BOTTLENECK-BASED HEURISTIC FOR PERMUTATION FLOWSHOP SCHEDULING

NOOR AMIRA BINTI ISA

A thesis submitted in fulfillment 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

AUGUST 2017
DEDICATION

To my beloved parents Isa Bin Awang and Siti Hawa Binti Ismail.

Thank You Mom, Dad and Family members

For taking care of me and always standing right behind of me,

Supporting me in whatever decisions I make,

May Allah S.W.T grant all of you a Jannah.

To all my friends

Thank you for everything we shared.

For the laugh, cry and every moments,

I am the luckiest person to have all of you during my up and down.
ACKNOWLEDGEMENT

All praises to Al-Mighty God, Allah S.W.T, for His blessing in giving me a chance and protection to complete and successfully finish this thesis, for showing and guiding me through His ways, lighted my path, removed all the obstacles, eased the writing task, and for giving me a lot of kind people in this world. Thank You Allah for lending me this great of achievement. Syukur, Alhamdulillah.

My special and heartily thanks to my supervisor, Associate PM Dr. Sh Salleh Bin Sh Ahmad for his most support and consistent encouragement, and kindly read my paper and offered invaluable detailed advices on grammar. Thank you for giving me a motivation, advice and invaluable guidance in helping me throughout the course of this study.

A big thanks to my family members especially to both of my parents that always give me a full support, motivation, always encouraged and prayed for me throughout the time of this study.

I would like to convey my deepest appreciate to my special friend, Amira Syuhada Zainudin and thanks to all my fellow friends who always helped me by sharing their knowledge and also encouraged me to complete this thesis. Finally, I would like to thank those who have contributed directly or indirectly towards the success of this research project.

May the Al-Mighty God, Allah S.W.T, richly bless all of you.
ABSTRACT

The newly heuristic is developed by introducing the bottleneck-based concept that was applied at the beginning of initial solution determination. The heuristic is known as Bottleneck-Based (BNB) heuristic. The previous study found that the scheduling activity become complex when dealing with large number of machine, m > 2, it is NP-hardness. Thus, the main objective of this study are to propose and develop a new heuristic for solving permutation flowshop scheduling by considering four-machines and n-jobs (n = 6, 10, 15, 20). Three phases were applied into this study in order to boost the makespan performance of the proposed heuristic. Two phases (bottleneck identification phase and initial sequence arrangement phase) were newly introduced and combined with the existing famous Nawaz, Enscore and Ham (NEH) insertion technique. There are four potential dominant machines (M1, M2, M3, M4) clustered as bottleneck machines. A total 1000 set random processing time for each job sizes was tested using Excel simple programming with built in Visual Basic for Application (VBA). The heuristic performance was evaluated based on the average makespan ratio, average percentage error, and percentage of solution performance obtained. This study considered the NEH heuristic as the best and appropriate tool for comparison purpose since NEH heuristic is the best performing heuristic in minimizing the makespan. The heuristic evaluation result showed that the BNB heuristic is performing better than the NEH at bottleneck machines M1, M3 and M4. While, heuristic verification result showed that the bottleneck algorithms performed the best in minimizing the makespan for set of problems with bottleneck machine M4. However, the overall result showed fluctuate values over the size of jobs. The result of this study shown that the developed BNB heuristic achieved good performance in solving small sized problems however further modification is needed for medium and large sized problems.
CONTENTS

RESEARCH TITLE
DECLARATION
DEDICATION
ACKNOWLEDGEMENT
ABSTRACT
ABSTRAK
CONTENTS
LIST OF TABLES
LIST OF FIGURES
LIST OF EQUATIONS
LIST OF ABBREVIATIONS
LIST OF APPENDICES

CHAPTER 1  INTRODUCTION

1.1  Background of the study
1.2  Problem statement
1.3  Objectives of the study
1.4  Scope of the study
1.5  Significant of the study
1.6  Thesis layout
CHAPTER 2  LITERATURE REVIEW  6
  2.1  Introduction  6
  2.2  Scheduling  6
    2.2.1  Flowshop scheduling  8
    2.2.2  Job shop scheduling  9
    2.2.3  Open shop scheduling  9
  2.3  Permutation flowshop scheduling  10
  2.4  Heuristic approach  11
    2.4.1  NEH heuristic  16
    2.4.2  Bottleneck-based heuristic  17
  2.5  Solving scheduling problem with specific objectives  18
    2.5.1  Minimization of makespan and flowtime  19
    2.5.2  Earliness and tardiness  27
  2.6  Summary of the chapter  31

