FLOW SHOP SCHEDULING USING DUAL – BOTTLENECK APPROACH

SITI NASUHA BINTI MOHD NASIR

A project report submitted as a partial
fulfilment of the requirement for the award of the
Degree of Master of Mechanical Engineering

Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia

January, 2015

ABSTRACT

In general, scheduling is a key factor for manufacturing productivity. It can have a major impact on the productivity of a process. The purposes of scheduling are to maximize the efficiency of the operation, minimize the production time and reduce the costs. This study discusses about flow shop scheduling problem, which is one of the most well-known problems in the area of scheduling. It is a production planning problem in which n jobs have to be processed in the same sequence on m machines. Most of these problems concern the objective of minimizing makespan. This study focused on developing a new scheduling algorithm for six machines, flow shop scheduling where two of the processes have a high tendency of dominant bottleneck characteristics. The scheduling problem resembles six machine flow shop scheduling of $P1$, $P2$, $P3$, $P4$, $P5$ and $P6$, where $P1$ and $P6$ are emphasized as the dual dominant machines. This study also evaluate the performance of the dual-bottleneck approach algorithm against Campbell *et al*. algorithm and optimum solution from complete enumeration using Visual Basic for Application in Microsoft Excel. 100 sets of simulations data were randomly generated to six jobs problem by using this new dual bottleneck approach algorithm. The experimental results are divided into two rules which consist of three major groups of dominance level range to produce the best makespan of job sequence. The result analysis showed that dual-bottleneck approach produces a better result at rule $P1 < P6$ compared to Campbell *et al*. algorithm. Meanwhile, at rule $P1 > P6$ dual-bottleneck approach algorithm produced results worse than Campbell *et al*. algorithm.

ABSTRAK

Secara umumnya, penjadualan adalah faktor utama bagi produktivi pembuatan. Ia boleh memberikan kesan yang besar terhadap produktivi sesuatu proses. Matlamat penjadualan adalah untuk memaksimumkan kecekapan operasi, mengurangkan masa pengeluaran dan mengurangkan kos. Kajian ini membincangkan tentang masalah penjadualan aliran (*flow shop*), yang merupakan salah satu masalah yang paling terkenal dalam bidang penjadualan. Ia adalah masalah perancangan pengeluaran di mana *n* kerja perlu diproses dalam urutan yang sama pada *m* mesin. Kebanyakan masalah adalah berkaitan dengan objektif untuk meminimumkan *makespan*. Kajian ini memberi tumpuan kepada membangunkan algoritma penjadualan baru untuk aliran enam mesin di mana dua daripada prosesnya mempunyai kecenderungan tinggi memenuhi ciri-ciri *bottleneck* yang dominan. Masalah penjadualan ini digambarkan sebagai aliran enam mesin *P*1, *P*2, *P*3, *P*4, *P*5 dan *P*6, di mana *P*1 dan *P*6 merupakan mesin dwi-dominan. Kajian ini juga menilai prestasi pendekatan algoritma dwi-dominan berbanding algoritma Campbell *et al.* dan penyelesaian optimum dari penghitungan (*enumeration*) lengkap menggunakan *Visual Basic for Application* dalam Microsoft Excel. 100 set data simulasi rawak dijana untuk enam masalah kerja dengan menggunakan kaedah algoritma dwi-dominan baru ini. Keputusan eksperimen dibahagikan kepada dua peraturan yang terdiri daripada tiga kumpulan utama bagi julat tahap dominan untuk menghasilkan *makespan* yang terbaik mengikut urutan kerja. Analisis hasil kajian menunjukkan bahawa kaedah dwi-dominan menghasilkan keputusan yang lebih baik pada peraturan $P1 < P6$ berbanding dengan Campbell *et al*. algoritma. Sementara itu, pada peraturan $P1 > P6$ kaedah algoritma dwi-dominan menghasilkan keputusan yang kurang baik daripada algoritma Campbell *et al.*.

CONTENTS

CHAPTER 2 LITERATURE REVIEW

CHAPTER 3 METHODOLOGY

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

GT	-	Group Technology
MRP	-	Material Requirement Planning
MRP II	-	Manufacturing Resources Planning
NEH	-	Nawaz, Enscore and Han
CDS	-	Campbell, Dudek and Smith
DAN	-	Dannenbring
$P1$	-	Process 1
$P2$	-	Process 2
$P3$	-	Process 3
$P4$	-	Process 4
$P5$	-	Process 5
$P6$	-	Process 6
BM	-	Bottleneck Machine
i	-	Process sequence of the job
j	-	Number according to the scheduling sequence ($j = 1, 2, 3 \ldots . n$)
k	-	Sub-problem
NP	-	Non-Deterministically Polynomial
VBA	-	Visual Basic for Application
WIP	-	Work-In-Progress
HDD	-	Hard disk device
FFL	-	Flexible Flow Line
FCFS	-	First-come, first served
SPT	-	Shortest processing time
EDD	-	Earliest due date
CR	-	Critical ratio
BAM	-	Bottleneck Adjacent Matching
BAM2	-	Bottleneck Adjacent Matching 2
BAM4	-	Bottleneck Adjacent Matching 4
BSP	-	Bottleneck Scheduling Performance
JIT	-	Just in Time

BTS	-	Basic Tabu Search
BBFFL	-	Bottleneck-Based Flexible Flow Line
BBISG	-	Bottleneck-Based Initial Sequence Generator
BBMIP	-	Bottleneck-Based Multiple Insertion Procedure
BN	-	Bottleneck Value

LIST OF APPENDIX

CHAPTER 1

INTRODUCTION

1.1 Introduction

In general, scheduling is a key factor for manufacturing productivity. It can have a major impact on the productivity of a process. The purposes of scheduling are to maximize the efficiency of the operation, minimize the production time and reduce the costs. This operation is done by telling a production facility what to make, when and on which machine (Conway, Maxwell & Miller, 1967). These machines are assumed to be set up in series and the environment is referred to as flow shop. Although, the flow shop manufacturing is used widely in production system but it is known that it is not an easy task to finding an optimal solution for flow shop scheduling problem. This study explored and investigated a manufacturing process scheduling which resembles a six machines flow shop. The study developed a new flow shop scheduling using dual - bottleneck approach.

1.2 Background of the Study

Scheduling is a decision-making process that concerns the allocation of limited resources to a set of tasks with the view of optimizing one or more objectives. In today's world of global competition, effective scheduling has become vital in order to meet customer requirements as promptly as possible while maximizing the profits. Hence, it plays an important role in most manufacturing and production system as well as in most information processing environments (Pinedo, 2008).

