11.69
10	15x5	958	958	0.00	<u>958</u>	0.00

11	20x5	1222	1272	4.09	1228	0.49
12	20x5	1039	1039	0.00	1052	1.25
13	20x5	1150	1153	0.26	1166	1.39
14	20x5	1292	1292	0.00	1292	0.00
15	20x5	1207	1466	21.46	1422	17.81
17	10x10	748	907	21.26	902	20.59
18	10x10	848	988	16.51	981	15.68
19	10x10	842	968	14.96	968	14.96
21	15x10	1046	1265	20.94	1278	22.18
23	15x10	1032	1130	9.50	1130	9.50
25	15x10	977	1215	24.36	1243	27.23
27	20x10	1235	1538	24.53	1520	23.08
Over all Mean Gap %				**11.45**		**10.73**

4.1.3 Applegate & Cook Job Shop Benchmark Problems

The proposed heuristic is also tested against the Applegate and Cook benchmark problems [25]. The results are compared with the Hybrid heuristic and decision is made based on over-all mean gap. The results are given in the table 4.6

It is concluded from the comparison table that for ORB1 and ORB4 the hybrid heuristic gives better results, while for ORB2 and ORB3 the proposed heuristic provides better results. The over-all mean gap for proposed heuristic is 20.56%, which is better than hybrid heuristic.

Table 4-6: Results of Applegate and Cook problems, proposed heuristic vs hybrid heuristic

APPLEGATE and COOK (1991)						
Problem code	**Problem size**	**Optimum**	**Hyb.H**	**%GAP**	**P.H**	**%GAP**
ORB1	10X10	1059	1251	18.13	1255	18.51
ORB2	10X10	1050	1365	30.00	1330	26.67
ORB3	10X10	1005	1225	21.89	1178	17.21
ORB4	10X10	887	1013	14.21	1063	19.84
Over-all Mean GAP %				**21.06**		**20.56**

4.1.4 Taillard Job Shop Benchmark Problems

The proposed heuristics is tested against Taillard benchmark problems [26]. The results are compared with the Hybrid heuristic and decision is made based on over-all mean gap. The results are given in the table 4.7.

It is concluded from the results table that the over-all mean gap% given by proposed heuristic is better than the hybrid heuristic.

Table 4-7: Results of Taillard problems, proposed heuristic vs hybrid heuristic

TAILLARD (1993)						
Problem code	**Problem size**	**Optimum**	**Hyb.H**	**%GAP**	**P.H**	**%GAP**
TA11	**20X15**	**1364**	**1688**	**23.75**	**1709**	**25.29**
TA12	**20X15**	**1367**	**1657**	**21.21**	**<u>1652</u>**	**20.85**
TA13	**20X15**	**1350**	**1798**	**33.19**	**<u>1730</u>**	**28.15**
Over-all Mean GAP %				**26.05**		**<u>24.76</u>**

4.2 Comparative Analysis of Flow Shop Problems

The flow shop benchmark problems were developed by carlier in 1993 [27]. These problems are represented by code CAR, and are of different dimensions ie 11x5, 13x4, 12x5 7x7 etc. The strength of proposed heuristic is tested against carlier benchmark problems. The dispatching rules selected for this comparative analysis are FCFS, SPT, and LPT. Decision is made on the basis of minimum makespan. The results of dispatching rules and proposed heuristic are shown in the table 4.8.

Table 4-8: Results of Carlier flow shop benchmark problems, proposed heuristic vs FCFS, LPT, and SPT

CARLIER FLOW SHOP BENCHMARK PROBLEMS					
Code	**Size**	**FCFS**	**LPT**	**SPT**	**PH**
CAR 1	**11x5**	**9298**	**10649**	**7718**	**<u>6583</u>**

CAR 2	**13x4**	**8665**	**10330**	**7741**	**7302**
CAR 3	**12x5**	**10122**	**10973**	**8237**	**6680**
CAR 4	**14xx4**	**9991**	**11274**	**8679**	**7120**
CAR 5	**12x5**	**9311**	**11804**	**9313**	**6964**
CAR 6	**8x9**	**11579**	**10858**	**9989**	**6739**
CAR 7	**7x7**	**8170**	**8228**	**7012**	**6022**
CAR 8	**8x8**	**9963**	**10453**	**9578**	**6747**

4.2.1 CAR1

For Carlier (CAR1) problem, among the three dispatching rules, the best performance is shown by SPT with a Makespan value of 7718. The worst performance is shown by LPT with a Makespan value of 10649. The Makespan reported by proposed heuristic is 6583, which is a much better value.