CHAPTER 3  METHODOLOGY  33
  3.1  Introduction  33
  3.2  Methodology of the study  33
  3.3  Data collection  36
  3.4  Introduction to simulation program  36
    3.4.1  Excel environment, formulation, tolerance  37
    3.4.2  VBA coding  37
  3.5  BNB heuristic concept  37
    3.5.1  Bottleneck machine identification phase  40
    3.5.2  Initial sequences phase  42
      3.5.2.1  Six-jobs and ten-jobs  42
      3.5.2.1  Fifteen-jobs and twenty-jobs  43
    3.5.3  Insertion phase  44
  3.6  Simulation experimental design  46
  3.7  Summary of the chapter  49

CHAPTER 4  RESULTS ANALYSIS AND DISCUSSION  50
  4.1  Introduction  50
  4.2  Bottleneck-based heuristic procedure  50
CHAPTER 5  CONCLUSION AND RECOMMENDATION FOR FUTURE WORKS

5.1  Introduction  85
5.2  Conclusion  85
5.3  Recommendation for future works  86

REFERENCES  89

APPENDIX  95
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Summary of heuristic from previous research</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Summary of makespan and flowtime from previous research</td>
<td>25</td>
</tr>
<tr>
<td>2.3</td>
<td>Summary of earliness and tardiness from previous research</td>
<td>30</td>
</tr>
<tr>
<td>2.4</td>
<td>Summary of scheduling topic</td>
<td>32</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of average processing time for six-jobs</td>
<td>41</td>
</tr>
<tr>
<td>3.2</td>
<td>Example of dominance calculation for six-jobs</td>
<td>42</td>
</tr>
<tr>
<td>3.3</td>
<td>Experimental design of generating test problems</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Example of processing times</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Example of machine dominance calculation</td>
<td>52</td>
</tr>
<tr>
<td>4.3</td>
<td>Processing time in descending order</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Initial sequences arrangement</td>
<td>53</td>
</tr>
<tr>
<td>4.5</td>
<td>Example of first two job sequencing</td>
<td>54</td>
</tr>
<tr>
<td>4.6</td>
<td>Example for third job sequencing</td>
<td>55</td>
</tr>
<tr>
<td>4.7</td>
<td>Example for fourth job sequencing</td>
<td>56</td>
</tr>
<tr>
<td>4.7</td>
<td>Example for fourth job sequencing (continued)</td>
<td>57</td>
</tr>
<tr>
<td>4.8</td>
<td>Example for fifth job sequencing</td>
<td>57</td>
</tr>
<tr>
<td>4.8</td>
<td>Example for fifth job sequencing (continued)</td>
<td>58</td>
</tr>
<tr>
<td>4.8</td>
<td>Example for fifth job sequencing (continued)</td>
<td>59</td>
</tr>
<tr>
<td>4.9</td>
<td>Example for sixth job sequencing</td>
<td>60</td>
</tr>
<tr>
<td>4.9</td>
<td>Example for sixth job sequencing (continued)</td>
<td>61</td>
</tr>
<tr>
<td>4.9</td>
<td>Example for sixth job sequencing (continued)</td>
<td>62</td>
</tr>
<tr>
<td>4.10</td>
<td>Machine bottleneck identification for six-jobs</td>
<td>63</td>
</tr>
<tr>
<td>4.11</td>
<td>Heuristics evaluation against optimum result</td>
<td>63</td>
</tr>
<tr>
<td>4.12</td>
<td>Evaluation of heuristics performance for six-jobs</td>
<td>64</td>
</tr>
<tr>
<td>4.13</td>
<td>Machine bottleneck identification for ten-jobs</td>
<td>68</td>
</tr>
<tr>
<td>4.14</td>
<td>Evaluation of heuristics performance for ten-jobs</td>
<td>68</td>
</tr>
<tr>
<td>4.15</td>
<td>Machine bottleneck identification for fifteen-jobs</td>
<td>71</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.16</td>
<td>Machine bottleneck identification for fifteen-jobs</td>
<td>71</td>
</tr>
<tr>
<td>4.17</td>
<td>Machine bottleneck identification for twenty-jobs</td>
<td>73</td>
</tr>
<tr>
<td>4.18</td>
<td>Evaluation of heuristics performance for twenty-jobs</td>
<td>74</td>
</tr>
<tr>
<td>4.19</td>
<td>Overall performance of heuristic evaluation for all problem sizes</td>
<td>76</td>
</tr>
<tr>
<td>4.20</td>
<td>BNB verification for six-jobs</td>
<td>79</td>
</tr>
<tr>
<td>4.21</td>
<td>BNB verification for ten-jobs</td>
<td>80</td>
</tr>
<tr>
<td>4.22</td>
<td>BNB verification for fifteen-jobs</td>
<td>81</td>
</tr>
<tr>
<td>4.23</td>
<td>BNB verification for twenty-jobs</td>
<td>82</td>
</tr>
<tr>
<td>4.24</td>
<td>Overall performance of heuristic verification for all job sizes</td>
<td>83</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Flowshop scheduling illustrations</td>
<td>8</td>
</tr>
<tr>
<td>3.1</td>
<td>Flow chart of heuristic development</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>720 iterations coding (six-jobs only)</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Example of 100 set of random data coding</td>
<td>38</td>
</tr>
<tr>
<td>3.4</td>
<td>Optimum answer coding (six-jobs only)</td>
<td>39</td>
</tr>
<tr>
<td>3.5</td>
<td>Example of bottleneck machine coding</td>
<td>39</td>
</tr>
<tr>
<td>3.6</td>
<td>Example of BNB and NEH makespan coding</td>
<td>40</td>
</tr>
<tr>
<td>3.7</td>
<td>Flowchart of insertion phase</td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td>Dotplot graph of heuristics performance for six-jobs</td>
<td>65</td>
</tr>
<tr>
<td>4.2</td>
<td>BNB/NEH makespan performance for six-jobs</td>
<td>67</td>
</tr>
<tr>
<td>4.3</td>
<td>BNB/NEH makespan performance for ten-jobs</td>
<td>70</td>
</tr>
<tr>
<td>4.4</td>
<td>BNB/NEH makespan performance for fifteen-jobs</td>
<td>72</td>
</tr>
<tr>
<td>4.5</td>
<td>BNB/NEH Makespan performance for twenty-jobs</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>Overall performance of BNB average makespan ratio for all job sizes</td>
<td>78</td>
</tr>
</tbody>
</table>
LIST OF EQUATIONS

