BOTTLENECK-BASED HEURISTIC FOR THREE MACHINE FLOW SHOP SCHEDULING

MOHD SALLEH BIN ABDUL RAHIM

A thesis submitted as partial fulfillment of the requirement for The Award of Masters of Mechanical Engineering.

Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia

MAY, 2011

ABSTRACT

This paper considers a 3 machine flow shop (M1M2M3) with tendency of dominant (bottleneck) machine at M1. The developed bottleneck based heuristics from previous studies are considered in this case by Hezzeril (2010) and Irwan (2010), but were only tested for dominant machine at M2 and M3 respectively. The heuristics have successfully produced 67.24% of optimum solution at the middle process or M2 and 90.80% at the last process or M3 for 6 jobs problem. While for 10 jobs problem, the heuristics can produce 14.64% at M2 and 90.98% at M3. As an extension of this study, the bottleneck based heuristic scope is enlarged by developing a new heuristic for dominant machine at M1 and combining it with the previously developed heuristics for dominant machine at M2 and M3. The main objective is to develop scheduling heuristic to evaluate the performance at M1 based on bottleneck analysis for M1M2M3 flow shop and to combine with the developed heuristics from previous studies. The computer program involved were Microsoft Excel and Visual Basic for Applications (VBA) and the test of performance were conducted at 6 and 10 jobs problem. A simulated random data within specific limitation being assigned at each job's processing time produces new recommended job arrangements. The generated makespan was compared with optimum makespan from complete enumeration and lower bound (LB) analysis. Total sets of 1000 simulated data at 6 and 10 jobs were allocated into 3 dominance level of P1DL; weak, medium, and strong. Optimal solutions were obtained based on the total results data that produce the ratio of 1. Based on the results, 62.40% of the solution generated is optimum result for 6 jobs while 56.33% of the solution generated equals to lower bound for 10 jobs. The heuristic performed moderately and decreased slightly when number of jobs increased, showing that BMM1 heuristic is more suitable for lesser number of jobs.

ABSTRAK

Kajian ini mempertimbangkan 3 mesin flow shop (M1M2M3) dengan kecenderungan dominan (bottleneck) di mesin M1. Heuristik berpandukan bottleneck daripada Hezzeril (2010) dan Irwan (2010) dipertimbangkan dalam kes ini tetapi hanyalah diuji untuk mesin yang dominan di M2 dan M3 sahaja. Heuristik tersebut berjaya menghasilkan penyelesaian optimum sebanyak 67.24% di proses pertengahan atau M2 dan 90.80% di proses pengakhiran atau M3 untuk 6 masalah kerja. Bagi 10 masalah kerja, heuristik tersebut dapat menghasilkan 14.64% di M2 dan 90.98% di M3. Sebagai lanjutan daripada kajian ini, skop heuristik berpandukan bottleneck ini dibesarkan lagi dengan menghasilkan heuristik baru untuk mesin yang dominan di M1 dan menggabungkan ia dengan heuristik sebelumnya yang dominan di M2 dan M3. Objektif utama ialah untuk menghasilkan penjadualan heuristik bagi menilai prestasi di M1 berpandukan analisis bottleneck untuk flow shop M1M2M3 dan untuk menggabungkan dengan heuristik yang telah dihasilkan daripada kajian sebelumnya. Program komputer yang terlibat adalah Microsoft Excel dan Visual Basic for Applications (VBA) dan cubaan prestasi dijalankan pada 6 dan 10 masalah kerja. Simulasi data rawak dalam lingkungan yang tertentu pada proses masa bagi setiap kerja menghasilkan susunan kerja yang baru. Makespan yang terhasil telah dibandingkan dengan makespan optimum daripada enumeration lengkap dan analisis lower bound (LB). Sejumlah 1000 data simulasi pada 6 dan 10 kerja telah ditempatkan ke dalam 3 peringkat dominan bagi P1DL iaitu lemah, sederhana dan kuat. Penyelesaian optimum yang didapati berpandukan kepada jumlah keputusan data yang menghasilkan nisbah 1. Berpandukan kepada keputusan, 62.40% daripada penyelesaian yang dihasilkan adalah optimum bagi 6 kerja manakala 56.33% daripada penyelesaian yang bersamaan dengan lower bound bagi 10 kerja. Heuristik ini menunjukkan prestasi sederhana dan sedikit menurun apabila bilangan kerja meningkat, menunjukkan heuristik BMM1 lebih sesuai digunakan untuk bilangan kerja yang sedikit.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

