DYNAMIC WEIGHTED IDLE TIME HEURISTIC FOR FLOWSHOP SCHEDULING

AMIRA SYUHADA BINTI ZAINUDIN

A thesis submitted in partial 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
DEDICATED

To

My husband, Mr. Muhammad Hafeez
For your love, patience, friendship and making everything possible

My mother, Madam Azlina
A strong and gentle soul who taught me to trust in Allah, believe in hard work and that so much could be done with little

My father, Mr. Zainudin
For earning an honest living for us and for supporting and encouraging me to believe in myself

My research partner a.k.a my bestfriend, Noor Amira Isa
Who help me a lot to finished my thesis

My family and friends
Without whom none of my success would be possible
ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful

Alhamdullilah, all praise to Allah for the strengths and His blessing in completing this thesis. My deep gratitude goes first to my supervisor; Prof. Madya Dr. Sh Salleh bin Sh Ahmad who expertly guided me through my graduate education and who shared the excitement of two years of discovery. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works have contributed to the success of this research. Next to him, my deepest gratitude to my beloved husband Mr. Muhammad Hafeez for gives me a full support and advice during my progress in completing this project. I am feeling oblige in taking the opportunity to sincerely thanks to my parents; Mr Zainudin and Mrs Azlina, whom I am greatly indebted for me brought up with love and encouragement to this stage. At last but not the least I am thankful to all my teachers and friends who have been always helping and encouraging me throughout the year. I have no valuable words to express my thanks, but my heart is still full of the favors received from every person.
The constructive heuristic of Nawaz, Enscore and Ham (NEH) has been introduced in 1983 to solve flowshop scheduling. Many researchers have continued to improve the NEH by adding new steps and procedures to the existing algorithm. Thus, this study has developed a new heuristic known as Dynamic Weighted Idle Time (DWIT) method by adding dynamic weight factors for solving the partial solution with purpose to obtain optimal makespan and improve the NEH heuristic. The objective of this study are to develop a DWIT heuristic to solve flowshop scheduling problem and to assess the performance of the new DWIT heuristic against the current best scheduling heuristic, i.e., the NEH. This research developed a computer programming in Microsoft Excel to measure the flowshop scheduling performance for every change of weight factors. The performance measure is done by using $n$ jobs ($n=6,10$ and $20$) and $4$ machines. The weight factors were applied with numerical method within the range of zero to one. Different weight factors and machines idle time were used at different problem sizes. For 6 jobs and 4 machines, only idle time before and in between two jobs were used while for 10 jobs and 20 jobs the consideration of idle time was idle time before, in between two jobs and after completion of the last job.

In 6 jobs problem, the result was compared between DWIT against Optimum and NEH against Optimum. While in 10 jobs and 20 jobs problem the result was compared between DWIT against the NEH. Overall result shows that the result on 6 and 10 jobs problem the DWIT heuristic obtained better results than NEH heuristic. However, in 20 jobs problem, the result shows that the NEH was better than DWIT.