Different forms of resources and tasks in manufacturing systems or service industries can result in different classifications of scheduling. As Pinedo states, the resources can take many forms such as machines in a workshop, crews of an airplane or a ship or processing units in a network of computers. The tasks may be operations on an assembly line, take-offs and landings at an airport or phases of a construction project. Therefore, according to the different types of resources and tasks, and also considering the technological constraints that exist, various classical scheduling problems can be defined and formulated, such as flow-shop scheduling, job shop scheduling and open shop scheduling (Pinedo, 2008).

Flow shop manufacturing is a very common production system in many manufacturing facilities, assembly lines and industrial processes (Bareduan & Hasan, 2010). In this environment, the operations of all jobs must follow the same order following the same route along the machines assumed to be set up in the series. It can be briefly described as follows; there are a set of m machines and a set of n jobs. Each job comprises a set of m operations which must be done on different machines. All jobs have the same processing operation order when passing through the machines.

It is known that finding an optimal solution for a flow shop scheduling problem is a difficult task and even a basic problem involving six machines is already NP-hard (Pinedo, 2008). Therefore, many researchers have concentrated their efforts on finding near optimal solutions within acceptable computation time using heuristics. Most heuristics are developed by the researchers after gaining familiarity and in-depth understanding of the system's characteristic or behavior (Bareduan & Hasan, 2010).

In scheduling, the "bottleneck" in the processing is the main problems concerned with the manufacturing and process industries. A bottleneck is a constraint within the system that limits throughput. A bottleneck may be a machine, scarce or highly skilled labor or specialized tools. Many researchers in production and operation management have come out with various heuristic with estimated optimal value to solve the scheduling problem of interest.

Heuristic are general guidelines for obtaining feasible but not necessarily optimal solution to problems. Heuristic is developed by considering the work center that may be a single machine and group of machines or an area where a particular type of work in done or by product in a flow, assembly line or group technology-cell (GT-cell) configuration. Therefore, in current manufacturing world, the optimal heuristic is needed in order to minimize the effect of the bottleneck. This means, it will intend to minimize the time it takes to do work, or specifically, the makespan in flow shop. The makespan is defined as the amount of time from start to finish completing a set of multi-machine jobs where machine order is pre-set for each job (Pinedo, 2008).

The primary reasons for the use of bottleneck approach in many studies on scheduling are the associated ease in feasible schedules and implementing as well as control the process planning at the job and flow shop level in real life situation. Thus, the current static flow shop scheduling problem has been chosen to study with the stated assumptions and hence, the development of heuristics scheduling is considered in the flow shop with bottleneck machines. The common measurements of performances are the minimization of makespan and total flow time of jobs. The selection of bottleneck-based heuristic and the proposed dual bottleneck-based heuristics are extensively investigated for the performance by generating a large number of problems with specific bottleneck conditions.

1.3 Problem Statement

The *n* job with m machine flow shop scheduling is a Non-Deterministically Polynomial (NP) Hard problem. Optimal solutions can only be obtained by enumeration techniques. But these methods take a large amount of computational effort and time (Bareduan & Hasan, 2012). That is why heuristic method is developed to solve these problems. Independent research has indeed confirmed that heuristic evaluation is a very efficient usability engineering method.

As continuation work from the literature, this study is directed towards developing a new heuristic for solving the six machines flow shop scheduling problem. It involves the development of a new algorithm based on dual - bottleneck approach and analyzes the performance of the dual - bottleneck algorithm for six jobs problem in flow shop scheduling.

1.4 Objectives of the Study

The objective of this study is to:

i. Develop a scheduling algorithm based on the dual - bottleneck approach for six machines flow shop scheduling.
ii. Evaluate the performance of the dual-bottleneck approach algorithm against Campbell *et al*. algorithm and optimum solution from complete enumeration using Visual Basic for Application in Microsoft Excel.

1.5 Scope of the Study

The following are the scopes in conducting this study:

i. This scheduling is specifically for flow shop scheduling.
ii. The number of machine in the flow shop is limited to six machines.
iii. The problem analysis will be done for six jobs problem.
iv. The algorithm will be developed and tested using Microsoft Excel and Visual Basic for Application (VBA) programming.
v. The performance of the algorithm will be measured using makespan criteria and tested with randomly data generates from VBA in Microsoft Excel.

1.6 Significance of the Study

In order to remain competitive in current global environment, enterprises must be competent in certain areas such as short product lifecycle, product varieties, minimal inventories, concurrent processing of different products and short delivery times. The main objective in the scheduling system is to decrease the processing time of products so that the products could be delivered to customers on time.

The previous research has found several ways in developing scheduling heuristic using bottleneck approach and Visual Basic of Application (VBA) programming in Microsoft Excel. The good thing about this method is there is no high skilled person required and it involves low cost in developing the scheduling. The programs are flexible enough which allow user to modify the existing scheduling data and can easily be understood. Hence, the previous research should be continued because it can give big impact on the productivity of such companies. This cheap and easy to understand method should be very useful for small companies to save budget and time while productivity can be increased.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Generally, scheduling is a form of decision-making that plays a crucial role in manufacturing and service industries. In the current competitive environment, effective scheduling has become a necessity for survival in the marketplace (Pinedo, 2008). Scheduling can be defined as an allocation of limited resources to tasks over time. The resources may be machines in a workshop, runways at an airport, and crews at a construction site, processing units in a computing environment, and so on. In manufacturing, the purpose of scheduling is to minimize the production time and costs by developing schedule with the optimal solution.

In the scheduling, it shows the process of converting outline plans into a time-based graphic presentation given information on available resource and time constraints. Production scheduling tools greatly outperform older manual scheduling methods (Baker, 1974). These provide the production scheduler with powerful graphical interfaces which can be used to visually optimize real-time workloads in the various stages of production and pattern recognition allow the software to automatically create scheduling opportunities which might not be apparent without view into the data.

The objectives can also take many different forms. One objective may be the minimization of the completion time of the last task and another may be the minimization of the number of tasks completed after their respective due dates.

2.2 The Concept of Scheduling

In scheduling, the limited resources consist of one or more machines, and tasks are modeled as jobs that can be executed by the machines. A task (job) first becomes available for processing at its ready time, and it must receive the amount of processing equal to its processing time. Typically, a problem in scheduling is characterized by the types of machines and jobs in the system, by the constraints imposed, and by a desired optimality principle (Szwarc, 1973).

The machine environment characteristic is that a machine can handle, at most, one job at a time, and each job can be processed by only one machine at a time. Generally, a machine can begin its next job immediately after the current job is completed, and there are no machine breakdowns at any moment of time. For the scheduling problem considered in this project, preemption is not allowed during the processing of any operation, which means that the execution of a job on a machine will proceed without interruption once it starts. The machine scheduling problem is in fact a sequencing problem where a schedule is completely specified by the sequence in which jobs are performed.