<table>
<thead>
<tr>
<th>EQUATION</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Machine dominant value</td>
<td>41</td>
</tr>
<tr>
<td>3.2</td>
<td>BNB makespan Ratio</td>
<td>47</td>
</tr>
<tr>
<td>3.3</td>
<td>NEH makespan ratio</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>Percentage of BNB optimum result</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>Percentage of NEH optimum result</td>
<td>47</td>
</tr>
<tr>
<td>3.6</td>
<td>BNB/NEH makespan ratio</td>
<td>47</td>
</tr>
<tr>
<td>3.7</td>
<td>Percentage of BNB makespan equal NEH</td>
<td>48</td>
</tr>
<tr>
<td>3.8</td>
<td>Percentage of BNB makespan larger than NEH</td>
<td>48</td>
</tr>
<tr>
<td>3.9</td>
<td>Percentage of BNB makespan less than NEH</td>
<td>48</td>
</tr>
<tr>
<td>3.10</td>
<td>Percentage of accurate BNB result</td>
<td>48</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

NEH - Nawaz, Enscore and Ham
VBA - Visual Basic for Application
BNB - Bottleneck-Based
PFSP - Permutation Flowshop Scheduling Problem
NP - Non-Deterministically Polynomial
SA - Simulated Annealing
CPU - Computational Time
CDS - Campbell, Dudek and Smith
GA - Genetic Algorithm
WY - Woo and Yim
RZ - Rajendran and Ziegler
FL - Framinan and Leisten
GS - Gelders and Sambandam
MN - Miyazaki and Nishiyama
NEHKK1 - Nawaz, Enscore and Ham, Kalcynzki and Komburowski 1
NEH-D - NEH based on Deviation
FSMP - Flowshop Multiple Processors
CDS1 - Campbell, Dudek, and Smith 1
PAM - Hundal and Rajgopal modified Palmer
CDS2 - Campbell, Dudek, and Smith 2
HO - Ho Heuristic
GLB - Global Lower Bound
DC - Daniels and Chambers
CR - Chakravaryh and Rajendran
DCH - Daniels and Chambers Heuristic
SAH - Simulated Annealing Heuristic
FSDS - Flowshop With Sequence-Dependent Setup Times
PA - Percentage Advantage
API - Average Percentage Improvement
WPT-MS - Weighted Processing Time and Minimum Slack Components
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy-ET</td>
<td>Greedy-Type Procedure for Earliness or Tardiness</td>
</tr>
<tr>
<td>EDD</td>
<td>Earliest Due Dates</td>
</tr>
<tr>
<td>NEHedd</td>
<td>Nawaz, Enscore and Ham Earliest Due Dates</td>
</tr>
<tr>
<td>SPT</td>
<td>Simple Priority Rule</td>
</tr>
<tr>
<td>M1</td>
<td>Machine Number 1</td>
</tr>
<tr>
<td>M2</td>
<td>Machine Number 2</td>
</tr>
<tr>
<td>M3</td>
<td>Machine Number 3</td>
</tr>
<tr>
<td>M4</td>
<td>Machine Number 4</td>
</tr>
</tbody>
</table>
LIST OF APPENDICES

