Group 25

Pre-Project Report

The study of scheduling and machine balancing in flexible manufacturing system

Submitted by

Mr.	Krittapad	Harnchang	ID	603 15019 21
Mr.	Chayanin	Jitupatham	ID	603 15060 21
Mr.	Chawakorn	Faiboon	ID	603 15077 21

Senior Project Committee Members

Asst. Prof. Pisit	Jarumaneeroj	(Ph.D.)	Advisor
Asst. Prof. Pramual	Suteecharuwat	(Ph.D.)	Committee
Asst. Prof. Napassavong	Osothsilp	(Ph.D.)	Committee

Department of Industrial Engineering, Faculty of Engineering,

Chulalongkorn University, Academic Year 2020

Advisor's signature ....

Abstract

Flexible Manufacturing System (FMS) is a manufacturing system which is a highly automated group of machine cells. FMS can change production to fit the number of needs and product types, but the problem is the scheduling and conveying complexity that arises from the ever-changing production process. The processing has to be changed inevitably. If sequencing or conveying is not flexible, it will cause production delays and difficult to control production costs . Therefore, the objective of this research is to minimize delay and maximize utilization of machines to lower costs. We have adopted the logistic principle to solve the scheduling problem by applying a math model to the solution to achieve more efficient results, which will be used in production planning. The results can be used to improve the delay of each job and utilization of machines by comparing the results with the real operation time.

Table of contents

Chapter 1 : Introduction

1.1 Background

Most recently, the 4th Industrial Revolution was a combination of the virtual and physical worlds to work together (cyber-physical systems). Activities were transformed into electronic (digitalization) such as cloud storage instead of documents. Everything is connected via the Internet network (Interconnection) with automation and Artificial Intelligence that can control machines to communicate data with each other and large data analytics (Big Data Analytics), the current approach of the industry is called in different ways, such as the US called Smart Manufacturing, Europe call Factories of the Future (FoF), or in German, Industry 4.0.

Flexible manufacturing systems(FMS) is a main part of industry 4.0. By continuous demand sensing and With a highly automated group technology machine cell, consisting of a group of processing workstations that are interconnected by an automated material handling and storage system, and controlled by a distributed computer system. Each machine has an automatic tool changer (ATC) that can change tools automatically and hence consecutive operations can be performed with negligible setup times. This feature allows for FMSs to produce various types of parts with high machine utilization. However, such a performance can be achieved if indispensable operational decisions are made more efficiently and effectively.The goal of flexible manufacturing concept is to cope with fluctuating customer demands with high system throughput, i.e. maximizing productivity and flexibility at the same time.

The flexibility in FMS enables the production system to share its resources in processing a wide array of products at the same time,thus increasing the overall utilization and reducing delay. However, the scheduling in FMS is mostly composed of NP-Hard combinatorial problems which are difficult to solve and develop the relevant optimal polynomial algorithm. In this research, the researchers will study scheduling problems in FMS and improve schedule by using the Logistics algorithm. The solution of this study aims to experiment on multi objective heuristics on FMS.

This study considers the scheduling problem in FMS. The problem covers two decisions : 1) determining the sequence of jobs to start operation, 2) determining the machine and material handling equipment assignment for each operation and subsequently setting the routing path for each job. These problems need to be considered simultaneously to satisfy the multi-objective of minimizing tardiness and maximizing utilization of machines. Minimizing tardiness is closely related to reducing production cost in terms of penalty charging . Maximizing proper utilization of overall machines means better return of investment(ROI), return of asset(ROA) and also productivity. The Scope of the project covers 6 main points : 1) we were interested in FMS only in the field of factories manufactured using CNC machines. 2) There are no assembly steps for this problem. 3) Fully automated machine. 4) We do not focus on material handling equipment problems. 5) There is no set up time in machines. 6). Any batches can distract to operate(Non batch mode).

Since the problem is classified as an NP-Hard problem, the use of heuristics is obliged. This research uses variable neighborhood search(VNS) as an improvement heuristic approach to solve the problem. First, this research creates an initial solution by using the earliest due date(EDD) and first in first out (FIFO) method and creating a network of FMS to determine job schedules, job paths, machines and material handling equipment assignment. Then, collect delay and utilization of machines as a benchmark. Next, improve the solution by using the VNS algorithm. The VNS algorithm procedure in this research begins with using an intra route operator in order to switch job assignments in each machine and then using an inter route operator(Relocation,Exchange) to improve the initial solution in order to switch job assignments on other machines. Lastly, using a shaking operator to avoid local search. After that, heuristics will repeat 3 operators until they find the best solution of minimizing delay and maximizing utilization of overall machines.