The result of this study can be used for further research in modifying the weight factors and idle time selections in order to improve the NEH heuristic.
Heuristik konstruktif Nawaz, Enscore dan Ham (NEH) telah diperkenalkan pada tahun 1983 untuk menyelesaikan penjadualan flowshop. Ramai penyelidik telah meneruskan penyelidikan NEH dengan menambah langkah-langkah dan prosedur baru untuk memperbaiki algoritma sedia ada. Oleh itu, satu heuristik baru yang dikenalpasti sebagai Dynamic Weighted Idle Time (DWIT) menggunakan kaedah faktor berat dinamik untuk menyelesaikan penyelesaian separa dengan tujuan mendapatkan optima makespan dan memperbaiki NEH. Objektif kajian ini adalah untuk membangunkan (DWIT) heuristik untuk menyelesaikan masalah penjadualan flowshop dan menilai prestasi heuristik DWIT berbanding heuristik terbaik, iaitu NEH. Kajian ini membangunkan pengaturcaraan komputer dalam Microsoft Excel untuk mengukur prestasi penjadualan flowshop untuk setiap perubahan faktor pemberat. Ukuran prestasi dilakukan dengan menggunakan n pekerjaan (n=6,10 dan 20) dan 4 mesin. Faktor pemberat digunakan dalam julat sifar hingga satu. Faktor pemberat dan masa terbiar yang berbeza telah digunakan pada saiz masalah yang berbeza. Untuk 6 pekerjaan dan 4 mesin, hanya masa terbiar sebelum dan di antara dua pekerjaan telah digunakan manakala, bagi 10 pekerjaan dan 20 pekerjaan pertimbangan masa terbiar adalah masa terbiar sebelum, di antara dua pekerjaan dan selepas selesai tugas terakhir. Dalam masalah 6 pekerjaan, keputusan yang diperolehi telah dibandingkan antara DWIT terhadap Optima dan NEH terhadap Optima. Manakala dalam masalah 10 pekerjaan dan 20 pekerjaan keputusanannya telah dibandingkan antara DWIT terhadap NEH. Secara keseluruhan keputusan yang diperolehi pada 6 dan 10 pekerjaan, DWIT mendapat keputusan yang baik berbanding NEH. Manakala, pada 20 pekerjaan keputusan menunjukkan NEH lebih baik berbanding DWIT. Hasil kajian ini boleh digunakan untuk penyelidikan selanjutnya dalam mengubahsuai factor berat dan pilihan waktu terbiar untuk meningkatkan NEH heuristik.
# TABLE OF CONTENTS

- TITLE i
- DECLARATION ii
- DEDICATION iii
- ACKNOWLEDGEMENT iv
- ABSTRACT v
- ABSTRAK vi
- TABLE OF CONTENTS vii
- LIST OF TABLES viii
- LIST OF FIGURES x
- LIST OF SYMBOLS AND ABBREVIATIONS xi
- LIST OF APPENDICES xii

## CHAPTER 1 INTRODUCTION

1.1 Background of study 1
1.2 Problem statement 3
1.3 Objective of study 4
1.4 Scope of study 4
1.5 Project justification 4
1.6 Thesis layout 5

## CHAPTER 2 LITERATURE REVIEW

2.1 Introduction 6
2.2 Scheduling 7
   2.2.1 Forward scheduling 9
   2.2.2 Backward scheduling 10
   2.2.3 Types of scheduling environments 10
2.2.4 Job shop scheduling 11
2.2.5 Open shop scheduling 16
2.2.6 Batch shop scheduling 18
2.2.7 Flowshop scheduling 18
2.3 Heuristic 23
2.3.1 Nawaz, Enscore and Ham (NEH) heuristic 25
2.4 Sequencing rules 32
2.5 Weighted idle time 33
2.6 Makespan 38
2.7 Analysis literature review 40
2.8 Summary of the chapter 41

CHAPTER 3 METHODOLOGY 42
3.1 Introduction 42
3.2 Research procedure 44
3.2.1 Construct scheduling environment in Microsoft Excel spreadsheets 44
3.2.2 Develop Visual Basic Application for optimum solution 45
3.2.3 Gantt chart 45
3.3 Flowchart of DWIT heuristic and WIT heuristic 46
3.4 Dynamic weighted idle time (DWIT) heuristic 47
3.4.1 Six jobs and four machines 48
3.4.2 Ten jobs and four machines 58
3.4.3 Twenty jobs and four machines 61
3.4 Summary of the chapter 64

CHAPTER 4 RESULTS AND DISCUSSION 65
4.1 Introduction 65
4.2 Six jobs and four machines 65
4.2.1 Result of 100 sets of random data 66
4.2.2 Result of 10 replications of 100 sets of random data 67
4.2.3 Data analysis of six jobs and four machines 69
4.3 Ten jobs and four machines 71
4.3.1 Result of 100 sets of random data 71
4.3.2 Result of 10 replications of 100 sets of random data
4.3.3 Data analysis of ten jobs and four machines
4.4 Twenty jobs and four machines
4.4.1 Result of 100 sets of random data
4.4.2 Result of 10 replications of 100 sets of random data
4.4.3 Data analysis of twenty jobs and four machines
4.5 Comparison result of ten jobs and twenty jobs problem
4.6 Discussion of results
4.7 Discussion of the study
4.8 Summary of the chapter