The purpose of scheduling in manufacturing industry is to minimize the production time and costs, by telling a production facility what to do, with which staff and on which equipment. The aim of production scheduling is to maximize the efficiency of the operation and reduce costs. By comparing to manual scheduling methods, the production scheduling tools are great. The production scheduler can provide graphical interfaces which can be used to visually optimize real-time workloads in various stages of the production.

2.3 Scheduling Classification

The companies use backward and forward scheduling to allocate plant and machinery resources, plan human resources, plan production processes and purchase materials. Forward scheduling is planning the tasks from date resources become available to determine the shipping date or the due date. Otherwise, backward scheduling is planning the tasks from the due date or required by date to determine the start date or any changes in capacity required (Conway *et al*., 1967).

2.3.1 Forward Scheduling

Forward scheduling can be defined as a schedule ahead from a point in time, through taking job with a number of tasks and allocates those tasks to resources as early as possible when the resources allow. The first available time that the resource is available to be used the task should make use of it. Hence, the forward scheduling enables the scheduler to determine the earliest possible completion time for each job and thus, the amount of lateness or the amount of slack can be determined (Stevenson, 2012).

2.3.2 Backward Scheduling

Backward scheduling is also known as pull scheduling where it is a method of determining a production scheduling by working backwards from the due date to the start date and computing the materials and time required at every operation or stage. The example using the backward system are material requirement planning (MRP) and manufacturing resources planning (MRP II).

This method is more complicated than forward scheduling because the possibility of infeasibility caused by creating jobs that should have been started yesterday or even earlier. If the resultant schedule is not feasible, the loading sequences in a backward schedule need to be changed (Bareduan & Sulaiman, 2004).

2.4 Scheduling Criteria

Scheduling in the right technique depends on the volume of orders, the nature of operations, and the overall complexity of jobs, as well as the importance placed on each of four criteria. Those four criteria are (Heizer and Render, 1999):

i. Minimize completion time – This criterion is evaluated by determining the average completion time per job.
ii. Maximize utilization – This is evaluated by determining the percent of time the facility is utilized.
iii. Minimize work-in-progress (WIP) inventory – This is evaluated by determining the average number of jobs in the system. The relationship

between the number of jobs in the system and WIP inventory will be high. Therefore, the fewer the number of jobs that are in the system, the lower the inventory.

iv. Minimize customer waiting time – This is evaluated by determining the average number of late days.

2.5 Flow Shop Scheduling Problem

In many manufacturing and production systems, jobs have to be processed by several machines in a given order. This multi-operation simulation is often called a shop scheduling model, where a number of jobs are to be processed in a shop consisting of several machines. The shop scheduling models are divided into two types of model that is flow-shop model and job-shop model. In the aforementioned shop models, there are no precedence relationships between jobs prescribing the order in which job processing must be carried out (Krajewski and Ritzman, 2005). While the machine sequence of all jobs is given, the scheduling problem is to find the best job processing sequence according to a desired optimality principle.

Flow shop scheduling problem is one of the most well-known problems in the area of scheduling. It is a production planning problem in which *n* jobs have to be processed in the same sequence on *m* machines. Most of these problems concern the objective of minimizing makespan. Makespan is the time between the beginning of the execution of the first job on the first machine and the completion of the execution of the last job on the last machine. To minimize the makespan is equivalent to maximize the utilization of the machines (Pinedo, 2008).

Johnson in 1954 is the pioneer in the research of flow shop problems. He proposed an ‘‘easy’’ algorithm to the two machine flow shop problem with makespan as the criterion. Since then, several researchers have focused on solving *m* machine (m >2) flow shop problems with the same criterion. However, these fall in the class of NP-hard (Garey, Johnson, & Sethi, 1976), complete enumeration techniques must be used to solve these problems. As the problem size increases, this approach is not computationally practical. For this reason, researchers have constantly focused on developing heuristics for the hard problem.

In the flow shop, a set of jobs has to be processed on *m* machines. Every machine has to process each one of the jobs and every job has the same routing through the machines. The objective is to compute the completion times of all jobs on the final machine (makespan). A flow shop instance consists in scheduling *n* jobs ($i = 1\ldots\ldots\ldots n$) on *m* machines M ($j = 1\ldots\ldots$m) . A job consists in m operations and the j^{th} operation of each job must be processed on machine j. So, one job can start on machine j if it is completed on machine *j*-1 and if machine *j* is free. Each operation has a known processing time which specifies the time required by machine *m* for processing job *j*. Each job is to be processed on all machines M1, M2… M *m* in this order (Moleshi & Mirzae, 2009).

In this context, each job has been assigned exactly *m* operations where as in real situations a job may have fewer operations, certain heuristic algorithms propose that the jobs with higher total process time should be given higher priority than the jobs with less total process time. From a review of the literature, it can be noticed that several heuristic approaches in the field of flow shop scheduling have been developed to minimize both the maximum flow time and the makespan.

2.6 Example of Flow Shop in Hard Disk Manufacturing Industry

A flow-shop is a shop design in which machines are arranged in series jobs begin processing on an initial machine, proceed through several intermediary machines, and conclude on a final machine. Scheduling in real production situations of hard-disk manufacturing system contrasts to classical scheduling where each job visits each machine only once. The flow shop as hard-disk devices (HDD) manufacturing shop floor consists several serial workstations. Each workstation is composed of only one machine for production of a total of n jobs. Each job is provided with a different sequence of operations. Some workstations can produce some jobs depending on the processing step of those jobs.

In this paper, the flow shop scheduling problems with the objective of minimizing the makespan of jobs are considered. Minimizing makespan is directly related with maximizing the system throughputs which is considered as the most important of hard-disk devices industry requirements. Moreover, most manufacturing systems have to manage with the quality of semiconductor material. The time gaps constraint in the manufacturing system must then be considered. High quality of processing with a critical controlling time is included for hard-disk manufacturing. Time gaps are controlled from the beginning step to the ending step. Hence, any job with its completion time exceeding the limited time gaps will lead to loss. Moreover, controlling processing time constraint is an important issue on the industry where requires high quality production especially in a hard-disk manufacturing system (Chamnanlor *et al*., 2012). The overall typical hard disk manufacturing process flow is shown in Figure 2.1. The hard disk manufacturing process flow consists of six machines.