GT	-	Group Technology
NP	-	Non-Deterministically Polynomial
M1	-	Machine number 1
M2	-	Machine number 2
M3	-	Machine number 3
MRP	-	Material Requirement Planning
MRP II	-	Manufacturing Resources Planning
WIP	-	Work-in-Progress
J1	-	Job 1
J2	-	Job 2
J3	-	Job 3
NEH	-	Nawaz, Enscore and Ham
CDS	-	Campbell, Dudek and Smith
P1	-	Process 1
P2	-	Process 2
P3	-	Process 3
FCFS	-	First-Come, First-Served
SPT	-	Shortest Processing Time
EDD	-	Earliest Due Date
CR	-	Critical Ratio
BBFFL	-	Bottleneck-based Heuristic for Flexible Flow Line
TCG	-	Transform Classical Flow Shop to Generalized Shop
DLMP	-	Dominance Level of Middle Process
BB	-	Bottleneck Based
BMM1	-	Bottleneck Machine M1
LB	-	Lower Bound
DL	-	Dominance Level

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

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. Scheduling in manufacturing systems is classically associated with scheduling a set of jobs on a set of machines in order to maximize the profit. Manufacturing system is classified as job shop, flow shop and open shop. Technological constraints demand that each job should be processed through machines in a particular order and gives a significant special case named as flow shop.

Flow shop scheduling is one of the most important problems in the area of production management. It can be briefly described as follows: There are a set of *m* machines (processors) 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. There are no precedence constraints among operations of different jobs. Operations cannot be interrupted and each machine can process only one operation at a time.

In scheduling, the “bottleneck” in the processing is the main problems concerned by 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 tool. 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 or “rules of thumb” for obtaining feasible but not necessarily optimal solution to problems. Heuristic is developed by considering the work centre that may be a single machine; 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.

1.2 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. That is why heuristic method is developed to solve these problems. Independent research (Jeffries et al. 1991) has indeed confirmed that heuristic evaluation is a very efficient usability engineering method.

From previous studies, two bottleneck-based heuristics have been developed for three machine flow shop scheduling with the tendency of dominant machine at the middle and last process. From the results based at strong dominance level, the heuristics can produce 67.24% at the middle process and 90.8% at the last process for six jobs problem. While for ten jobs problem, the heuristics can produce 14.64% at the middle

process and 90.98% at the last process. This study is directed towards developing a new heuristic for solving the three machine flow shop scheduling problem for six jobs and ten jobs at the first process and combining it with the previously developed heuristics. It also involves the development of a new algorithm for dominance level computation and a new computer program to evaluate the effectiveness of the overall heuristic.

1.3 Objectives of study

The main objective is to develop a new scheduling heuristic and to evaluate its performance for M1M2M3 flow shop by combining the developed heuristics from previous studies.

1.4 Scope of study

i. The study will focus on M1M2M3 flow shop.
ii. The study will develop a new scheduling heuristic for M1M2M3 flow shop scheduling problem by combining the heuristics from previous studies.
iii. The study involves the development of a computer program that can be used to evaluate the performance of the new heuristic.
iv. The computer program will be developed using Microsoft Excel and Visual Basic for Application.
v. The performance evaluation of the heuristic will be done by using makespan computation of six jobs and ten jobs problem.
vi. The study will compare the performance of the new heuristic against the result of previous studies.

1.5 Significance of 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. Scheduling system is a very important criterion in manufacturing industry. 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.

Previous research has found several ways in developing scheduling heuristic using bottleneck approach and Macro-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.

1.6 Expected result

At the end of this study, it is hoped that the study will produce a near optimal solution that will minimize the makespan in flow shop scheduling. This study will develop a constructive bottleneck-based heuristic that can minimize the makespan of a three machine flow shop at the first process using absolute bottleneck analysis. The findings from this study will compliment the previous studies of two bottleneck-based heuristics developed for three machine flow shop scheduling with the tendency of dominant machine at middle and last process. By developing the heuristic and combining it with

the previously developed heuristics, this study will develop a new heuristic for solving the three machine scheduling problem.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Scheduling is a decision-making process that is used on a regular basis in many manufacturing and services industries. It deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. The resources and tasks in an organization can take many different forms. The resources may be machines in a workshop, runways at an airport, crews at a construction site, processing units in a computing environment, and so on. The tasks may be operations in a production process, take-offs and landings at an airport, stages in a construction project, executions of computer programs, and so on. Each task may have a certain priority level, an earliest possible starting time and a due date. 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 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 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 (Jain, 2005).