4.2.2 CAR2

For Carlier (CAR2) problem, among the three dispatching rules, the best performance is shown by SPT with a Makespan value of 7741. The worst performance is shown by LPT with a Makespan value of 10330. The Makespan reported by proposed heuristic is 7302, which is a much better value.

4.2.3 CAR3

For Carlier (CAR3) problem, among the three dispatching rules, the best performance is shown by SPT with a Makespan value of 8237. The worst performance is shown by LPT with a Makespan value of 10973. The Makespan reported by proposed heuristic is 6680, which is a much better value.

4.2.4 CAR4

For Carlier (CAR4) problem, among the three dispatching rules, the best performance is shown by SPT with a Makespan value of 8679. The worst performance is shown by LPT with a Makespan value of 11274. The Makespan reported by proposed heuristic is 7120 which is a much better value.

4.2.5 CAR5

For Carlier (CAR5) problem, among the three dispatching rules, the best performance is shown by FCFS with a Makespan value of 9311. The worst performance is shown by LPT with a Makespan value of 11804. The Makespan reported by proposed heuristic is 6964 which is a much better value.

4.2.6 CAR6

For Carlier (CAR6) problem, among the three dispatching rules, the best performance is shown by SPT with a Makespan value of 9989. The worst performance is shown by FCFS with a Makespan value of 11579. The Makespan reported by proposed heuristic is 6739 which is a much better value.

4.2.7 CAR7

For Carlier (CAR7) problem, among the three dispatching rules, the best performance is shown by SPT with a Makespan value of 7012. The worst performance is shown by LPT with a Makespan value of 8228. The Makespan reported by proposed heuristic is 6022 which is a much better value.

4.2.8 CAR8

For Carlier (CAR8) problem, among the three dispatching rules, the best performance is shown by SPT with a Makespan value of 9578. The worst performance is shown by LPT with a Makespan value of 10453. The Makespan reported by proposed heuristic is 6747 which is a much better value.

In this chapter, the results of the proposed heuristic are compared with other heuristics. The proposed heuristic can be used for both job shop and flow shop problems. Different benchmark problems are utilized in this chapter ranging from simple to complex. The job shop benchmark problems include: Fisher and Thompson problems, Lawrence problems, Applegate and Cook problems, and Taillard problems. The flow shop benchmark problems include: Carlier problems. The optimum results, the results of traditional heuristics, and results of hybrid heuristic for these benchmark problems are taken from literature. The results of proposed heuristic are find out. The comparative

analysis tables are produced for each set of benchmark problems and the results of proposed heuristic and other heuristics are compared. The results of proposed heuristic that are better than other heuristics are underlined. The overall mean gap has been used as a criterion for accepting or not accepting the results. The overall mean gap for every set of problems is calculated for both proposed heuristic as well as other heuristics. The overall gap is minimum for proposed heuristic, which the proposed heuristic performs better than other heuristics.

CHAPTER 5: CONCLUSION AND FUTURE WORK

The main objective of this research was to develop a new heuristic that can be applied to flow shop problems and job shop problems to minimize Makespan. For these different existing heuristics are studied in detail. A new heuristic was developed which is discussed in chapter 4 in detail. The heuristic is coded in MATLAB. Decision regarding the strength of heuristic is made based on mean Gap. Gap is the deviation of the Makespan value obtained by a particular heuristic from the optimum Makespan value. A comparative analysis is carried out by comparing the results of proposed heuristic with other heuristics and the proposed heuristic gives better results as compared to others. This chapter summarizes the achievements of the research and recommendations regarding future work.

The proposed heuristic is coded in MATLAB. The process plan and processing times are stored in excel sheet. From excel these data are imported into MATLAB as an input to the proposed heuristic.