The researcher aims to focus on the solution of how to solve the problem by using a logistic algorithm. We aim for the big three problems in scheduling that are why we do multi- objective optimization. The result of our solution will be a work schedule that minimizes delay and maximizes utilization of machines.

Chapter 2

Literature review

In literature, there are many studies relating to utilization, delays and makespans problems but none of the studies dealing with all problems at the same time. Most studies only solve one objective. The researchers want to propose a multi optimization model consisting of utilization of machine, delays and makespans. We aim to use a two-phase heuristic approach and Before talking about the model, we will inform you of FMS and the problems characteristics.

FMS can be divided into 2 types by number of machines and level of flexibility. Type of FMS that is divided by the number of machines has 3 types. First, Single machine cell (SMC) consists of one CNC machining center combined with a parts storage system for unattended operation. Completed parts are periodically unloaded from the parts storage unit, and raw work-parts are loaded into it. The cell can be designed to operate in either a batch mode or a flexible mode or in combinations of the two. When operated in a batch mode, the machine processes parts of a single style in specified lot sizes and is then changed over to process a batch of the next part style. When operated in a flexible mode, the system satisfies three of the four flexibility tests. It is capable of (1) processing different part styles, (2) responding to changes in production schedule, and (4) accepting new part introductions.

Second, Flexible manufacturing cell (FMC) consists of two or three processing workstations (typically CNC machining centers or turning centers) plus a part handling system. The part handling system is connected to a load/unload station. In addition, the handling system usually includes a limited parts storage capacity. One possible FMC is illustrated in Figure 2.2 .

Finally, the Flexible manufacturing system (FMS) has four or more processing workstations connected mechanically by a common part handling system and electronically by a distributed computer system. Thus, an important distinction between an FMS and FMC is the number of machines: an FMC has two or three machines, while FMS has four or more." *A* second difference is that the FMS generally includes non-processing workstations that support production but do not directly participate in it. These other stations include part/pallet washing stations, coordinate measuring machines, and so on. A third difference is that the computer control system of FMS is generally larger and more sophisticated, often including functions not always found in a cell, such as diagnostics and tool monitoring. These additional functions are needed more in an FMS than in FMC because the FMS is more complex.

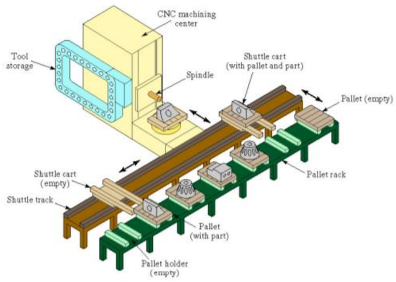

Figure 2.1 single machine cell (Osullivan, D. 1999)

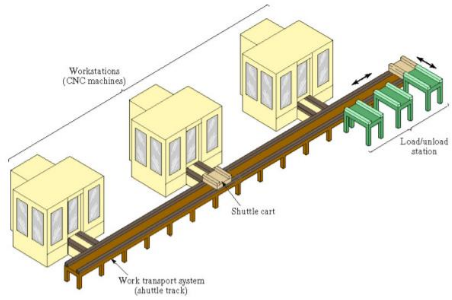

Figure 2.2 flexible manufacturing cell (Osullivan, D. 1999)

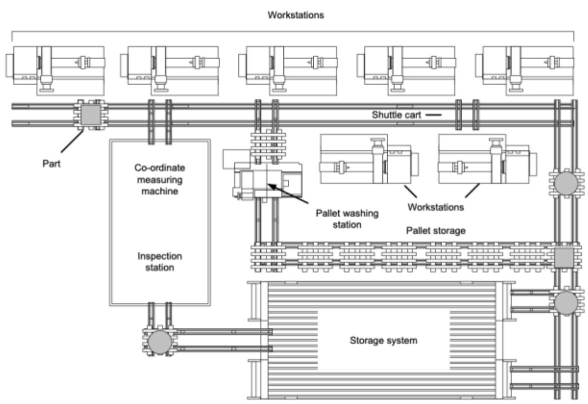

Figure 2.3 Flexible manufacturing system (Osullivan, D. 1999)