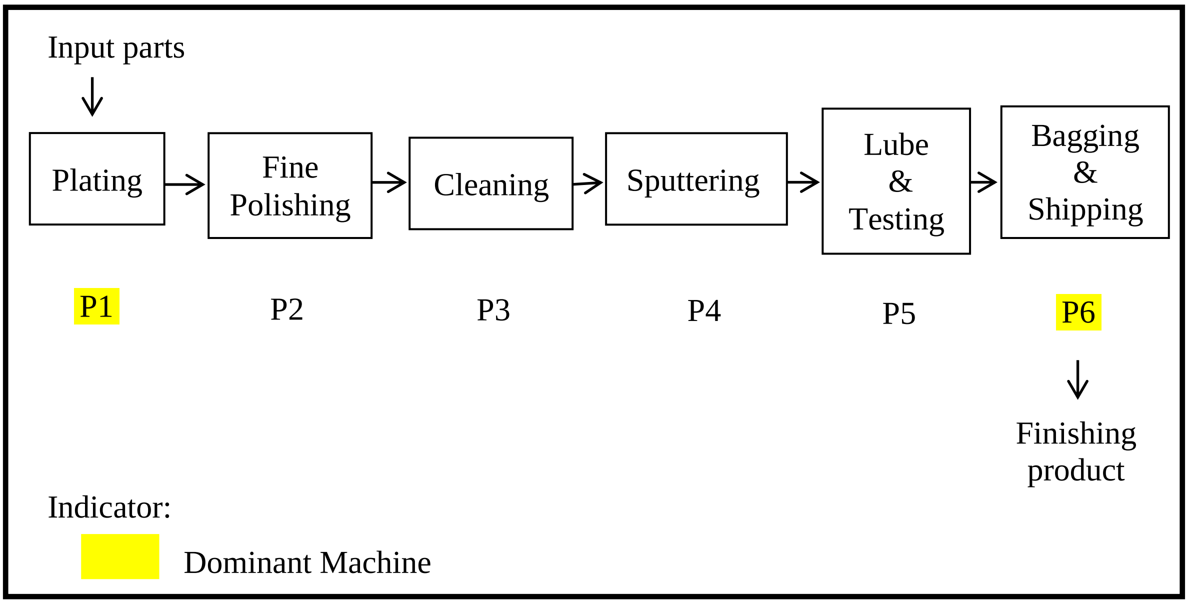

Figure 2.1: Typical Hard Disk Manufacturing Process Flow

2.7 Johnson's Rule Algorithm

This algorithm came up where Johnson in 1954 proposed processing time on machine 1 must be as long as the shortest on that the idle time on machine 2. Similarly, the unavoidable idle time on machine 1 must be as long as the shortest processing time on machine 2. This lead to a better bound on makespan. Johnson's algorithm is the best sequence to give an optimal solution for the 2 machine case (Su & Lin, 2006).

The steps in Johnson's algorithm are as follows: First identify the shortest processing time among all jobs on both machines. Get the shortest time on the first machine to start the flow and get the shortest time on the second machine and schedule it as the end of the flow. This simple algorithm can be extended to optimize *n*-job, 3 machine flow shop problem under certain conditions. Johnson's extended this algorithm to causes where the second machine was dominated by either the first or third. Johnson's rule cannot be extending to three machines unless that the second machine is not "bottleneck" (Su & Lin, 2006).

Nevertheless, several heuristic method were modeled on the application of Johnson's rule algorithm such as the Campbell, Dudek and Smith heuristic known as the CDS heuristic uses a multiple application of the Johnson's rule algorithm (Campbell *et al*., 1970). In order to apply the Johnson's rule in solving the multistage $F_m/prmu/C_{max}$ problem, the CDS heuristic breaks the m-stage problem into two stage problems. This method creates a total of m – 1 new two stage sub – problems and then the Johnson's rule is applied to each of them in order to search for the minimum makespan. Since the CDS heuristic uses the sorting methodology based on the Johnson's rule index value therefore it is classified under the category of index development utilizing the $F_2//C_{max}$ analogy (Framinan *et al*., 2004).

In describing the CDS heuristic, Lee (2000) uses the following explanations: Let k be the counter for the m – 1 sub – problems. The new computed processing time for the first stage is denoted by a (i, k) where i denotes the job and k denotes the k^{th} sub – problem. The second stage processing time is denoted by b (i, k).

The new processing time for each job belongs to both stages are computed using these equations:

$$a\ (i,k) = \sum_{i=1}^{k} P\ (i,j), (k = 1,2,..m-1) \tag{2.1}$$

$$b\ (i,k) = \sum_{i=m-k+1}^{m} P\ (i,j),\ (k = 1,2,m-1) \tag{2.2}$$

For each of the sub – problem, application of Johnson's rule on the newly computed processing time of both a (i, k) and b (i, k) provides a set of scheduling sequence. All the generated sequences are then applied to the m – stage problem for makespan computation. The sequence that generates the smallest makespan is considered as the best alternative. This heuristic together with a few other related researches were discussed by Framinan *et al*. (2004) in their review and classification of heuristic for permutation flow shop scheduling with makespan objective that emphasizing the Johnson's rule algorithm. These approaches are summarized in Table 2.1.

Table 2.1: Permutation flow shop heuristic using index development and $F_2// C_{max}$ analogy (Framinan *et al*., 2004)

References	First Stage Processing Time (*P*1'*j*)	Second Stage Processing Time (*P*2'*j*)
Campbell *et al.* (1970)	$\sum_{i=1}^{k} P\,(i,j)\,,(k=1,2,\dots m-1)$	$\sum_{i=m-k+1}^{m} P\,(i,j),(k=1,2,\dots m-1)$
Dannenbring (1977)	$\sum_{i=1}^{m} (m-i+1)P\,(i,j)$	$\sum_{i=1}^{m} iP\,(i,j)$
Rock and Schmidt (1983)	$\sum_{i=1}^{[\frac{m}{2}]} P\,(i,j)$	$\sum_{i=\left\lceil\frac{m}{2}\right\rceil+1}^{m} P\,(i,j)$
Selim and Al-Turki (1987)	$\sum_{i=1}^{m} \lambda_i P\,(i,j)$	$\sum_{i=1}^{m} \mu_i P\,(i,j)$
Lai (1996)	$\sum_{i=1}^{[\frac{m}{2}]} P\,(i,j)$	$\sum_{i=m-\left\lceil\frac{m}{2}\right\rceil+1}^{m} P\,(i,j)$

i denotes machines or stages (*i* = 1,2,..*m*), *j* denotes jobs

2.8 Bottleneck Approach

The bottleneck phenomena occur frequently in many manufacturing systems. Most processes involve multiple operations, and often their capacities are not identical. A bottleneck is an operation that has lowest capacity of any operation in the process and thus limits the system's output (Krajewski and Ritzman, 2005). The bottleneck or dominant set may be thought of as a set of precedence constraint on jobs and useful develop an algorithm for an NP-hard problem since a large number of sequences can be disregarded (Pinedo, 2002). Bottleneck management is a very important task on the shop floor and is really effective in production scheduling.