<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Processing time for six-jobs</td>
<td>96</td>
</tr>
<tr>
<td>B</td>
<td>Makespan result for BNB, NEH and optimum</td>
<td>103</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Background of the study

Scheduling is an act of planning and prioritizing activities with the time limit for the job completion in order to meet with certain requirement, constraints or to achieve the goal of objectives (Emmons and Vairaktarakis, 2012). Since the time always be the greatest constraint, therefore the scheduling becomes very important to organize the activities efficiently and optimally (Xhafa and Abaraham, 2008).

In worldwide industries, scheduling plays an important role in the production system since the resources are becoming more critical to be controlled. The resources are referred to the machine, manpower, material and many more (Chakraborty, 2009). The greatest outcome can be gained if the scheduling of the resources is successfully organized. It was a great advantage to the manufacturer in worldwide industries if the latest successful researches easy to understand and easily applied to the current manufacturing system (Mukhopadhyay, 2015). Hence, efficient scheduling system helps the manufacturer to increase the profit by cutting all the unnecessary cost associated with inefficient scheduling (Sule, 2008).
Scheduling model were categorised into many categories such as flowshop, job shop, open shop and dependent shop (Kalczynski and Komburowski, 2005). In flowshop scheduling, all the jobs are processed on multiple machines in an identical sequence with same or different processing time. This flowshop scheduling is to reduce or minimize the completion time for all the processed jobs defined as makespan. The processing order of the jobs on the machine is the same for each subsequent step of processing and this is categorised as permutation flowshop scheduling (Marichelvam, 2014).

The permutation flowshop scheduling is a well-known combinatorial optimization problem that arises in many manufacturing systems. Over the last few decades, permutation flowshop problems have widely been studied and solved as a static problem (Rahman et al., 2015). The permutation flowshop scheduling problem became famous and interesting topic among the researchers when Nawaz, Enscore and Ham (NEH) introduced the NEH heuristic around three decade ago where the heuristic was declared as the highest performing method in minimizing the makespan and achieved the answer near to optimum solution (Framinan et al., 2003). Many researchers have modified or used NEH heuristic as a basis to improve the scheduling solutions (Woo and Yim, 1998; Allahverdi and Aldowaisan, 2002; Kalcynski and Komburowski, 2005; Kalcynski and Kamburowski, 2007; Dong et al., 2007; Saleh, 2014; Isa, 2015). The current available heuristic for permutation flowshop scheduling problem was classified as constructive or improvement methods (metaheuristic). A constructive heuristic builds a sequence of jobs so that once a decision was made, it cannot be reversed. Meanwhile, improvement heuristic starts with any sequence of jobs and then attempts to decrease the value of the objective by amending the sequence. Clearly, an improvement method can be applied to the sequence obtained from a constructive heuristic (Osman and Kelly, 1996). As a continuing effort from the literature, this research is intended to improve the NEH solution and to develop a new constructive heuristic by using bottleneck-based (BNB) approach.
1.2 Problem statement