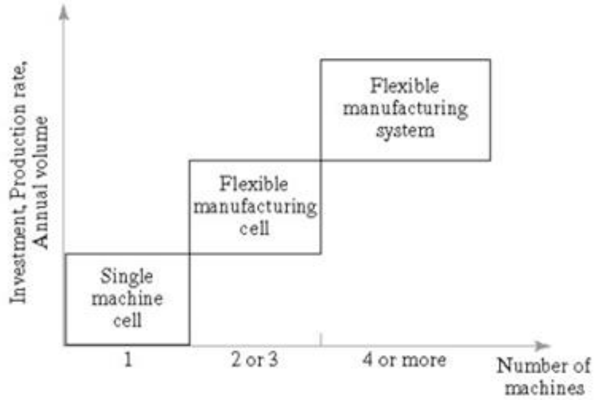

Figure 2.4 Comparison between 3 type of Flexible manufacturing (Osullivan, D. 1999)

Determining the optimal solution to Vehicle routing problem (VRP) is NP-hard, so the size of problems that can be solved, optimally, using mathematical programming or combinatorial optimization may be limited. Therefore, commercial solvers tend to use heuristics due to the size and frequency of real world VRPs they need to solve.

The researcher compared scheduling methods with VRP by looking at a comparative overview, for example: comparing vehicles with machines , place compared with order and each node compared to work. This is how it compares to VRP with time windows.

2.1 Multi - Objective Optimization

Multi-objective optimization (also known as multi-objective programming, vector optimization, multicriteria optimization, multi attribute optimization or Pareto optimization) is an area of multiple criteria decision making that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. Multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Minimizing cost while maximizing comfort while buying a car, and maximizing performance whilst minimizing fuel consumption and emission of pollutants of a vehicle are examples of multi-objective optimization problems involving two and three objectives, respectively. In practical problems, there can be more than three objectives. (Meittinen, K., 1999)

In 2010, Prakash, Chan and Deshmukh's study of FMS scheduling with knowledge based genetic algorithm approach showed that the multi-objective of maximize throughput and minimize mean flow is solve by Simple Genetic Algorithm (SGA) and The Knowledge Based Genetic Algorithm (KBGA). Another research said that scheduling is an important method for solving the mixed-model two-sided assembly lines problem to reach maximum production efficiency. Many factors such as multiple objectives and learning effects have to be considered in solving the sequencing problem. These make the problem more complicated as known as "NP-Hard problem". In this research, a Biogeography-based Optimization (BBO) algorithm is adopted for solving the sequencing problem to minimize variance of production rates, utility work, and setup time. The results are compared with well-known algorithms such as Non-dominated Sorting Genetic Algorithms (NSGA-II) and Discrete Particle Swarm Optimization (DPSO). The experiments show that BBO has performed a convergence of 98.67% and ratio 76.13%, consequently, which are better than NSGA-II and DPSO.(Naruemitwong & Chutima, 2013)

Moreover, Rifai, Nguyen, Aoyama, Dawal, and Masruroh (2018) found that non-dominated sorting biogeography-based optimization(NSBBO-trapezoidal) models performed favorably and were comparable to standard NSBBO and NSGA-II. This is the study of scheduling problems of FMS having multi loading-unloading and shortcuts infused in the reentrant characteristics which are considered in minimizing makespan and total earliness.

Finally, most reference research uses genetic algorithms (GA) to solve the multi objective in FMS. However, it would be better if we can use another method to find the solution.

2.2 Heuristic Based Scheduling

Scheduling is the process of arranging, controlling and optimizing work and workloads in a production process or manufacturing process. Scheduling is used to allocate plant and machinery resources, plan human resources, plan production processes and purchase materials.

Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process. In manufacturing, the purpose of scheduling is to minimize the production time and costs, by telling a production facility when to make, with which staff, and on which equipment. But it's for academic purposes. From a business point of view, the first priority purpose is to keep the customer's due date. Most major factories ask for scheduling to smooth flow production, level the production, keep safety stock, keep cycle time, or keep assigning jobs to auto-machines or lines as the next priority.