A characteristic of the machine environment 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. In general, 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 thesis, 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. A machine scheduling problem is in fact a sequencing problem where a schedule is completely specified by the sequence in which jobs are performed.

In manufacturing area, the purpose of scheduling is to minimize the production time and costs, by telling a production facility what to make, with which staff and on which equipment. The aim of the production scheduling is to maximize the efficiency of the operation and reduce costs. The production scheduler tools are great. They provide the production scheduler with powerful graphical interfaces which can be used to visually optimize real-time workloads in various stages of the production, and pattern recognition (Wikipedia, 2010).

The benefits of production scheduling include:

i) process change-over reduction
ii) inventory reduction
iii) reduced scheduling effort

iv) increased production efficiency

v) labor load leveling

vi) accurate delivery date quotes

vii) real-time information

2.2.1 Forward scheduling

Forward scheduling or in push mode operations, the provider sends work along in the absence of any call from the customer. In this mode, the providers determine when and what is the work flow. Some system uses this approach, for example, radio and television station. Many manufactured goods flow because the provider chooses to produce them, not because a customer ordered them. The schedule starts from its start time until the whole process is finished without considering its due date.

2.2.2 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 (Salleh *et al*. 2004).

2.2.3 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.

2.3 Shop scheduling models

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. Usually, it is assumed that the machines have unlimited buffer space and a job can be stored in the buffer for an unlimited amount of time. If the machines have limited buffer space, then blocking occurs when the buffer is full. In this case, the job at the upstream machine cannot be released into the buffer after completing its

processing and has to remain at the upstream machine. This occurrence prevents a job in queue at that machine from beginning its processing.

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. While the machine sequence (i.e., the processing route) of all jobs is given, the scheduling problem is to find the best job processing sequence according to a desired optimality principle (Jain, 2005).

Scheduling can be difficult for a number of reasons. One is that in reality, an operation must deal with variability in setup time, processing time, interruption and change in the set of jobs. Another major reason is that except for very small problems, there is no method for identifying the optimal schedule and it would be virtually impossible to sort through the vast number of possible alternative to obtain the best schedule.

2.4 Flow shop scheduling problem

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.

A flow shop is characterized by more or less continuous and uninterrupted flow of jobs through multiple machines in series. In such a shop, the flow of work is unidirectional since all jobs follow the same technological routing through the machines.

Although this description of flow shop resembles an assembly-line operation, there are several differences:

i) A flow shop is equipped to handle a variety of jobs as opposed to a standard product manufactured by an assembly-line.

ii) The jobs in a flow shop do not have to be processed on all machines; that is, a job may skip some operations according to its technological requirements. However, in an assembly-line, all jobs have to move from one station to another without skipping any work-station.

iii) In a flow shop, each machine is independent of other machines and can be loaded independently; whereas in assembly-line operations, each work station depends on the preceding one.

iv) Each job has its own processing time at each machine in a flow shop; however, all units of a product have a standard time at each work station in an assembly-line (Ashour, 1972). Because of these differences, Heller (1959) characterized a flow shop as a conservative assembly line.

Johnson (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; RinnooyKan, 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……*n*) on *m* machines M (j=1……*m*) . A job consists in *m* operations and the jth 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,……,Mm in this order.

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.

2.5 Bottleneck-based heuristic

Heuristics can be classified into three types: index-development, solution-construction, and solution-improvement. However, some heuristics may consist of one or more of these types. A dispatching rule is an index-development type, and a multiple- insertion heuristic, such as NEH, is a solution-construction type. Meta-heuristics, such as tabu search and simulated annealing, can be regarded as a solution-improvement type. Obviously, solution- improvement type heuristics require the longest computation time to find a solution (Chun-Lung Chen and Chuen-Lung Chen, 2009).