CHAPTER 5 CONCLUSION AND RECOMMENDATION
5.1 Introduction
5.2 Summary of research
5.3 Conclusion
5.4 Recommendation

REFERENCES
APPENDIX
# LIST OF TABLES

2.1 Summary of scheduling environments 21  
2.2 Processing time of NEH heuristic 29  
2.3 Summary of Nawaz, Enscore and Ham (NEH) heuristic 31  
2.4 Weight factor of WIT heuristic 35  
2.5 Processing time of WIT heuristic 35  
2.6 Summary of weighted idle time 37  
2.7 Analysis of literature review 40  
3.1 Weight factor value for six jobs and four machines 48  
3.2 Non-increasing order of total processing time 49  
3.3 Sequence of first two job 49  
3.4 Total weighted idle time of first two jobs 50  
3.5 Sequences of three jobs 51  
3.6 Total weighted idle time of three jobs 52  
3.7 Sequences of four jobs 53  
3.8 Total weighted idle time of four jobs 54  
3.9 Sequences of five jobs 55  
3.10 Total weighted idle time of five jobs 56  
3.11 Final result of minimum makespan 57  
3.12 Weight factor value of 10 jobs and 4 machines 59  
3.13 Non-increasing order of total processing time 59  
3.14 Sequence for first two job 60  
3.15 Total weighted idle time of first two jobs 60  
3.16 Weight factor value for 20 jobs and 4 machines 61
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.17</td>
<td>Non-increasing order of total processing time</td>
<td>62</td>
</tr>
<tr>
<td>3.18</td>
<td>Sequence for first two jobs</td>
<td>63</td>
</tr>
<tr>
<td>3.19</td>
<td>Total weighted idle time of first two jobs</td>
<td>63</td>
</tr>
<tr>
<td>4.1</td>
<td>Summary of results for DWIT heuristic against NEH heuristic for 6 jobs</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>Heuristic verification result for 6 jobs</td>
<td>68</td>
</tr>
<tr>
<td>4.3</td>
<td>Summary of results for DWIT heuristic against NEH heuristic for 10 jobs</td>
<td>71</td>
</tr>
<tr>
<td>4.4</td>
<td>Heuristic verification result for 10 jobs</td>
<td>72</td>
</tr>
<tr>
<td>4.5</td>
<td>Summary of result for DWIT heuristic against NEH heuristic for 20 jobs</td>
<td>74</td>
</tr>
<tr>
<td>4.6</td>
<td>Heuristic verification result for 20 jobs</td>
<td>75</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Example of job shop scheduling</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Example of flowshop scheduling</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>The two possible schedules for a two-machine two-job no-wait makespan problem</td>
<td>34</td>
</tr>
<tr>
<td>3.1</td>
<td>Flowchart of research methodology</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Example of scheduling environment in Microsoft Excel</td>
<td>44</td>
</tr>
<tr>
<td>3.3</td>
<td>Example of gantt chart of the flowshop scheduling</td>
<td>45</td>
</tr>
<tr>
<td>3.4</td>
<td>Flowchart of DWIT heuristic</td>
<td>46</td>
</tr>
<tr>
<td>3.5</td>
<td>Flowchart of NEH heuristic</td>
<td>47</td>
</tr>
<tr>
<td>4.1</td>
<td>Dotplot of NEH vs OPT, WIT vs OPT and DWIT vs OPT for 6 jobs</td>
<td>69</td>
</tr>
<tr>
<td>4.2</td>
<td>Individual plot of NEH vs OPT, WIT vs OPT and DWIT vs OPT for 6 jobs</td>
<td>70</td>
</tr>
<tr>
<td>4.3</td>
<td>Time series plot of DWIT vs NEH of 100 sets of data for 10 jobs</td>
<td>73</td>
</tr>
<tr>
<td>4.4</td>
<td>Time series plot of DWIT vs NEH of 100 sets of data for 20 jobs</td>
<td>76</td>
</tr>
<tr>
<td>4.5</td>
<td>Individual plot of DWIT vs NEH between 10 jobs and 20 jobs</td>
<td>77</td>
</tr>
<tr>
<td>4.6</td>
<td>Area graph of DWIT vs NEH between 10 jobs and 20 jobs</td>
<td>77</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWIT</td>
<td>Dynamic Weighted Idle Time</td>
</tr>
<tr>
<td>WIT</td>
<td>Weighted Idle Time</td>
</tr>
<tr>
<td>JSSP</td>
<td>Job Shop Scheduling Problem</td>
</tr>
<tr>
<td>FSP</td>
<td>Flowshop Scheduling Problem</td>
</tr>
<tr>
<td>FJSP</td>
<td>Flexible Job Shop Problem</td>
</tr>
<tr>
<td>FJSSP</td>
<td>Flexible Job Shop Scheduling Problem</td>
</tr>
<tr>
<td>PFSP</td>
<td>Permutation Flowshop Scheduling Problem</td>
</tr>
<tr>
<td>OSSP</td>
<td>Open Shop Scheduling Problem</td>
</tr>
<tr>
<td>NEH</td>
<td>Nawaz, Enscore and Ham</td>
</tr>
<tr>
<td>VBA</td>
<td>Visual Basic Application</td>
</tr>
<tr>
<td>IE</td>
<td>Industrial Engineering</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>RA</td>
<td>Rapid Access</td>
</tr>
<tr>
<td>TS</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>SL</td>
<td>Sarin and Lefoka</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>WIP</td>
<td>Work-in-process</td>
</tr>
<tr>
<td>NP-complete</td>
<td>Non-deterministic polynomial time</td>
</tr>
<tr>
<td>NP-hard</td>
<td>Non-deterministic polynomial-time hard</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>100 Sets Generated Data for 6 Jobs Problem</td>
<td>95</td>
</tr>
<tr>
<td>B</td>
<td>100 Sets Generated Data for 10 Jobs Problem</td>
<td>99</td>
</tr>
<tr>
<td>C</td>
<td>100 Sets Generated Data for 20 Jobs Problem</td>
<td>102</td>
</tr>
<tr>
<td>D</td>
<td>Processing Time</td>
<td>105</td>
</tr>
<tr>
<td>E</td>
<td>Coding</td>
<td>113</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Background of study