An algorithm for scheduling batches of parts in a multi-cell flexible manufacturing system. (Das & Canel ,2005) This research performed the solution of scheduling problems in multi-cell FMS with flow shop characteristics which considered the material handling systems by using strong branch and bound algorithms. In 2015, Candan and Yazgan's study showed that the FMS scheduling problem is one of NP hard problems. So, the researchers use a metaheuristic called genetic algorithm (GA) in order to minimize the makespan of the problem. Also in the scheduling-location(ScheLoc) problem, which is a combination of two complex problems: the machine-location problem and scheduling problem, this study investigates a deterministic and discrete parallel-machine ScheLoc problem for minimizing the makespan. A new mixed integer programming formulation based on network flow problems is proposed. A polynomial-time heuristic is designed for efficiently solving large-scale problems (Shijin Wang, Ruochen Wu, Feng Chu, Jianbo Yu and Xin Liu, 2019). Another problem that relates to the FMS Planning Problem(FMSPP) is Multiple-day Music Rehearsal Problems (MMRP). In the MMRP model, Music pieces may vary in terms of time and performer. The objective is to minimize the total cost of show-up and waiting time by all players. The MMRP is similar to FMSPP where additional tools(player) must be installed in the machine before processing and improper orders scheduling would create waiting time that can result in higher makespan and delay (Jarumaneeroj & Sakulsom, 2019).

Moreover, The researcher got some knowledge from the tablet film coating. The tablet film coating optimizes in tardy jobs and makespans by using heuristics approach to solve. In this research, the initial solution was created by Earliest Due Date and Longest Processing time. the improvement of the answer was done by using Variable neighbourhood search (VNS) ,we will describe VNS through our study in next topic, and the result of this research we set it to be an instance.(Pontrakul & Jarumaneeroj, 2019)

In scheduling problems of FMS , There are many improvement heuristics such as VNS, ALNS, GA and branch and bound.

2.3 Machine Loading Problem

Nagarjuna, Mahesh and Rajagopal (2006) found that the objective of the machine loading problem considered is to minimize the system unbalance while satisfying the technological constraints such as availability of machining time and tool slots. In this article, the loading problem in random type FMS, which is viewed as selecting a subset of jobs from the job pool and allocating them among available machines, is considered. A heuristic based on a multi-stage programming approach is proposed to solve this problem. Another research, optimum loading of machines in a FMS,the objectives are to maximize profitability and utilization of the system. The proposed model assigns operations to different machines considering capacity of machines, batch-sizes,processing time of operations, machine costs, tool requirements, capacity of tool magazine. A genetic algorithm (GA) is then proposed to solve the formulated problem. (Abazari , Solimanpur & Sattari, 2011)

Moreover, in article of A robust cardinality-constrained model to address the machine loading problem showed that Several deterministic models have been proposed in the literature to solve the machine loading problem (MLP), which considers a set of product types to be produced on a set of machines using a set of tool types, and determines the quantity of each product type to be produced at each time period and the corresponding machine tool loading configuration. However, processing times are subject to random increases, which could impair the quality of a deterministic solution. Thus, we propose a robust MLP counterpart, searching for an approach that properly describes the uncertainty set of model parameters and, at the same time, ensures practical application. We exploit the cardinality-constrained approach, which considers a simple uncertainty set where all uncertain parameters belong to an interval, and allows tuning the robustness level by bounding the number of parameters that assume the worst value. The resulting plans provide accurate estimations on the minimum production level that a system achieves even in the worst conditions. The applicability of the robust MLP and the impact of robustness level have been tested on several problem variants, considering single- vs multi-machine and single- vs multi-period MLPs. We also consider the execution of the plans in a set of scenarios to evaluate the practical implications of MLP robustness. Results show the advantages of the robust formulation, in terms of improved feasibility of the plans, identification of the most critical tools and products, and evaluation of the maximum achievable performance in relation to the level of protection. Moreover, low computational times guarantee the applicability of the proposed robust MLP counterpart.(Lugaresi, Lanzarone, Frigerio & Matta, 2020)

System unbalance, Under-utilized time of a machine is treated as the unused capacity of that machine whereas over-utilized time is considered as the overload on a machine. Some articles available in the literature, e.g. Prakash, Tiwari, and Shankar (2008), Kumar, Prakash, Tiwari, Shankar, and Baveja (2006) and Swarnkar and Tiwari (2004) have considered system unbalance as the sum of under-utilized and over-utilized times while under-utilized times are treated as positive values and it is considered as negative values for overutilized times.