The bottleneck phenomena occur frequently in many manufacturing systems. Goldratt and Cox (1992) stated the idea that the bottleneck resource governs the overall system's performance. Bottleneck management is a very important task on the shop floor and is really effective in production scheduling. 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. 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.

2.6 Nawaz, Enscore, and Ham (NEH) heuristic

The well known NEH heuristic from Nawaz, Enscore and Ham proposed in 1983 has been recognized as the highest performing method for the permutation flowshop scheduling problem under the makespan minimization criterion. This performance lead is maintained even today when compared against contemporary and more complex heuristics as shown in recent studies.

Several studies place NEH as the best performing method. Direct evaluations against older methods are given in Turner and Booth (1987) and 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), Suliman (2000) and Davoud Pour (2001), as well as those of Hundal and Rajgopal (1988) and Ho and Chang (1991). 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 flowshop scheduling problem.

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.

The heuristic procedure proposed by Nawaz, Enscore Jr. & Ham is based on the assumption that a job with more total processing time on all the machines should be given higher priority than a job with less total processing time. The algorithm can be stated as follows;

Step 1: For each job v calculate

$$P_v = \sum_{k=1}^{m} p_{kv}$$

where p_{kv} = processing time of job *v* on machine *k*, and

m = number of machines.

Step 2: Arrange the jobs in descending order of P_v .

Step 3: Pick the two jobs from the first and second position of the list of Step 2, and find the best sequence for these two jobs by calculating makespan for the two possible sequences. Do not change the relative positions of these two jobs with respect to each other in the remaining steps of the algorithm. Set i = 3.

Step 4: Pick the job in the *i*th position of the list generated in Step 2 and find the best sequence by placing it at all possible *i* positions in the partial sequence found in the previous step, without changing the relative positions to each other of the already assigned jobs. The number of enumeration at this step equals *i*.

Step 5: If n = i, STOP, otherwise set i = i + 1 and go to Step 4.

2.7 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.

There were four sequencing rules commonly used in practice as:

1) First-come, first served (FCFS) – Job is processed in sequence in which they entered the shop.
2) 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.
3) 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.
4) 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.

2.8 Previous research

Table 2.1: Summary of previous research

Title	Description	Summary
A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines by Chun-Lung Chen and Chuen-Lung Chen (2009)	This study developed a bottleneck-based heuristic (BBFFL) to solve flexible flow line problem with a bottleneck stage, where unrelated parallel machines exist in all stages, with the objective of minimizing makespan. The essential idea of BBFFL is scheduling jobs at bottleneck stage may affect the performance of heuristic for scheduling jobs in all the stages.	The paper has a similar purpose which is to minimize the makespan in a flow line with bottleneck stage. Its idea of scheduling jobs at bottleneck stage affects the heuristic performance is proportional with this study. However, this paper involves only flexible flow line problem whereas the proposed study involves typical flow shop with three machines.
Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines by Chun-Lung Chen and Chuen-Lung Chen (2009)	This paper considers flexible flow line problem with unrelated parallel machines at each stage and with a bottleneck stage on the line. The objective is to minimize total tardiness. Two bottleneck-based heuristics with three machine selection rules are proposed. The heuristics develop an indicator to identify a bottleneck stage in the flow line. Seven commonly used dispatching rules are investigated for comparison purposes. Results show that bottleneck-based heuristics significantly outperform all the dispatching rules for the test problems.	The paper uses bottleneck-based heuristics to minimize total tardiness. It also uses three machine selection rules. From the paper, bottleneck-based heuristics are better than dispatching rules.