Scheduling approaches for flow shop and job shop problems with bottleneck stages usually include three steps which are (Adler *et al*., 1993; Pinedo, 2002):

i. Identify bottleneck stage
ii. Schedule bottleneck stage
iii. Schedule non-bottleneck stages

However, Conway (1997) stated that it is often important to schedule subordinate resources carefully to ensure timely support of constraint resources. Using bottleneck-based heuristics to solve the flow shop problems has attracted many researchers. Adler *et al.* (1993) considered a practical scheduling problem for plants that produce multiple paper bags. The machine environment can be regarded as a flexible flow shop, and the machines at a stage may not all be identical. They developed an ad hoc bottleneck- based heuristic to solve the specific problem. Chen and Lee (1998) suggested a bottleneck-based group scheduling procedure to solve flow line cell scheduling problems.

The procedure was based on the bottleneck machine and attempted to fully utilize the bottleneck machine and minimize makespan. Lee *et al*. (2004) developed a bottleneck-based heuristic to solve a multistage hybrid flow shop problem with identical parallel machines at each stage and with minimum total tardiness as the objective. The heuristic first focuses on the bottleneck stage, constructs the schedule of the bottleneck stage, and constructs schedules for other stages based on the schedule of the bottleneck stage. The heuristic uses the sum of processing times of a job at the upstream stages to be the arrival time of the job at the bottleneck stage. If the procedure results in an infeasible schedule, then the arrival times of the jobs at the bottleneck stages will be iteratively modified until a feasible schedule is obtained (Chen & Chen, 2009). They compared the performance of eight well-known dispatching rules and the bottleneck-based heuristic. The computational results showed that the heuristic dominated all the dispatching rules.

2.9 Nawaz, Enscore, and Ham (NEH) Heuristic

The well-known NEH heuristic from Nawaz, Enscore and Ham proposed in 1983 has been recognized as a heuristic with better performance than other heuristics based on makespan minimization. This performance lead is maintained even today when compared against contemporary and more complex heuristics.

Several studies place NEH as the best performing method. Direct evaluations against older methods are given in Taillard (1990) where NEH is shown to provide better results than other highly cited heuristics such as the CDS method of Campbell *et al*. (1970). More importantly, in Ruiz and Maroto (2005), NEH was tested against 25 other heuristics, including the more modern and complex algorithms of Koulamas (1998) and Suliman (2000).The results supported by careful statistical analyses, show that NEH is vastly superior to all tested methods and at the same time are much faster. As a result, NEH is used today as a seed sequence in many, if not all, effective metaheuristics proposed for the permutation flow shop scheduling problem.

In addition, many heuristics have been developed for pure flow shop problems. CDS (Campbell *et al*., 1970), DAN (Dannenbring, 1977), and NEH (Nawaz *et al.*, 1983) are well-known ones. These heuristics have recently been applied to solve FFL problems. Note that NEH is a solution-construction type heuristic.

The idea of the NEH heuristic is very simple. First, NEH finds the priority order by sorting the jobs according to their non-increasing total processing times. Later, the first unscheduled job in this order is inserted in the best position among all possible positions of the current subsequence of already scheduled jobs. The NEH insertion phase is rather straightforward with the exception of an undefined tie-breaking method.

2.10 Sequencing Rules

Sequencing is prioritizing jobs assigned to a resource. The form of the optimal sequencing rule depends on several factors, including the pattern of arrivals of jobs, the configuration of the job shop or flow shop, constraints, and the optimization objectives (Stevenson, 2012).

There were four sequencing rules commonly used in practice as:

i. First-come, first served (FCFS) – Job is processed in sequence in which they entered the shop.
ii. Shortest processing time (SPT) – Job is sequenced in increasing order of their processing times. The job with the shortest processing time is first, the job with the next shortest processing time is second and so on.
iii. Earliest due date (EDD) – Job is sequenced in increasing order of their due dates. The job with the earliest due date is first, the job with the next earliest due date is second, and so on.
iv. Critical ratio (CR) – Critical ratio scheduling requires forming the ratio of the processing time of the job, divided by remaining time until the due date, and scheduling the job with the largest ratio next.

2.11 Previous Research Related to the Study

2.11.1 Research on the Re-Entrant Flow Shop Scheduling Problem

Bareduan and Hasan (2010) had investigated the potential development of a bottleneck-based makespan algorithms and heuristic to minimize the makespan of an Internet based collaborative design and manufacturing process that resembles a four machine permutation re-entrant flow shop. It was shown that using bottleneck-based analysis, effective makespan algorithms and a constructive heuristic known as the bottleneck adjacent matching 2 (BAM2) heuristic can be developed to solve for near-optimal scheduling sequence. In this paper, the author divided the dominance level groups into three levels which weak dominance, medium dominance and strong dominance. The simulation results indicated that at strong dominance level, the BAM2 heuristic is capable to produce near optimal results and this heuristic generates results which are very much compatible to the NEH. It was concluded that at a strong machine dominance level and specific 10 to 20 job numbers problem, this heuristic shows better makespan performance compared to the NEH.

Bareduan and Hasan (2008) developed and evaluated a bottleneck-based scheduling heuristic of a four machine permutation re-entrant flow shop with the process routing of M1, M2, M3, M4, M3 and M4. It was shown that the first process at M1 has a high tendency of exhibiting dominant characteristic which is at strong P1 dominance level. The bottleneck adjacent matching 4 (BAM4) heuristic is developed based on the bottleneck correction factor algorithm introduced to the makespan computation using bottleneck approach. BAM4 heuristic is capable to produce near optimal results for all the problem sizes studied. A total of 3000 simulations were conducted. Base on the results, it was concluded that within strong P1 dominance level and at the job numbers (n = 10 and 20), this heuristic generates results which are very much compatible to the NEH. However, for smaller job numbers (n = 6), NEH is superior. The bottleneck approach presented in this study also can be utilized to develop specific heuristics for other re-entrant and ordinary flow shop operation systems that shows significant bottleneck characteristics.

Bareduan and Hasan (2012) have presented the methodology to develop an effective makespan computation algorithms and heuristic using bottleneck analysis. The case under study was a specific re-entrant flow shop with the process routing of M1, M2, M3, M4, M3, M4 in which M1 and M4 have high tendency of exhibiting bottleneck characteristics. It was shown that using this bottleneck approach, two alternative bottleneck-based makespan algorithms were successfully developed for the identified bottleneck and to search for the near optimal scheduling solutions. In arranging the schedules, BAM heuristic utilized and index values generated from the bottleneck correction factor introduced in the makespan algorithms which to ensure the accuracy of the makespan computation. Therefore, the author considered two type of size problems which refer to small size and medium and large size. For small size problems, the heuristic results were compared with the optimum makespan generated from complete enumeration. Otherwise, medium and large size problems were measured by comparing its makespan with the solutions generated by the NEH and lower bound.