2.4 Variable Neighbourhood search

Variable neighborhood search (VNS),proposed by Mladenović & Hansen in 1997, is a metaheuristic method for solving a set of combinatorial optimization and global optimization problems. It explores distant neighborhoods of the current incumbent solution, and moves from there to a new one if and only if an improvement was made. The local search method is applied repeatedly to get from solutions in the neighborhood to local optima. VNS was designed for approximating solutions of discrete and continuous optimization problems and according to these, it is aimed for solving linear program problems, integer program problems, mixed integer program problems, nonlinear program problems, etc.

Systematic change of neighborhood within a local search algorithm yields a simple and effective metaheuristic for combinatorial optimization. We present a basic scheme for this purpose which can be implemented easily using any local search algorithm as a subroutine. Its effectiveness is illustrated by improvements in the GENIUS algorithm (N., Mladenović & P., Hansen, 1997).

From the tablet film coating, they improved heuristics by a series of improvement operators – 2OPT, RELOCATION, and SWAP – mimicking the concept of Variable Neighborhood Search (VNS) until no improvement could be found. The results from our proposed heuristics are comparatively good, when compared to those of the optimization model, in terms of solution quality as the gap is less than 2% for instances of 10-30 orders.

In this research, we aim to do three steps of improvement heuristics from the concept of VNS. First, we do intra-route. Intra-route is a node exchanging in the same list which means swapping the work sequence in the same machine. Next, inter-route is most likely to intra-route but inter-route is a node exchanging between lists also means swapping work sequences between machines. Finally, if intra-route and inter-route can not find a better solution, we need to do shanking. Shaking is a random sequence with condition. It is a tool that helps to avoid local solutions so we can have more chances to find a better tour.

Chapter 3
Methodology

3.1 Problem Description

The Flexible Manufacturing that we have studied, the system will consist of a number of CNC Machines (Computer Numerical Control) which can work for some operation in some job. FMS is negligible the set up time. Each job has a different processing time and procedure. Moreover, the material handling system is required to feed raw materials to machines and move work-in-process from current machine to next machine to

The design and use of flexible manufacturing systems (FMS) involve some intricate operations research problems. FMS design problems include, for example, determining the appropriate number of machine tools of each type, the capacity of the material handling system, and the size of buffers. FMS planning problems include the determination of which parts should be simultaneously machined, the optimal partition of machine tools into groups, allocations of pallets and fixtures to part types, and the assignment of operations and associated cutting tools among the limited-capacity tool magazines of the machine tools. FMS scheduling problems include determining the optimal input sequence of parts and an optimal sequence at each machine tool given the current part mix. FMS control problems are those concerned with, for example, monitoring the system to be sure that requirements and due dates are being met and that unreliability problems are taken care of. This paper defines and describes these FMS problems in detail for OR/MS to work on.

From above, The Flexible manufacturing system is a multi - objective problems that consist of

1.Machine Loading Problem with non-batch mode (MLP_{nb}) - utilization of overall machines.

2.FMS Scheduling Problem with non-batch mode ($FMSP_{nb}$) - interest in time to sequence jobs or machines.

For the above, our solution is to create a production plan that takes into account the delay of each job . Including utilization of machines at the same time.

There are many models available that can be applied to help answer some of the preceding problems. Each model can structure the problems differently. Each model ignores or aggregates some features of the system to focus on particular aspects. The models have provided either operational or qualitative insights into some of the FMS decision problems.

3.2 Data Preparation

1.Order information - will be the input data for production planning. The Manufacturer will know all orders and deadlines before start production.

- Job Format can be used to estimate how long the production takes. The format of work in one order can be the same or different.
- Quantity produced - how many pieces of each type of product you want to produce.
- Date and time information for delivery

3.3 Mathematical Formulation

Indices

- I is a set of pieces
- J is a set of batches
- M is a set of machines.
- $(I,J)_{\emptyset}$ is a set of $(I,J) \cup \emptyset$

Set of parameters

- T_{ij}^{m} is a processing time of piece i in order j on machine m (minute)
- D_{j} is a due date of batch j
- Q_{j} is quantity of batch j
- e_{ij} is a ready time for piece i in batch j to manufacture

Decision variables

- $x_{ij-i'j'}^{m}$ is a binary decision variable that piece i in batch j on machine m is produced before piece i' in batch j' on machine m
- S_{ij}^{m} is starting time of piece i in batch j on machine m
- A_{ij}^{m} is a tardy time of piece i in batch j on machine m
- B_{ij}^{m} is a finish time of piece i in batch j on machine m before due date
- $Y_{ij} = \begin{cases} 1: when\ system\ manufacture\ piece\ i\ in\ batch\ j\ late \\ 0: Other \end{cases}$
- $Z_{j} = \begin{cases} 0: when\ system\ manufacture\ batch\ j\ late \\ 1: Other \end{cases}$

- C_{max} is a total operating time of system or makespan (minute)
- C^m is a total operating time of machine m (minute)

Objective function

There are 2 objective functions that we consider to achieve in the study.