In the production scheduling system, researchers have found that the scheduling activities became more complex when dealing with more than two machines and it became a NP-hardness problem. Non-deterministic polynomial-time hard known as NP-hardness, in computational complexity theory is defined as a class of problems that are informally said “at least as hard as the hardest problems in NP” (Wegener, 2005). The researches then have focused on the development of implicit enumeration technique. However, it leads to emphasize the development of heuristic to find a near optimal solution for large size problems since implicit enumeration technique have a major constraint that there is the limitation on the problem sized that can be solved by that technique in a reasonable time (Allahverdi and Aldowaisan, 2002). It is the reason on why it is difficult to produce a new heuristic with answer near to optimum solution for a large number of machines and jobs in permutation flowshop scheduling. NEH algorithm appears to be the best heuristic in solving the makespan criterion problem, but it is not wise to just rely on NEH heuristic without further study since the improvement can still be made. Framinan et.al (2003) and Abedinnia et. al (2016) recommended further studies to be conducted to the NEH by choosing different sorting criterion beside using the ascending order of indicator values. Thus, this study is about to create a simple heuristic to boost the scheduling performance with the objective of makespan minimization since the previous finding shows that the modification is needed in order to strengthen the heuristic. The idea is by developing a new heuristic for permutation flowshop scheduling using bottleneck-based concept. The bottleneck phenomenon occurs frequently in many manufacturing systems (Chen and Chen, 2009). Identifying bottleneck resources and scheduling the jobs rationally helps in ensuring the feasibility and effectiveness of scheduling result and also help in reducing the difficulty in follow-up scheduling. Zhang and Wu (2012) shows that the local search effort for the bottleneck machines has generate higher quality of solution result and at reasonable short computational time. In this study, the bottleneck concept was applied first as initial solution step before the insertion phase where it was used as an indicator value in identifying the bottleneck machines. Besides that, this study also increases the understanding of bottleneck-based concept by developing computer program that can be used to
robustly test the performance of the algorithms using Visual Basic for Application (VBA) in Microsoft Excel. Since all the data was tabulated in Excel spreadsheet, this research is providing more visible analyses on the performance of the bottleneck-based heuristic in permutation flowshop scheduling.

1.3 Objectives of the study

The objectives for this study are:

1. To propose and develop a new heuristic using bottleneck-based (BNB) concept for permutation flowshop scheduling problem.
2. To analyse the performance of the BNB heuristic in flowshop scheduling environment.

1.4 Scope of the study

The scopes for this study are as follow:

1. Apply bottleneck-based concept for flowshop scheduling using Microsoft Excel.
2. Perform the bottleneck-based concept by using measurement of 4 machines and n-jobs \((n = 6, 10, 15, 20)\) in flowshop environment.
3. Measure the makespan performance of the case study by using Excel simple programming in Visual Basic for Application (VBA).
4. The performances of BNB heuristic were compared with the optimum solution \((n = 6)\) and NEH heuristic \((n = 6, 10, 15, 20)\).
1.5 Significant of the study

Almost all the manufacturing industries used the scheduling planning in their production system daily process. Scheduling heuristic studies is important because:

i. It helps the industries to minimize the idle time.
   When the job sequence was organized properly, the completion time can be reduced as well as the idle time.

ii. It helps the industries cutting the cost of underutilize skill labour and expensive machines.
   A simple programming used in scheduling activities that was easy to conduct can help the industries reduces the operation cost since the programming is well known and low cost.

iii. It increases the profit gained by the industries.
    The industries can reduces the operation cost through good and systematic scheduling activities, thus increasing the profit gained.

iv. It increases the related literature or academic reports in the field of production scheduling.
    Since the scheduling plays an important role in production and manufacturing industries, the latest research helps in guiding the industries and manufacturer in applying efficient scheduling system.

1.6 Thesis layout

The further design of this thesis includes the review of scheduling system and the method used to develop a heuristic by previous researchers in Chapter 2. The methodology on the techniques and procedure used in this case study was highlighted in Chapter 3. The heuristic procedure was briefly discussed, and also the result and discussion of heuristic performance and heuristic verification was presented in Chapter 4. Finally, the research conclusion and recommendation for future works was discussed in Chapter 5.
CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this section, a literature review regarding the heuristic algorithm and information related to this topic was collected and reported. All the knowledge gained from the literature contributes in better understanding of the heuristic and also the concept used in this research. In general, this review was divided into subtopics which are scheduling, permutation flowshop scheduling, heuristic approach, solving scheduling problem and lastly a summary of chapter.