Heuristic algorithms for two machine re-entrant flow shop by Caixia Jing, Guochun Tang and Xingsan Qian (2008)	This paper focuses on a two machine re-entrant flow shop scheduling problem with the objective of minimizing makespan. The authors assume that all jobs are available at time zero and machines do not breakdown during the work. Each machine can handle only one sub-job at a time and different operations of the same job cannot be processed simultaneously. Preemption is not allowed. There is no setup time required before jobs are processed on any machine or setup times are included in the processing times.	The purpose of this paper is similar with this study which is to minimize makespan. The authors made some assumptions or limitations. This shows that before developing a schedule, assumptions have to be identified.
Improvement heuristic for the flow-shop scheduling problem: An adaptive-learning approach by Anurag Agarwal, Selcuk Colak and Enes Eryarsoy (2006)	In this paper, the authors propose an improvement-heuristic approach for the general flow-shop problem based on the idea of adaptive learning. The authors compare their results to the best-known upper-bound solutions and find that for many problems they match the best known upper bound. For one problem the authors discover a new upper bound.	The authors use a different approach by using heuristic based on adaptive learning. This is different than bottleneck-based heuristic in this study but can be used to solve flow shop scheduling problem. The authors use upper-bound solutions to compare results. While in this study, makespan from complete enumeration and maximum lower bound are used to compare results.
A Fast Method for Heuristics in Large-Scale Flow Shop Scheduling by Li Xiaoping, Liu Lianchen and Wu Cheng (2006)	This paper describes a generalized flow shop model, which is an extension of the classical model, in which not all machines are available at time zero. The general completion time computing method is used to compute completion time of generalized flow shops. The transform classical flow shop to generalized shop (TCG) method is used to transform classical schedules into generalized schedules with	The authors develop a fast method for heuristics to solve a large-scale flow shop scheduling. They describe a generalized flow shop model in which not all machines are available at time zero. They compute completion time of generalized flow shops by using general completion time computing method.

	fewer jobs.	This study uses start stop data to compute completion time of flow shop scheduling.
A heuristic for minimizing the makespan in no-idle permutation flow shops by Pawel Jan Kalczynski and Jerzy Kamburowski (2005)	The paper deals with the problem of finding a job sequence that minimizes the makespan in m-machine flow shops under the no-idle condition. This condition requires that each machine must process jobs without any interruption from the start of processing the first job to the completion of processing the last job. Since the problem is NP-hard, the authors propose a constructive heuristic for solving it. The purpose of this paper is to present a new constructive heuristic for minimizing the makespan in no-idle permutation flow shops	The purpose of this paper is similar with the study which is to minimize makespan. The authors concentrate on m-machine flow shops under the no-idle condition. The same limitation is used which is each machine must process jobs without any interruption. The problem is also an NP-hard.

From previous research in Table 2.1, all of the researches have used heuristic to solve flow shop scheduling problem. Several types of heuristic were involved and some of them used bottleneck based heuristic which is the same approach with this study. Most of them are with the objective to minimize makespan. Although the methods used by them are different with this study, the purpose is the same. There are also several types of flow shop involved such as re-entrant flow shop, large scale flow shop and permutation flow shop but in this study, only a simple flow shop is being considered.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology consists of the steps of every work progress for completing a new scheduling heuristic for flow shop. This chapter delivers the explanation in details about the methods followed in conducting the research. It also acts as a guideline to develop a bottleneck-based heuristic for three machine flow shop scheduling and simulate it by using the macro programming in Microsoft Excel.

3.2 Methodology of the study

Below is the list of the methodologies that briefly explains the work progress flow chart (Figure 3.1):

i. Gather the information about bottleneck-based heuristic, flow shop scheduling and existing popular heuristics for flow shop with three machines, how to develop and simulate the scheduling system.

ii. Understand the concepts of flow shop scheduling and focus on flow shop with three machines and existing popular heuristics for flow shop with three machines.

iii. Understand bottleneck-based makespan algorithm for flow shop.