Delay

$$min\ z_1 = \sum_{i,j,m \in (I,J,M)} Y_{ij} A_{ij}^{m}$$

(1.1)

From Equation (1.1). The first objective function of the FMSPnb is focused on reducing the tardy time resulting from late deliveries.

Utilization of overall machines

$$max\ z_2 = \underline{C} = \frac{\sum_{m \in M} (\frac{C^m}{C_{max}} * 100)}{M}$$

(1.2)

From Equation (1.2), the second objective function is interesting in increasing overall machine utilization. It is percent of overall utilization of overall machines.

Constraints

Demand constraints

- Out from depot
  - $\sum_{i,j \in (I,J)} x_{\emptyset - ij}^{m} = 1 \quad ; \forall m \in M$

    (2)
- Back to depot

- $\sum_{i,j\in(I,J)} x^{m}_{ij-\emptyset} = 1 \quad ; \forall m \in M$ (3)

The constraints in Equations (2) and (3) require that the sequences of production on the automated machine have starting and ending points at the reference point (Depot).

- Out degree
  - $\sum_{i'\in I} \sum_{j'\in J} x^{m}_{ij-i'j'} = 1 \quad ; \forall m \in M \;, \forall (i,j) \in (I,J)$ (4)
- Flow balancing
  - $\sum_{i',j'\in(I,J)} x^{m}_{i'j'-ij} - \sum_{i',j'\in(I,J)} x^{m}_{ij-i'j'} = 0; \forall m \in M \;, \forall (i,j) \in (I,J)$ (5)

Equations (4) and (5) are given that the flow paths of the Flexible production line in each automated machine are continuous, with any item being produced on one machine and only once. After the product list is manufactured, it must either continue producing the product list and connect it to the beginning reference point to indicate the end of the production sequence.

- Assignment
  - $\sum_{m\in M} \sum_{i\in I} x^{m}_{ij} = Q_j \quad ; \forall j \in J$ (6)

Time Window Constraints

- starting time for the depot
  - $S^{m}_{ij} \geq S^{m}_{\emptyset} \quad ; \forall m \in M, \forall (i,j) \in (I,J)$ (7)
- starting time for all
  - $S^{m}_{i'j'} \geq S^{m}_{ij} + T^{m}_{ij} x^{m}_{ij} - M(1 - x^{m}_{ij}) \quad ; \forall m \in M, \forall (i,j) \in (I,J)$ (8)

- $S_{ij}^{m} \geq e_{ij}$; $\forall m \in M, \forall (i,j) \in (I,J)$ (9)

From inequality (7) and (8), the initial time of item ij is determined on machine m, where the initial time of list item ij must be greater than or equal to the completion time of the previous item. However, the start time must be within the time frame, which is set by the inequality (9), because item ij will only be able to start production in the period e_{ij} (ready time) to D_j (due date).

- $S_{ij}^{m} + T_{ij}^{m} - D_j = A_{ij}^{m} + B_{ij}^{m}$; $\forall m \in M, \forall (i,j) \in (I,J)$ (10)

Equation (10) indicates the delay of item ij on machine m. Considering Equation (10), it is found that when the production of item ij is delayed, the resulting value on the left side of Equation (10) is positive. As a result, A_{ij}^{m} is positive and B_{ij}^{m} is zero. On the other hand, if the item ij is produced before the delivery deadline, The resulting value on the left side of Equation (10) will be negative therefore A_{ij}^{m} will be zero and B_{ij}^{m} will be positive .

- $C^{m} \geq S_{ij}^{m} + T_{ij}^{m}$; $\forall m \in M, \forall (i,j) \in (I,J)$ (11)
- $C_{max} \geq C^{m}$; $\forall m \in M$ (12)

Equation (11) indicates the completion time of the system from the time that the final product was finished.