Industrial Engineering (IE) is a branch of engineering that has strong connection with the management. Industrial engineering is more concerned with the design of production and service system. It can be said to overlap with operational management, operational research and manufacturing engineering. Industrial engineering is correlated to productivity and quality. The role of industrial engineer is to ensure that productivity and quality management are maintained and even increased over time. The job as industrial engineer requires ability to analyse and specify integrated components of people, machines, materials, and facilities to create efficient and effective systems that will produce goods and services beneficial for human being (Savory, 2005).

Scheduling is known as the process of arranging, controlling and optimizing work in a production process or manufacturing process. Scheduling is the procedure of generating the schedule which is a physical document and generally informs the happening of things and demonstrate a plan for the timing of certain activities. Generally, scheduling problem can be approached in two steps; in primary step sequence is planned or decides how to choose the next job. In the next step, planning of start time and possibly the completion time of each job is performed (Malik and Dhingra, 2013).
Generally, scheduling is required in the manufacturing process and particularly in engineering (Nayan, 2015). It is the process of arranging works in a production. It involves the generating of a schedule on how to organize more than one task or process. The main purpose of the scheduling system in the industry is to increase the productivity and to reduce both the processing time and operating costs. Moreover, scheduling process can be regarded as a decision-making process. It is important to ensure that the process can achieve the target within a certain period of time. In order to obtain the optimal solution, effective and efficient scheduling is necessary.

Flowshop scheduling is one of the classes of scheduling problems other than job shop scheduling and open shop scheduling. Flowshop scheduling is a special case where there is strict order for all operation to perform all jobs. It is very interesting area of study to be applied in a manufacturing process. Optimum result can be obtained from the processing time of each machine. Besides, in flowshop scheduling, a series of machines process the same jobs in sequence and the sequence to process this job is the same for each machine (Nayan, 2015).