Therefore, the simulation results showed that at weak and strong which is first processing time ($P1$) dominance level, the BAM heuristic was capable to produce near optimal results for all the problem sizes studied. Within the weak and strong $P1$ dominance levels and medium to large size problems (n =10 and 20), BAM shows very close performance against the LB and better makespan performance compared to the NEH. However, within the medium $P1$ dominance level and for small size problems (n=6), BAM did not produce better results than NEH. This paper also described the bottleneck scheduling performance (BSP) indexes procedure which capable to ascertain that some specific generated job arrangements was the optimum schedule.

2.11.2 Research on the Flow Shop Scheduling Problem

Moslehi *et al.* (2009) have considered the problem of two-machine flow shop scheduling in which the objective function is to minimize the sum of maximum earliness and tardiness ($n/2/P/ET_{max)}$. Since this problem tries to minimize earliness and tardiness, the results can be useful for different production systems such as just in time (JIT). This objective function has already been considered for n jobs and *m* machines, but the proposed algorithms are not efficient to solve large scale problems. In this case, the objective function value needs to be reduced due to that it is very important in JIT production systems.

The problem of finding the optimal sequence with the objective function of the sum of maximum earliness and tardiness (ET_{max}) in a two-machine flow shop is represented by $n/2/P/ET_{max}$. In order to design the problems, two significant points were considered. The first point was the type and properties of the objective function and the second was the type of the problem (two-machine flow shop). In designing the experiments, two separate groups were recognized with regard to the nature of the problems generated. The first group consisted of 220 problems with sizes from 4 to 1000 jobs while the second group consist of 160 problems with sizes ranging from 4 to 50 jobs were generated. The results showed that the proposed algorithm in the first group had a very high efficiency, in which the problems up to 1000 jobs could be solved in a reasonable time. The proposed branch-and-bound algorithm had an acceptable efficiency in this group, as it was able to solve problems with up to 20 jobs in a reasonable time. However, computational results showed that the efficiency of the proposed algorithm reduced in the second group as compared to the first group. More than 82% of the problems are shown to reach optimal solutions.

Ladhari and Haouari (2000) introduced a lemma for changing the objective function of maximum lateness (L_{max}) to the objective function of optimal makespan (C_{max}), in which more simple algorithms were defined for the former objective function. In this paper, an effective branch and bound algorithm for the permutation $F2|r_j, l_j, q_j|C_{max}$ problem was presented. By developing the branch and bound algorithm, the problem of minimizing the maximum lateness in a two-machine flow shop will be solved to release dates and time lag constraints. The importance of this NP-hard problem is twofold, it arises as a strong relaxation of the classical permutation flow shop problem, and it generalizes several well studied two-machine flow shop problems. The computational experiments result show that randomly generated instances with up to 1000 jobs can be solved optimally. Hence, the challenge is to embed the $F2|r_j, l_j, q_j|C_{max}$ solution as a strong lower bound for the optimization of the permutation flow shop problem.

An interesting issue that is worthy of future research is to consider a model in which different sequences of jobs are allowed on the two machines, rather than consider the permutation problem. It appears that several ideas developed in this paper can be extended to the generalized problem, but this will require for further research examination.

Schmidt (2000) has reviewed results on scheduling problems with limited machine availability for processing. The number of results shows that scheduling with availability constraints attracts more and more researchers, as the importance of the applications are recognized. For very few cases there exist optimal on-line algorithms. More cases can be solved by nearly on- line algorithms but the majority of cases can only be solved to optimality by on-line algorithms. For off-line settings either classical algorithm could be generalized to solve the problem in polynomial time, or it could be shown that the problem becomes NP-complete due to the availability constraints. The author summarized the results in in different for a given problem type between performance criteria involving NP-completeness and those for which a polynomial algorithm exists. These summaries were covered only preemptive scheduling problems because it is easy to show that if preemption is not allowed optimality cannot be reached by this type of algorithms. If availability constraints come from unexpected breakdowns, fully on-line algorithms are needed. But many results of optimality concern at best nearly on-line algorithms which are in case of preemptive scheduling.

Blazewicz *et al.* (2001) have studied the heuristic algorithms to solve the two machine flow shop problem with limited machine availability. The objective is to minimize the makespan and the problem is known to be NP-hard for two machines. The constructive and local search was analyzed based on heuristic algorithms for the two-machine case. The algorithms are tested on easy and difficult test problems with up to 100 jobs and 10 intervals of non-availability. This study was evaluated using easy and difficult problem instances. It turned out that at least 5870 out of 6000 easy instances and 41 out of 100 difficult instances could be solved to optimality using a combination of two constructive methods which are Johnson's rule and a new look-ahead heuristic based on local optimization as well as a simulated annealing algorithm. For 127 easy instances the optimality could not be proved. The time limit to achieve this result was roughly 60s for each instance. The worst relative performance for easy and difficult instances was 2.6% and 44.4% above the optimum, respectively. At least 5812 out of 6000 easy instances and 13 out of 100 difficult instances could be solved combining the two constructive methods only with an average computation time of 0.33 and 3.96 s per instance, respectively. The heuristics proved to be robust with respect to relative shop and machine availability.

The results suggest that the presented heuristic solution approach is a good alternative for solving large two-machine flow shop scheduling problems with limited machine availability. A next step in the research could be to design heuristic algorithms for the *m*-machine flow shop scheduling problem with availability constraints.

2.11.3 Research on the Permutation Flow Shop Scheduling Problem

Schaller and Valente (2013) were performing the several procedures for developing non-delay schedules for a permutation flow shop with family setups when the objective is to minimize total earliness and tardiness. These procedures consist of heuristics that were found to be effective for minimizing total tardiness in flow shops without family setups, modified to consider family setups and the total earliness and tardiness objective. The author considered three distributions of family setups and six sets of distributions that determine the tightness and range of due dates. The solutions generated were compared against optimal solutions for small-sized problems and the solutions found by one of the procedures for large-sized problems.