5.1 Achievements of the Research

The objectives that are mentioned in chapter 1 are step wised successfully achieved. The first three chapters cover the theory related to scheduling, different techniques of solving flow shop and job shop problems, different scheduling environments, and benchmark problems and its complexity. In chapter 4 the proposed heuristic is explained in detail by solving an example problem. In chapter 5 the results of proposed heuristic are compared with other heuristics and it was concluded that the proposed heuristic performs better than other heuristics.

5.2 Proposed Heuristic and its Strength

The proposed heuristic uses eight sequences to find the Makespan value. The heuristic is coded in MATLAB and is applied to distinct set of benchmark problems. The results are compared with other heuristics. The strength of any heuristic lies in its ability to hit the optimum or near optimum value. The proposed heuristic in some cases hit the

optimum value and in some cases, it hits the near optimum value. Some of these cases are listed below in table 5.1.

Table 5-1: Some of the optimum or near optimum Makespan values achieved by proposed heuristic

Selected from Lawrence Benchmark Problems				
Proposed Heuristic Achieve Optimum Makespan Value				
Problem Code	**Problem Size**	**Optimum Value**	**P. Heuristic**	**Gap%**
LA-5	**10X5**	**593**	**593**	**0**
LA-10	**15X5**	**958**	**958**	**0**
LA-14	**20X5**	**1292**	**1292**	**0**
Proposed Heuristic Achieve Near Optimum Makespan Value				
LA-1	**10X5**	**666**	**682**	**2.4**
LA-6	**15X5**	**926**	**930**	**0.43**
LA-11	**20X5**	**1222**	**1228**	**0.49**

5.3 Future Work

The performance of proposed heuristic was good across distinct set of benchmark problems and successfully achieved optimum or near optimum Makespan value. Still some limitations that are given below can be worked out in future.

i) **Scheduling problems are of multiple objectives. The heuristic that is developed in this research is limited only to Makespan. There are some other performance criteria like lateness, machine utilization etc. that can be used as an objective function.**

ii) **The proposed heuristic is applied to deterministic scheduling environment which is based on assumptions. These assumptions are not acceptable in real scheduling environment, so in future the proposed heuristic can be extended to dynamic environment to deal with real scheduling situations.**

iii) **The proposed heuristic can be used as an initial solution to a genetic algorithm. It uses only three sequences and hitting optimum or near optimum values so, if this heuristic is used as starting solution to a genetic algorithm it will converge to optimum value in less time.**

This chapter describes the conclusion of the whole project and the future recommendations. The conclusion is that the objectives that are stated at the start of the project are successfully achieved. And at the end the future recommendations are to extend the proposed heuristic, used it for multi-objectives, dynamic environment, and due to its ability of hitting optimum or near optimum values can also be used as a starting solution to a genetic algorithm.

APPENDIX A

Appendix A contains MATALAB pseudocode of proposed heuristic.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Define Global Variable
Set Global Variables= "ProcessingTimes", "MachinesConstraints";
Get machine_constraints from excel_file= "ProcessingPlan";
Get Processing_time from excel_file= "Processing time";
Set [total number of jobs,TotalNoOfMachines]=size(MachineConstraints);
Set N=TotalNoOfJobs;
Set M=TotalNoOfMachines;
Set q=0;
%Begin For_Loop
For (i=1;i=N;i++)
Set q=q+1 ; %Processing time of every job
End_for_loop;

Set e=transpose(q);
Set [X1,idX1]=e in Descending order;

Set [X2,idX2]=e in ascending order;
Set p771=transpose(X1);
Set p772=transpose(X2);
%% Sequence_1 rank from Highest to Lowest %%%
Set seq_1=idx1;
Set sequence=seq_1;
Set Cmax=Machine_Assignment(sequence);
set c1=Cmax;
%%%Sequence_2 rank from Lowest to Highest %%%
Set seq_2=idx2;
Set sequence=seq_2;
Set Cmax=Machine_Assignment(sequence);
Set c2=Cmax;
%%%Sequence_3 based on SPT-LPT %%% 50/50
%%%Sequence_4 based on LPT-SPT %%% 50/50
%Begin If-Else
If N is even;

Set A_p0= first 50%of N;% to SPT
Set D_p0= last 50% of N; %to LPT

Set A_p0=transpose(A_p0);
Set D_p0=transpose(D_p0);