2.2 Scheduling

Scheduling is a planning activity with required task and with the time constraints to be performed. It is a decision making process (Marichelvam, 2014). Scheduling is considered as important factor in many manufacturing and services industries (Sule, 2008). According to Mendes (2013), scheduling is one of the most critical issues in the planning and managing of manufacturing process where the difficulty in finding the optimal schedule depends on the shop environment, the process constraint and the performance indicator. Production scheduling problems are faced by thousands of companies worldwide that are engaged in the production of tangible goods. Efficient scheduling leads to increased efficiency, utilization and ultimately, profitability. It is
the reason that has attracted the attention of many practitioners and researchers from both fields of production management and combinatorial optimization in solving the production scheduling problems effectively and efficiently (Xhafa and Abraham, 2008). In the scheduling process, the type and the amount of each resource need to be known so that we can determine when the tasks can feasibly be accomplished. When the resources have been specified, we can effectively define the boundary of the scheduling problem (Kenneth and Dan, 2013). In production planning terminology, scheduling models are divided into some categories known as flowshop, job shop and open shop scheduling.

2.2.1 Flowshop scheduling

In flowshop scheduling, each job processed by a series of machines must have the same sequence even though the processing times may be different. Most authors added the requirement that a job never revisits any stage which can be numbered as 1, 2, ..., m, and every job visits them in numerical order. In a pure flowshop each job has m tasks and visits all stages (Emmons and Vairaktarakis, 2012). Generally, jobs may have fewer than m tasks and may skip over some stations. Jobs may skip some machines in a skip-shop (Chakraborty, 2009). There is a certain cases in flowshop identified as re-entrant flowshop which a jobs may recycle and be reprocessed at the same station in multiple times (Ahmad, 2009). But as long as all the jobs followed the same path, the systems is still considered as flowshop.

According to Mukhopadhyay (2015), in a more complex compound flowshop, each machine may be replaced by a set of parallel machines which each job can choose one from the first cluster, one from the second and so on. In a finite queue flowshop there is limited storage in front of machines other than the first. An important special case is when no storage is allowed except at the first machine. For example, in the metal processing industries, that requirement is frequently encountered where the metal is rolled while it is hot. Delays between operations result in cooling, making the rolling operation prohibitively difficult (Thomas and David, 1993). The simple variation of flowshop called as “skip shops”, re-entrant
flowshop”, “compound flowshop”, and “finite queue” flowshop. The illustration may be seen in Figure 2.1. The descriptions of flowshop variation are as follow:

1. Skip shops
   Jobs may skip some machines and treated as zero processing time.

2. Re-entrant flowshop
   Jobs may repeat the process on certain machine.

3. Compound flowshop
   A group machine may replace the machine in the series.

4. Finite queue flowshop
   There is limited storage in front of machines.

![Image of flowshop scheduling illustrations](image)

Figure 2.1: Flowshop scheduling illustrations (a) skip flowshop, (b) re-entrant flowshop, (c) compound flowshop, and (d) finite queue flowshop (Thomas and David, 1993)
2.2.2 Job shop scheduling

Job shop is the one that most commonly used in general production system. Job shop scheduling problems with setup times arise in many practical situations such as aerospace industries, fabrication industries, printing industries and semiconductor manufacturing industries (Sharma and Jain, 2015). There are several machines and jobs in the shop where some or all of machines were used in some specific sequence. Jobs are independent to each other. There is a restriction that a job cannot enter the machine more than once. Usually the main objective of job shop scheduling is to minimize the makespan or the tardiness penalty (Sule, 2008). Rui et al.,(2014) used the efficiency of ant colony algorithm for solving the job shop scheduling problem.

A job shop is also known as a multi-stage production system where each job needs to undergo several operations to become a finished product. In a job shop, only a single machine is capable of processing each operation (Chiang and Lin, 2013). This caused blockage in production when any machine breaks down. According to Li et. al (2014), job shop scheduling problem can be generalized into the flexible job shop scheduling problem (FJSP) where two sub-problems are solved. The first one is called a routing sub-problem that assigns each operation to a machine selected from any of a set of suitable machines. The second one is called a scheduling sub-problem consisting of sequencing all the assigned operations on all machines in order to yield a feasible schedule to minimize a predefined performance criterion.

2.2.3 Open shop scheduling

For open shop scheduling, there is no operationally dependent sequence that a job must follow. A job may enter the machines in any sequence that the job needs (Sule, 2008). The term ‘open shop’ is often simply taken to mean a shop that rather than directly to customer order, it produces to final inventory. Just like the flowshop and job shop, the open shop scheduling problem consists of a set of $n$ jobs to be processed on a finite set of $m$ machines. In general, each job must be processed on every machine and consist of a series of $m$ operations which have to be scheduled in
REFERENCES