According to Modrák and Pandian (2010), in a shop floor of the industry, the routings which are based upon the jobs that need to be processed on different machines are one among the major activities. Therefore, the resource requirements are not based on the quantity as in flowshop but rather the routings for the products produced. However, both flowshop and job shop scheduling is to find a sequence of jobs on given machines and the objective is to minimizing the completion times for the production.

One of the problems solving technique for scheduling is by using heuristic algorithm. It is suitable approach to solve the large scale scheduling problems. In such case, heuristic algorithms find approximation solution but acceptable time and space complexity play indispensable role (Kokash, 2005). This algorithm is just to find a solution that is closest to the best result easily in a short time.
1.2 Problem statement

The Nawaz, Encore and Ham (NEH) algorithm proposed by Nawaz et al., (1983) uses the powerful job insertion technique after arranging the jobs in the descending order of their total processing times. It selects the first two jobs as the initial partial sequence and other jobs are inserted one by one from the third job to obtain a final optimal makespan and its corresponding sequence. It has been generally agreed that the NEH algorithm is known as one of the best available simple, constructive heuristic even today (Baskar, 2016). But, NEH heuristic is not the best one for flowtime optimisation (Allahverdi and Aldowaisan, 2002). Thus, this study has developed a new heuristic known as Dynamic Weighted Idle Time (DWIT) method by adding dynamic weight factors for solving the partial solution with purpose to obtain optimal makespan and improve the NEH heuristic. Based on Baskar (2016), NEH which has been introduced in 1983 is still the best known constructive heuristic to solve flowshop scheduling problem with makespan objective. For makespan objective function, the NEH always uses makespan even in deciding partial schedule arrangements. Weighted idle time is one of the newly proposed concepts for flowshop scheduling due to its potential to produce better result than NEH heuristic (Saleh, 2014). This proposed method utilizes the total weighted idle time for solving partial schedule before finally use makespan as the final decision of complete schedule. Based on Saleh (2014), from a total of 25 sets of data, 44% produced the same makepan performance for both weighted idle time and NEH solution. Another 36% showed that idle time produced better performance than NEH heuristic. Whereas, the remaining 20% showed that the NEH heuristic was the best. Therefore, this research proposal is intended towards conducting further in-depth investigations, experiments and analysis to show that a new heuristic based on the modified version of the weighted idle time can have the ability to compete with the NEH.
1.3 Objectives of study

The objectives of the study are as follows:

i. To develop a new heuristic identified as Dynamic Weighted Idle Time (DWIT) heuristic to solve flowshop scheduling problem.

ii. To evaluate the performance of the new DWIT heuristic against the current best scheduling heuristic, i.e., the NEH.

1.4 Scope of study

This research focused on the following:

i. Apply dynamic weighted idle time method by changing the weight factors at each of sequencing step.

ii. Randomised data was generated by using Visual Basic Application programming.

iii. The range of weight factors value to measure the flowshop scheduling performance for every changes of weight factors are within (0.0 ~ 1.0).

iv. Makespan criteria were used to identify the best performance of flow shop scheduling.

v. The performance measure is done by using 6 jobs 4 machines, 10 jobs 4 machines and 20 jobs 4 machines.