- $\sum_{i \in I} \sum_{m \in M} A_{ij}^{m} \leq M(1 - Z_j)$; $\forall j \in J$ (13)
- $-\sum_{i \in I} (Y_{ij} - 1) \leq MZ_j$; $\forall j \in J$ (14)

The inequality (13) and (14) show the relationship between the decision variable A_{ij}^{m}, Z_j and Y_{ij} to indicate the delay of batch j, i.e., if batch j is delayed or $A_{ij}^{m} > 0$ will result in $Z_j = 0$ and $Y_{ij} = 1$. On the other hand, if batch j is finished in time or $A_{ij}^{m} = 0$ will result in $Z_j = 1$ and $Y_{ij} = 0$.

Boundary Constraint

- $x_{ij}^{m} \in \{0,1\}$;$\forall m \in M, \forall (i,j) \in (I,J)$ (17)
- $y_{ij} \in \{0,1\}$;$\forall (i,j) \in (I,J)$ (18)
- $Z_j \in \{0,1\}$;$\forall (i,j) \in (I,J)$ (19)
- $S_{ij}^{m} \geq 0$;$\forall m \in M, \forall (i,j) \in (I,J)$ (20)
- $A_{ij}^{m} \geq 0$;$\forall m \in M, \forall (i,j) \in (I,J)$ (21)

These boundary equations and inequalities represent a range of decision variable values.Equations (17)-(19) define the value of the decision variable 0 or 1 (Binary integer), while the (20) and (21) inequalities define the boundaries of the decision variable with a non-negative value (Non-negativity decision variables).

Chapter 4

Procedure

4.1 Instance

The used FMS scheduling instances are derived from Optimum loading of machines in FMS benchmark instances with 80 pieces. Benchmark instances are divided in three classes according to the machine utilization, makespan and delays(tardy jobs) .

In Table 4.1, The first column is Batch name that is arranged from A to H and the second column is batch size which means amount of work in batch.The third column, operation number, means the process that needs to be done to finish the work. And the fourth to seventh are the times

Batch	Batch size	Operation number	machine1 (unit processing time) (hr.)	machine2 (unit processing time) (hr.)	machine3 (unit processing time) (hr.)	machine4 (unit processing time) (hr.)	Entry date	Due date
A	8	1	18	-1	12	15	1	120
B	9	1	-1	25	21	27	1	417
		2	-1	24	20	-1		
C	13	1	26	23	-1	-1	1	614
		2	-1	-1	11	15		
		3	-1	8	12	9		
D	6	1	14	13	-1	17	1	183
		2	19	18	15	21		
E	9	1	-1	22	19	16	1	456
		2	25	17	-1	26		
		3	-1	-1	-1	9		
F	10	1	16	-1	12	-1	1	230
		2	7	13	9	7		
G	12	1	19	19	23	-1	1	694
		2	13	16	10	-1		
		3	-1	23	26	-1		
H	13	1	-1	25	18	25	1	738
		2	7	7	15	10		
		3	-1	24	31	18		

that each machine needs to do the process and the last two columns are startdate and due date.

Table 4.1 Example for Optimum loading problem[Amir Musa Abazari]

4.2 Heuristic approach

The researchers also developed a two-stage heuristic to find the solution to this problem; the heuristics presented in this research are two-phase heuristics. In other words, in Step 1, the initial answer is generated from simple basic dispatch rules: Earliest Due Date (EDD) or other(Machine loading), and then develop the answer by Improvement heuristics, i.eVariable Neighborhood Search (VNS) Heuristic Method. In the Exact approach, the researchers will create a method by CPLEX.

4.2.1 Innitial solution

The researchers create an initial solution by separating any items in a batch, scheduling work for any machines and material handling equipment in the system by due date, minimum makespan and balanced operating time of any machine.

1. Earliest Due date (EDD) gives the job with the earliest due date based on assigned due dates the highest priority.
2. First-In, First-Out (FIFO) is a method for organising the manipulation of a data structure where the first entry is processed first.

4.2.2 Improvement of the answer

1. Shaking is a random transfer of any items from any machine to another. Shaking is used to avoid local solutions and find another feasible solution when other improvement heuristics can not find a better solution.
2. Relocation is a random transfer of any item from one machine to another. Relocation is classified as an Inter-route improvement operator.