A test with varying conditions was also conducted to see the effect of reducing setup times on the total earliness and tardiness obtained in scheduling. It was found that if setup times are reduced, then total earliness and tardiness can be significantly reduced. The reduction in total earliness and tardiness is achieved not only from additional effective capacity obtained by reducing setup times, but also utilizing the resources more effectively by scheduling smaller batches of jobs belonging to each family so that production of individual jobs is better matched to their due dates. The test also showed that the tightness of the due dates affected the reduction in total earliness and tardiness when setup times are reduced. When there was excess capacity and due dates were not tight reducing setup times did not impact earliness and tardiness as much as when capacity was tighter. This could be due in large part to the restriction that only non-delay schedules were considered.

2.11.4 Research on the Flexible Flow Line Scheduling Problem

Chen and Chen (2009) developed bottleneck-based heuristics to solve the multiple-stage flexible flow line problem with unrelated parallel machines and with a bottleneck stage in the system. The objective is to minimize the total tardiness. The author proposed two bottleneck-based heuristics with three machine selection rules to solve the problem. The heuristics first develop an indicator to identify a bottleneck stage in the flow line, and then separate the flow line into three stages which are the upstream stages, the bottleneck stage, and the downstream stages. The upstream stages are the stages ahead of the bottleneck stage and the downstream stages are the stages behind the bottleneck stage. This new approach is developed to find the arrival times of the jobs at the bottleneck stage to overcome the difficulty of determining feasible arrival times of the jobs at the bottleneck stage. Using the new approach, the bottleneck-based heuristics develop two decision rules to iteratively schedule the jobs at the bottleneck stage, the upstream stages, and the downstream stages.

In order to evaluate the performance of the bottleneck-based heuristics, seven commonly used dispatching rules and a basic tabu search algorithm are investigated for comparison purposes. Seven experimental factors which are number of jobs, number of stages, position of the bottleneck stage in the flow line, workload difference between bottleneck and non-bottleneck stages, variation of job processing time, tardiness factor and due range used to design 128 production scenarios and ten test problems are generated for each scenario.

The results show that when unrelated parallel machines are considered in the stages, the third machine selection rule, ECALLM which is to select the machine with the earliest completion time when the job is assigned to all the machines in the stage significantly dominates the other two machine selection rules. Also, ECALLM significantly affect the performance of the dispatching rules, the bottleneck-based heuristics and basic tabu search (BTS). Hence, the bottleneck-based heuristics significantly outperform several well-known dispatching rules for the test problems. Although the effective performance of the bottleneck-based heuristics is inferior to the basic tabu search algorithm, the bottleneck-based heuristics are much more efficient than the tabu search algorithm. Furthermore, a test of the effect of the experimental factors on the dispatching rules, the bottleneck-based heuristics, and the basic tabu search algorithm is performed, and some interesting insights are discovered.

Chen and Chen (2009) has concluded that machine selection rule is a key factor affecting the performance of the heuristics for the flexible flow line (FFL) problem where unrelated parallel machines exist in all the stages, with the objective of minimizing the makespan. A bottleneck-based heuristic for the candidate FFL problem (BBFFL) was proposed with three selection machine rules. The first machine selection rule, EAAM is to select the machine with the earliest available time among the available machines. The second selection rule, ECAM is to select the machine with the earliest completion time when the job is assigned to the available machines. The third selection rule, ECALLM is to select the machine with the earliest completion time when the job is assigned to all the machines at the stage, including available and unavailable machines. Thus, it will be used to determine the schedule of the jobs at each stage. Therefore, in BBFFL a variant of Johnson's rule is used to develop a bottleneck-based initial sequence generator (BBISG). Then, a bottleneck-based multiple insertion procedure (BBMIP) is applied to the initial sequence to control the order by which jobs enter the bottleneck stage to be the same as that at the first stage.

In order to evaluate the performance of bottleneck-based heuristics for flexible flow line (BBFFL), a series of computational experiments have been conducted. Five experimental factors which are the number of jobs, the number of stages, the variation of processing times, the position of bottleneck stages in the flow line and the workload difference between bottleneck and non-bottleneck stages were used to design 243 different production scenarios. These test problems were used to compare the performance of BBFFL with four well-known heuristics which are CDS, DAN, NEH and CDSD.

The results were shown that the third machine selection rule, ECALLM significantly dominates the other two machine selection rules, EAAM and ECAM. In addition, the proposed heuristic, BBFFL significantly outperforms the four well-known heuristics, CDS, CDSD, DAN and NEH. The heuristic, BBFFL/ECALLM produced best solutions for most of the test problems which 1639 out of 2430 and it deviates from the best solutions, on average by only 0.7%. Furthermore, the experimental design has shown that the performance of BBFFL is fairly stable. Therefore, the proposed bottleneck-based heuristic, BBFFL/ECALLM can be further applied to FFL and it can also be applied to solve other scheduling problems such as job shop problems with bottleneck stages.

2.12 Summary

This chapter illustrates about the previous research related to the computation of minimizing makespan and total tardiness, bottleneck-based approach, algorithm, scheduling problem and the variations of flow shop problem. The enhancement of manufacturing industry leads the researchers to focus and explore in this type of study.

Table 2.2 concludes the summary of previous research with the different scope of work in the flow shop scheduling problem. According to the literature review, all of the researchers have used heuristic and algorithms to solve flow shop scheduling problem. The main objective of researchers is to minimize makespan and total tardiness. The researchers attempted to utilize various ways in order to get the optimum result for the problem. Thus, based on the information and reference obtained, this study mainly focus on a simple flow shop with bottleneck machine is being considered.

Table 2.2 Summary of literature review

No.	Year	Title and Author	Descriptions
1	(Haouari & Ladhari, 2000)	Minimizing maximum lateness in a two-machine flow shop	In this research, a branch and bound algorithm were developed to solve the problem of minimizing the maximum lateness in a two-machine flow shop.
2	(Schmidt, 2000)	Scheduling with limited machine availability	This paper reviews results related to deterministic scheduling problems where machines are not continuously available for processing.
3	(Blazewicz *et al.*, 2001)	Heuristic algorithms for the two-machine flow shop with limited machine availability	In this paper studies presented that heuristic algorithm to solve the two machine flow shop problem with limited machine availability. The objective is to minimize the makespan.
4	(Bareduan & Hasan, 2008)	Bottleneck adjacent matching 4 (BAM4) heuristic for re-entrant flow shop with dominant machine	This paper presents the second version of scheduling heuristic to minimize the makespan of a re-entrant flow shop with dominant characteristic at first process. The process scheduling resembles a four machine permutation re-entrant flow shop.
5	(Chen & Chen, 2009)	A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines	In this study, a bottleneck-based heuristic (BBFFL) was developed to solve a flexible flow line problem with a bottleneck stage, where unrelated parallel machines exist in all the stages, with the objective of minimizing the makespan.
6	(Chen & Chen, 2009)	Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines	This study was developed bottleneck-based heuristics to solve the multiple-stage flexible flow line problem with unrelated parallel machines and with a bottleneck stage in the system. The objective is to minimize the total tardiness. Two bottleneck-based heuristics with three machine selection rules are proposed to solve the problem.