1.5 Project justification

Dynamic idle time weight factors were introduced as the manipulated variables for the scheduling. This study performance measure was done by using 6 jobs 4 machines, 10 jobs 4 machines and 20 jobs 4 machines. This performance measurement is based on previous study that starts with 6 jobs and 4 machines (Bareduan and Hasan, 2012). Thus, this study continues the performance measurement with 10 jobs 4 machines and 20 jobs 4 machines with a new method of study. The performance measurement study was done until 20 jobs only due to the simulation will take more time and several days to complete for bigger job numbers. Based on Seda (2007), the research also used 10 jobs and 20 jobs for the permutation
flowshop scheduling problem. When used the high number of jobs, the search of optimum in the space of permutations of jobs ended with a run time error so the research need to find another approach to compute the optimal solution for the job more than 20. Sahu (2009) compared the four heuristics in flowshop scheduling up to 10 jobs and 5 machines. From the analysis, it has been proved that NEH heuristic shows the minimum value of makespan when compared to other heuristic (Gupta’s heuristic, RA heuristic, CDS heuristic and Palmer’s heuristic) for most of the problems but limited to 4 machines problems. As the machine size increases, RA heuristic produced the best results (Malik and Dhingra, 2013). This study also used 4 machines to minimize the makespan and idle time. Therefore, based on Sahu (2009), this study focused to limit to 4 machine problems and try to improve the makespan performance. This study tested many different dynamic weight factors for idle time in order to obtain better performance of flowshop scheduling. The result was compared with the optimum makespan to evaluate its performance. This project identified the best dynamic idle time weight factors suitable for problems identified in scope of study. The finding of this investigation contributes to the area of flowshop scheduling solutions using constructive heuristic.

1.6 Thesis layout

In this thesis, the brief introduction and discussion about the literature review and research from other researchers are stated in Chapter 2. Besides, the methodology and the development of a new algorithm method of this research are highlighted in Chapter 3. Moreover, the experimental validation performance result of the new purposed algorithm heuristic is presented in Chapter 4. Finally, the research contributions, conclusion with future recommendation are discussed in Chapter 5.
CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses about the scheduling which is known to be very important in the production and industrial system. Scheduling is a decision-making process that was used on a regular basis in many manufacturing and service industries. Scheduling deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. One type of dynamic scheduling strategy is used to dispatching rules to determine when a resource become available and which task that resource should do next. The resources and tasks in an organization can take many different forms. There were maybe machines in a workshop, runaways at an airport, and crews at a construction site, processing units in a computing environment and so on. Each task may have a certain priority level, an earliest possible starting time and also a due date. So, there were different objective that need to be achieved. One objective maybe the minimization of the completion time of the last task, or maybe the minimization of the number of task completed after their respective due dates. Overall this chapter includes section about scheduling, heuristic, weighted idle time and makespan.
2.2 Scheduling

Scheduling is one of the important areas in the field of production management. It also the most necessary tool for decisions making process in engineering and manufacturing. Scheduling flowshop problem can be addressed as the setting with penalties for tardiness in delivering customer orders, as well as cost for holding both finished goods and work-in-process inventory (Bulbul et al., 2004 and Kemppainen, 2005).

Scheduling occurs in a very wide range of economic activities. It always involves accomplishing a number of things that tie up various resources for period of time. The resources are in limited supply. The things to be accomplished may be called “jobs” or “projects” or “assignments” and are composed of elementary parts called “activities” or “operations” and “delays”. Each activity requires certain amounts of specified resources for a specified time called the “process time” (Morton and Penticio, 1993).

According to Sule (2008), a schedule shows the planned time when processing of a specific job will start on each machine that the job requires. It also represents when the job will be completed on every machine. Thus, it is timetable for both jobs and machines. The starting time of a job on the first machine in its sequence of operation should be assuming zero lead time for the job. In a typical real-world scheduling problem, the set of jobs changes dramatically over time and the processing times of jobs are affected by various types of uncertainty. The goal is to determine how the available machine processing time is to be allocated among competing requests with the objective of optimizing the performance of the system. In general, methods for solving dynamic scheduling problems must address the combinatorial structure inherited in most interested scheduling problems (Terekhov et al., 2013).

Efficient scheduling is how manufacturing companies minimize the cost and punctuality to meet customer with the promised due date (Heizer and Render, 2014b). Although there has been an increasing interest in modelling and solving scheduling problems in dynamic and uncertain environments, scheduling research has mostly focused on devising effective method for solving deterministic problems with a complex combinatorial structure (Bidot et al., 2009, Aytug et al., 2005, Chaudhuri and Suresh, 1993). On the other hand, scheduling problems with a simpler
combinatorial structure but with stochastic and dynamic characteristic have been studied for a long time.