Set Seq_3=[A_p0:D_p0] %concatenated
Set Seq_4=[D_p0:A_p0] %concatenated

Elseif N is not even
Set A_p0= first 50%of N;% to SPT
      Set D_p0= last 50% of N; %to LPT
Set P= P[(N+1)/2];
```

```
Set A_p0=transpose(A_p0);
Set D_p0=transpose(D_p0);

      Set Seq_3=[A_p0:D_p0:p]; %concatenated
      Set Seq_4=[D_p0:A_p0:p]; %concatenated

Endifelse

set sequence=Seq_3;
set Cmax = Machine_Assignment(sequence);
set c3=Cmax;
set sequence=Seq_4;
set Cmax=Machine_Assignment(sequence);

set c4=Cmax;

%%% Sequence_5 based on  SPT-LPT-SPT-LPT %%% 25/25/25/25 Percent
%%% Sequence_6 based on  LPT-SPT-LPT-SPT %%% 25/25/25/25 Percent
Begin
If N is multipleof4

Set A_p0= first 25%of N; % (0%-25%)   SPT
Set D_p0= Next 25% of N; % (26%-50)   LPT
Set A_p= Next 25%of N;   % (51%-75%)  SPT
Set D_p= last 25% of N;  % (76%-100%) LPT


Set A_p0=transpose(A_p0);     %concatenated
Set D_p0=transpose(D_p0);     %concatenated
Set A_p=transpose(A_p); %concatenated
Set D_p=transpose(D_p); %concatenated


Set Seq_5=[A_p0:D_p0:A_p:D_p]
Set Seq_6=[D_p0:A_p0:D_p:A_p]

Elseif N is not multipleof4


Set A_p0= first 25% of N; % (0%-25%)   SPT
Set D_p0= Next 25% of N;  % (26%-50)   LPT
Set A_p = Next 25% of N;   % (51%-75%)  SPT
Set D_p = last 25% of N;  % (76%-100%) LPT

Set p = remaining elements of e

Set A_p0=transpose(A_p0);
Set D_p0=transpose(D_p0);
Set A_p=transpose(A_p);
Set D_p=transpose(D_p);
set p=transpose(p)
Set Seq_5=[A_p0:D_p0:A_p:D_p:p]
Seq_6=[D_p0:A_p0:D_p:A_p:p]
Set sequence=Seq_5;
Set Cmax=Machine_Assignment(sequence);
Set c5=Cmax;
```

```
Set sequence=Seq_6;
Set Cmax=Machine_Assignment(sequence);
Set c6=Cmax;
%%% Sequence_7 based on  LPT-SPT and so on %%%
% First to arrange values in p with ascend and descend sequence
Set x2= transpose(x2);
Set idx2= transpose(idx2);
Set x1= transpose(x1);
Set idx1= transpose(idx1);
Set p=zeros(N,2);
Set c772=0;
Set c771=0;
for i=1: size(p,1);
Set i77= mod(i,2);
      if i77 is not equal to zero
      Do
      set p(i,1)= p771(i-c771,1);
      set p(i,2)= p771(i-c771,2);
      set c772=c772+1;


      elseif i77 is equal to 0);
      Do
      Set p(i,1)= p772(i-c772,1);
      set p(i,2)= p772(i-c772,2);
      set c771=c771+1;
endif
endfor
set Seq_7=p(:,2)';
set sequence=Seq_7;
set Cmax=Machine_Assignment(sequence);
set c7=Cmax;
%%
%%% sequence_8 based on  SPT-LPT.....and so on %%%
Set x2= transpose(x2);
Set idx2= transpose(idx2);
Set x1= transpose(x1);
Set idx1= transpose(idx1);
Set p=zeros(N,2);
Set c772=0;
Set c771=0;
for i=1: size(p,1);
Set i77= mod(i,2);
if i77 is not equal zero ;
      Do
      Set p(i,1)= p772(i-c771,1);
      Set p(i,2)= p772(i-c771,2);
      set c772=c772+1;
      endDo

if i77 is equal to zero;
      Do
      set p(i,1)= p771(i-c772,1);
      set p(i,2)= p771(i-c772,2);
      set c771=c771+1;
      endDo
endif
```

```
set Seq_8=p(:,2)';
set sequence=Seq_8;
set Cmax=Machine_Assignment(sequence);
set c8=Cmax;