REFERENCES

Adler, L., Fraiman, N., Kobacker, E., Pinedo, M. L., Plotnicoff, J. C. & Wu, T. P. (1993). BPSS: A Scheduling Support System for the Packaging Industry. *Operations Research*, vol. 41, pp. 641–648.

Baker, K.R., 1974. Introduction to Sequencing and Scheduling. New York: Wiley.

Bareduan, S. A. & Hasan, S. (2012). Methodology to Develop Heuristic for Re-Entrant Flow Shop with Two Potential Dominant Machines Using Bottleneck Approach. *International Journal of Combinatorial Optimization Problems and Informatics*, 3(3), pp. 81-93.

Bareduan, S.A. & Hasan, S. (2008). Bottleneck Adjacent Matching 4 (BAM4) Heuristic for Re-Entrant Flow Shop with Dominant Machine. *IEEE International Conference on Industrial Engineering and Engineering Management*. Singapore: IEEE. pp. 148 – 152.

Bareduan, S.A. & Hasan, S. (2010). Makespan Algorithms and Heuristic for Internet-Based Collaborative Manufacturing Process Using Bottleneck Approach. *J. Software Engineering & Applications*, 3(1), pp. 91-97.

Blazewicz, J., Breit, J., Formanowicz, P., Kubiak, W. & Schmidt, G. (2001). Heuristic algorithms for the two-machine flowshop with limited machine availability. *Omega: The International Journal of Management Science*, 29(6), pp. 599-608.

Campbell, H.G., Dudek, R.A. and Smith, M.L (1970). Heuristic Algorithm For The N Job, M Machine Sequencing Problem. *Management Science*, 16(10), pp. 630-637.

Chamnanlor, C., Sethanan, K., Fu-Chien, C. & Gen, M. (2012). Reentrant Flow-Shop Scheduling With Time Windows for Hard-Disk Manufacturing By

Hybrid Genetic Algorithms. *Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference,* pp. 896-907.

Chen, C. L. & Chen, C. L. (2009). Bottleneck-based Heuristics to Minimize Total Tardiness for the Flexible Flow Line with Unrelated Parallel Machines. *Computers and Industrial Engineering*, 56(4), pp. 1393-1401.

Chen, C. L. & Chen, C. L. (2009). A Bottleneck-based Heuristic for Minimizing Makespan in a Flexible Flow Line with Unrelated Parallel Machines. *Computers and Operations Research*, 36(11), pp. 3073-3081.

Chen, Y. C. & Lee, C. E. (1998). Bottleneck-based Group Scheduling in a Flow Line Cell. *International Journal of Industrial Engineering-Applications and Practice*, vol. 5, pp. 288–300.

Conway, R. W., Maxwell, W. L. & Miller, L. W. (1967). Theory of Scheduling. New York: Dover Publications.

Dannenbring, D.G. (1977). An Evaluation Of Flow-Shop Sequence Heuristics. *Management Sciences*, 23(11), pp.1174–1182.

Framinan, J.M., Gupta, J.N.D. and Leisten, R. (2004). A Review And Classification of Heuristics For Permutation Flow-Shop Scheduling With Makespan Objective. *Journal of The Operational Research Society*, vol. 55, pp. 1243-1255.

Garey, M.R., Johnson, D.S. & Sethi, R. (1976). Complexity Of Flowshop And Jobshop Scheduling. Mathematics of Operations Research, 1(2), pp. 117-129.

Heizer, J.H. & Render, B. (1999). Operations Management. 5th edition. Indiana University: Prentice Hall.

Johnson, S.M. (1954). Optimal Two- And Three-Stage Production Schedule With Setup Times Included. *Naval Res Logistic*, vol. 1, pp. 61–68.

Koulamas, C. (1998). A New Constructive Heuristic For The Flow Shop Scheduling. *European Journal of Operational Research*, vol. 105, pp. 66-71.

Krajewski, L.J. & Ritzman, L.P. (2005). Operations Management: Processes And Value Chains. 7th Edition. Pennsylvania State University:Prentice Hall.

Ladhari, T. & Haouari, M. (2000). Minimizing Maximum Lateness in a Two Machine Flow Shop. *Journal of the Operational Research Society*, 51(9), pp. 1100-1106.

Lee, G. C., Kim, Y. D. & Choi, S. W. (2004). Bottleneck-focused Scheduling for a Hybrid Flow Shop. *International Journal of Production Research*, vol. 42, pp. 165–181.

Lee, L. (2000). Zero Wait Flow Shop With Multiple Processors With Heuristics, Algorithm And Mathematical Concepts. University of Houston: Ph.D. Thesis.

Moslehi, G., Mirzaee, M., Vasei, M., Modarres, M., & Azaron, A. (2009). Two-Machine Flow Shop Scheduling to Minimize the Sum of Maximum Earliness and Tardiness. *International Journal of Production Economics*, 122(2), pp. 763-773.

Pinedo, M. L. (2008). Scheduling: Theory, Algorithms and Systems. 3rd edition. New York: Springer Science + Business Media.

Ruiz, R. and Maroto, C. (2005). A Comprehensive Review And Evaluation Of Permutation Flowshop Heuristics. *European Journal of Operational Research*, vol. 165, pp. 479-494.

Schmidt, G. (2000). Scheduling with Limited Machine Availability. *European Journal of Operational Research*, vol. 121, pp. 1-15.

Schaller, J. & Valente, J. M. S. (2013). An Evaluation of Heuristics for Scheduling a Non-Delay Permutation Flow Shop with Family Setups to Minimize Total Earliness and Tardiness. *Journal of the Operational Research Society*, 64(6), pp. 805-816.

Steveson, W. J. (2012). Operations Management. 11th edition. New York: McGraw-Hill Companies, Inc.

Suliman, S.M.A. (2000). A Two Phase Heuristic Approach To The Permutation Flow-Shop Scheduling Problem. *International Journal of Production Economics*, vol.64, pp. 143-152.

Su, H. & Lin, T. (2006). Three Machine Flow Shop With Two Operations Per Job To Minimize Makespan. *Computers & Industrial Engineering*, vol. 50, pp.286 – 295.

Szwarc, W. (1973). Optimal Elimination Methods in the m x n Flow Shop Scheduling Problem. *Operations Research*, vol. 21, pp. 1250-1259.

Tailard, E. (1990). Some Efficient Heuristic Methods For The Flow Shop Sequencing Problem. *European Journal of Operational Research*, 47(1), pp. 65-74.