According to Watanabe et al. (2005), scheduling is the allocation of resources to perform a set of tasks over a period of time. Many real scheduling problems in the manufacturing industries are quite complex and very difficult to be solved by conventional optimization techniques. To develop a schedule, the processing time for each job on each machine the job requires must be known. To calculate the processing time for a job, it must consider both machines and job dependant factor such as setup time, unit processing time, machine speed, quality factors and also the number of unit needed. A machine schedule also displays the time when the machine is idle. Idle time occur because of no job is available for processing or because all jobs are being processed on other machines. When a machine is idle, it is the best plan to stop for maintenance activities so that no productive time is taken away from the machines.

The developing of effective and efficient scheduling approaches is necessary for the optimal solution purpose. Based on Heizer and Render (2014a), efficient scheduling is how manufacturing companies minimize the cost and punctuality to meet customer with the promised due date. The scheduling theory concern about the problems of allocating and prioritizing of customer orders correspond to an available facility. Effective scheduling depends on matching the schedule to performance.

The right technique of scheduling depends on the volume of orders, the nature of operations, the overall complexity of jobs and also the importance placed on each of four criteria (Heizer and Render, 2014a):

- Minimize completion time.
  - It is evaluated by obtaining the average completion time.
- Minimize customer waiting time.
  - It is evaluated by determining the average number of late hours or days.
- Minimize work-in-process (WIP) inventory.
  - A direct relationship exists between the number of jobs in the system and WIP inventory. Therefore, the less the number of jobs in the system, the lower the inventory.
Maximize utilization.

- Utilization is decided by determining the percentage of the time when the facility is utilized.

The four criteria that have been mentioned above are used to analyse the scheduling performances. Moreover, the good scheduling techniques must be simple, clear, easy to understand, easy to carry out, flexible and realistic (Heizer and Render, 2014a).

For some scheduling environments, it is perfectly valid to assume that job processing time are deterministic in which the implicit enumeration techniques and heuristics appears in the literature can be utilized (Aydilek and Allahverdi, 2009). However, for some other scheduling environments, the assumption of deterministic processing times may not be applicable. As stated by Sorouch (2007), the random variation in processing times needs to be taken into account while searching for a solution.

2.2.1 Forward scheduling

Forward scheduling or also known as in push mode operations, the provider send work along in the absence of any call from the customer. In this mode, the providers determine when and what is the work flow. In other words forward scheduling start the processing when a job is received. Some system used this approach, for example, radio and television station. Many manufacturers have a good flow because of the provider choose the work flow instead of a customer demanding the work flow. The schedule starts from its start time until the whole process is finished without considering its due date. Lova (2002) mentioned that forward sequence is built completing a partial sequence by scheduling each activity as early as possible (and following the establish order).
2.2.2 **Backward scheduling**

Backward scheduling or pull scheduling 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).

Backward scheduling 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. According to Lova (2002), the backward schedule passes starts from feasible schedule processing the activities in decreasing order of its feasible finish time. The backward sequence is built completing a partial sequence by scheduling each activity (following the establish order) as late as possible in the window delimited.

2.2.3 **Types of scheduling environments**

According to Sule (2008), in production planning terminology, scheduling models has been divided into the following categories:

i. **Single machine**
   - Jobs are processed by the machine one at a time. Each job has a processing time and due date and also may have other characteristics for example priority. The most important objective is to sequence jobs on the machines so as to minimize the penalty for being late (tardiness penalty).

ii. **Flowshop**
   - Jobs are processed on multiple machines in an identical sequence. However, the processing time of each job on each machine may be different. The goals for flowshop is to minimize the time required for completion of all jobs, called the makespan.

iii. **Parallel machines**
   - A number of identical machines are available and jobs can be processed on any one of them. Jobs may have dependency which is the next job in
REFERENCES


Kokash, N. 2005. An Introduction to Algorithms. Department of Informations and Telecomunications University of Thento, Italy.


Savory, P. 2005. Details and Description of Industrial Engineering, *Industrial and Management Systems Engineering Faculty Publications*. 33