Final_makespan=min([c1,c2,c3,c4,c5,c6,c7,c8])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
```

REFERENCES

[1] T. MELTON, "The Benefits of Lean Manufacturing: What Lean Thinking has to Offer the Process Industries," *Chemical Engineering Research and Design,* vol. 83, no. 6, pp. 662-673, June 2005.

[2] Z. M. &. J. F. Xiaoning Shen, "Multi-Objective Dynamic Job Shop Scheduling: A survey and Prospects," *International Journal of Innovative Computing, Information and Control,* vol. 10, no. 6, pp. 2113-2126, 2014.

[3] A. S. &. M. S. Jain, "Deterministic job-shop scheduling: Past, present and future," *European journal of operational research,* vol. 113, no. 2, pp. 390-434, 1999.

[4] S. &. K. M. K. Noor, "Hybrid scheduling system for flow shop and job shop scheduling problem," in *23 rd International Conference of CAD/CAM, Robotics & Factories of the Future Conference.,* Bogota, Colombia., 2007.

[5] B. &. S. B. Roy, "Les problemes d'ordonnancement avec contraintes disjonctives. Note ds, 9.," Paris, France, 1964.

[6] J. B. E. &. Z. D. Adams, "The shifting bottleneck procedure for job shop scheduling. „" *Management science,* vol. 34, no. 3, pp. 391-401, 1988.

[7] A. R. L. C. &. S. A. T. Jones, "Survey of job shop scheduling techniques.," 1999.

[8] M. &. C. X. Pinedo, Operations scheduling, McGraw Hill, 1999.

[9] J. E. K. &. T. D. Błażewicz, "Recent advances in scheduling in computer and manufacturing systems," *European Journal of Operational Research,* vol. 164, no. 3, pp. 573-574, 2005.

[10] M. S. Morshed, "A hybrid model for job shop scheduling," *Doctoral dissertation, University of Birmingham,* 2006.

[11] R. &. W. C. Zhang, "A hybrid approach to large-scale job shop scheduling," *Applied intelligence,,* vol. 32, no. 1, pp. 47-59, 2010.

[12] V. a. S. R. Vinod, "Dynamic job shop scheduling with sequence dependent setup time," *Advance Manufacturing Technology,* vol. 36, pp. 355-360, 2008.

[13] I. M. Alharkan, Algorithms for Scheduling & Sequencing.

[14] A. T. E. M. D. W. Pentico, Heuristic Scheduling System, New York, 1993.

[15] K. R. Baker, Introduction to scheduling and sequencing, New York: John Wiley and sons, 1974.

[16] .. M. a. W. ,. L. a. M. C. R., Theory of scheduling, 1967.

[17] R.Haupt, "A survey of priority rule-based scheduling," *OR Spectrum,* vol. 11, pp. 3-16, 1989.

[18] B. N. B. S. D. Goldsman, "Methods for selecting the best system (for simulation)," in *Winter Simulation Conference,* Phoenix, AZ, USA, USA, 1991.

[19] M. K. K. A. S. W. S. Maqsood, "Novel Heuristic for Low-Batch Manufacturing Process Scheduling Optimization with Reference to Process Engineering," *Chemical Product and Process Modelling,* vol. 6, no. 2, 2011.

[20] Panwalker, "A Survey of Scheduling Rules," *Operation Research* , pp. 45-61, 1977.

[21] L. L. a. K. Graham, "Optimization And Approximations in Deterministic Sequencing and Scheduling," *Operation Research,* pp. 287-326, 1979.

[22] M. L. Pinedo, Scheduling (Theory, Algorithms, and Systems), Springer, Cham.

[23] H. Fisher, " Probabilistic learning combinations of local job-shop scheduling rules.," *Industrial scheduling,* pp. 225-251, 1963.

[24] S. Lawrence, "Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement)," 1984.

[25] D. &. C. W. Applegate, "A computational study of the job-shop scheduling problem," *ORSA Journal on computing,* vol. 3, no. 2, pp. 149-156, 1991.

[26] E. Taillard, "Benchmarks for basic scheduling problems," *european journal of operational research,* vol. 64, no. 2, pp. 278-285, 1993.

[27] J. &. P. É. Carlier, "An algorithm for solving the job-shop problem," *Management science,* vol. 35, no. 2, pp. 164-176, 1989.