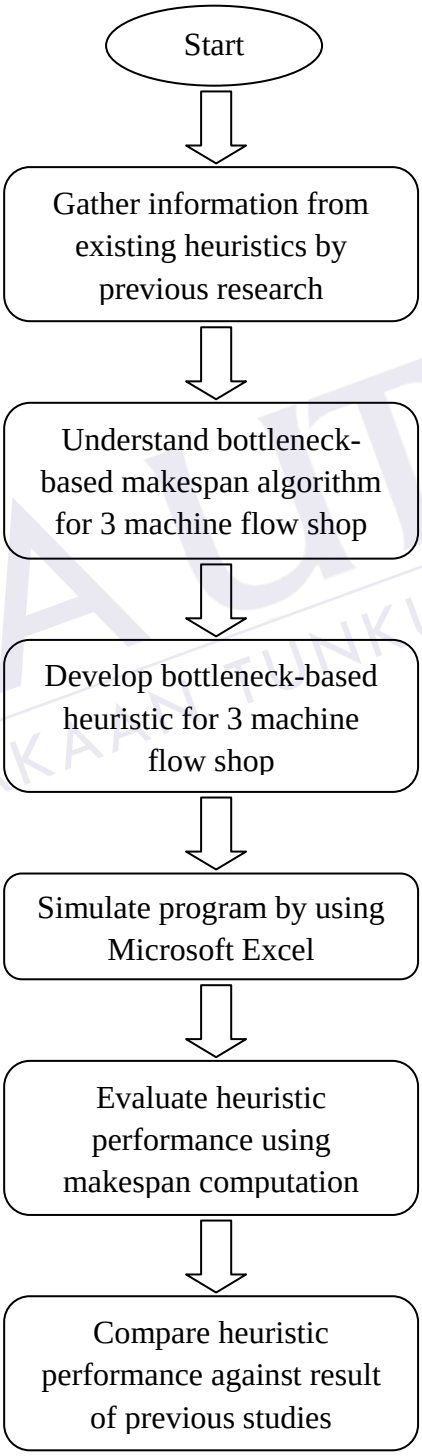
iv. Develop bottleneck-based heuristic for flow shop.

v. Convert the algorithm to Microsoft Excel coding.

vi. Develop simulation program in Microsoft Excel.

vii. Pilot runs the simulation for validation and error checking.

viii. Run simulation and analyze result.



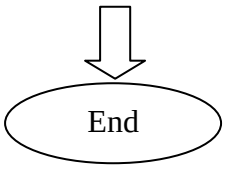


Figure 3.1: Work progress flow chart

3.3 Gather information

This project begins by gathering information about Bottleneck-based Heuristic for solving the scheduling problem. This information is taken from sources such as previous journals, thesis, internet and related books from library. At this stage, this study will define each resource that is related to the flow shop scheduling. The study also defines the input and output of the flow shop scheduling.

Below are the assumptions or limitations identified for developing the scheduling:

i. No machine can process more than one job at a time.
ii. No preemption is allowed.
iii. All setup times are included into the job processing times.
iv. There is unlimited storage between the machines.
v. All machines are continuously available (no breakdown).

3.4 Develop bottleneck-based heuristic

Before developing a new scheduling heuristic for flow shop, the researcher will consider the existing makespan algorithms and absolute bottleneck conditions. With the main objective to develop scheduling heuristic for three machine flow shop based on

bottleneck analysis, the same algorithms from previous studies will be used as a basis. The elaborations about the algorithm are shown below:

In cases where the M1 is always the dominant machine, the index can be described as below:

Let	i	=	process sequence of the job
	I	=	1, 2, 3 representing Ml, M2, M3
	j	=	number according to the scheduling sequence (j = 1, 2, 3...n)
	P (i,j)	=	processing time of the jth job at i^{th} process sequence

$$MAX\left[\{P(2,j-1)-P(1,j)\},\sum_{i=2}^{3}P(i,j)-\sum_{i=1}^{2}P(i,j)\right] \quad \text{(Equation 3.1)}$$

In cases where the M2 is always the dominant machine, the index is:

$$MAX\left[\{P(2,j)-P(3,j-1)\},\{P(2,j)-P(1,j-1)\}\right] \quad \text{(Equation 3.2)}$$

In cases where the M3 is always the dominant machine, the index would be:

$$MAX\left[\{P(2,j)-P(3,j-1)\},\sum_{i=1}^{2}P(i,j)-\sum_{i=2}^{3}P(i,j-1)\right] \quad \text{(Equation 3.3)}$$

3.5 Simulate program

A computer program will be used to simulate the data and measure the performance of the heuristic. In this research, the heuristic performance will be tested by using Microsoft Excel with built-in Microsoft Visual Basic. A computer requires programs to function, typically executing the program's instructions in a central processor and enables to develop the heuristic program. Microsoft Excel is used to develop generalized algorithm for flow shop scheduling. By using it, optimum scheduling can also be obtained for minimizing the completion time.

3.6 Evaluate heuristic performance

The performance evaluations of the heuristic using makespan computation of six and ten jobs problem will be evaluated by simulation experiment. In six jobs problem, the best schedule arrangement comes from complete enumerations which will provide the minimum makespan value. For comparison purpose, a similar test will also be conducted by using maximum lower bound technique. This technique will also be used for ten jobs problem.

A total of 1000 simulations will be conducted to six and ten jobs problem by using this new heuristic. The results from this new heuristic and lower bound technique will be compared with the optimum makespan obtained from complete enumeration except for ten jobs problem.

During each simulation, makespan from the heuristic and optimum makespan from complete enumeration are recorded. The ratio between this heuristic makespan and the optimum makespan from enumeration and from lower bound is then computed for performance measurement. The percentage of occurrence in which the makespan from this heuristic equals to the optimum makespan from complete enumeration and lower bound will also be calculated and these performances are based on the equations below:

$$\text{Makespan ratio} = \frac{\textit{Heuristic Makespan}}{\textit{Optimum Makespan}}$$

(Equation 3.4)

REFERENCES

Agarwal, A., Colak, S., & Eryarsoy, E. (2006). Improvement heuristic for the flow-shop scheduling problem: An adaptive-learning approach. *European Journal of Operational Research 169*, pp. 801–815.

Ahmad, Bareduan, S & Hasan, S. (2007). Makespan Computation for Cyber Manufacturing Centre Using Bottleneck Analysis: A Re-entrant Flow Shop Problem. *IMECS 2008, Vol. 2.*

Chen, C. L. & Chen, C.L. (2009). a) Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines. *Computers & Industrial Engineering 56*, pp. 1393–1401

Chen, C. L. & Chen, C.L. (2009). b) A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines. *Computers & Operations Research 36*, pp. 3073–3081

Chen, J. S., Pan, J. C. H., & Lin, C.M. (2008). A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem. *Expert Systems with Applications 34*, pp. 570–577.

Dhingra, A. & Chandna, P. (2010). A bi-criteria M-machine SDST flowshop scheduling using modified heuristic genetic algorithm. *International Journal of Engineering, Science and Technology, Vol. 2, No. 5*, pp. 216-225.

Gupta, J. N. D. & Stafford Jr., E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research 169*, pp. 699–711.

Jing, C., Tang, G., & Qian, X. (2008). Heuristic algorithms for two machine re-entrant flow shop. *Theoretical Computer Science 400*, pp. 137–143

Kalczynski, P. J. & Kamburowski, J. (2008). An improved NEH heuristic to minimize makespan in permutation flow shops. *Computers & Operations Research 35*, pp. 3001 – 3008

Kalczynski, P. J. & Kamburowski, J. (2005). A heuristic for minimizing the makespan in no-idle permutation flow shops. *Computers & Industrial Engineering 49*, pp. 146–154

Rad, S. F., Ruiz, R., & Boroojerdian, N. (2006). New High Performing Heuristics for Minimizing Makespan in Permutation Flowshops. *Amirkabir University of Technology, Tehran, Iran.*

Seda, M. (2007). Mathematical models of flow shop and job shop scheduling problems. *World Academy of Science, Engineering and Technology 31 2007.*

Xiaoping, L., Lianchen, L., & Cheng, W. (2006). A Fast Method for Heuristics in Large-Scale Flow Shop Scheduling. *Tsinghua Science and Technology ISSN 1007-0214 03/21*, pp. 12-18

Mansfield, R. (2008). *Mastering VBA for Microsoft Office 2007*. Indianapolis: Wiley Publishing.

Abdul Mutalib, M. A. H. (2009). *Bottleneck based heuristic for three machine flow shop scheduling with the tendency of dominant machine at the first process*. Universiti Tun Hussein Onn Malaysia: Bachelor Degree's Thesis.

Hairuddin, M. H. (2010). *Performance evaluation of M2 bottleneck based heuristic for M1M2M3 flow shop using computer program.* Universiti Tun Hussein Onn Malaysia: Bachelor Degree's Thesis.

Shariffuddin, I. (2010). *Performance evaluation of M3 bottleneck-based heuristic for M1M2M3 flow shop by using computer programming in Microsoft Excel.* Universiti Tun Hussein Onn Malaysia: Bachelor Degree's Thesis.

Universiti Tun Hussein Onn Malaysia (2009), *Thesis Writing Guide*, Centre for Graduate Studies.

Wikipedia. (2010). *Scheduling (Production Processes)*. Retrieved at September 11th 2010, from http://en.wikipedia.org