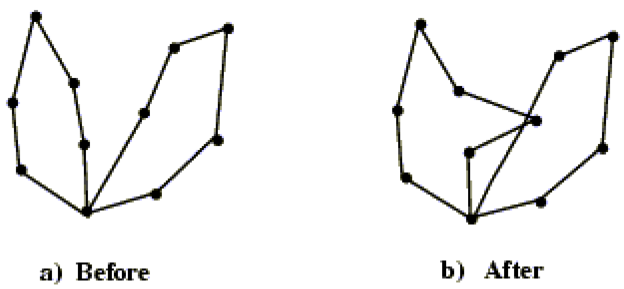

Figure 4.1 Relocation [Van Breedam 1994]

3. Exchange is another type of Inter-route enhancement which differs from Relocation where Exchange will randomly swap one pair of items from all batches on any two machines.

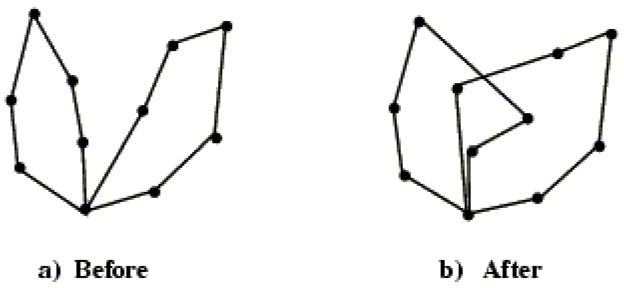

Figure 4.2 Exchange [Van Breedam 1994]

4. Intra route exchange is similar to Exchange but it will swap in the same machine.

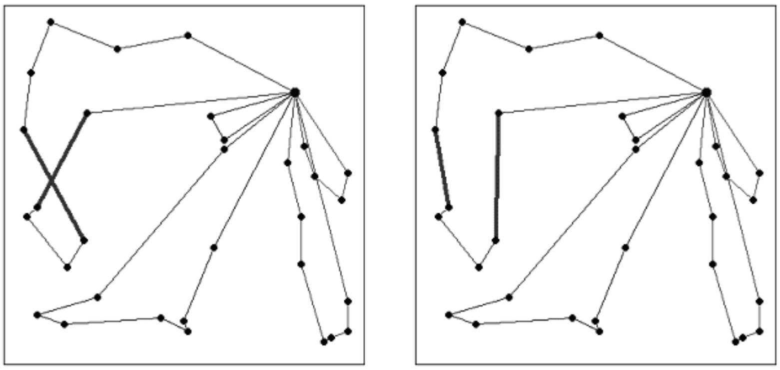

Figure 4.3 Intra-route Exchange [Heitor Silvério Lopes]

In addition to the steps mentioned above, we refer to the VNS basic principles of how to think and structure the solution. In the algorithm below show how the VNS works with shaking, local search and neighbourhood change.

Algorithm : VNS

```
Function VNS (x, kmax, tmax)
repeat
    k ⟵ 1
    repeat
        x' ⟵ Shake(x, k)                    // Shaking
        x'' ⟵ Best Improvement(x' )                // Local search
            repeat
                x' ⟵ x
                x ⟵ argmin_{ f(y) }, y ∈ N(x)
            until ( f(x) ≥ f(x') )
            return x'
        x ⟵ NeighbourhoodChange(x, x'', k)         // Change neighbourhood
            if f (x') < f(x) then
                x ⟵ x'          // Make a move
                k ⟵ 1           // Initial neighborhood
            else
            k ⟵ k+1                 // Next neighborhood
    until k = kmax
    t ⟵ CpuTime()
until t > tmax
```

4.3 Pseudo code

Pseudo code is a term which is often used in programming and algorithm based fields. It is a methodology that allows the programmer to represent the implementation of an algorithm. Simply, we can say that it’s the cooked up representation of an algorithm.It is simply an implementation of an algorithm in the form of annotations and informative text written in plain English. It has no syntax like any of the programming languages and thus can not be compiled or interpreted by the computer. We create some pseudocode before programming so this is our pseudo code for initial and improvement.

Pseudo Code for initial solution

```
import raw data (Batch,Piece,Process, Process time, Startdate and Duedate)
set machine and mhe
while All batch not finish
    while All piece in selected batch not finish
        count time
        select piece by EDD or FIFO
        if some mhe not running
            select most unload time mhe
        if some machine not running
            select most unload time machine
        count load/unload time in machine and mhe , current process in
        --machine and mhe
        mhe pick selected piece
        mhe go to selected machine
        check done of machine
            if current piece has next process
                set target to mhe
            else
                set target is finish good to mhe
            mhe go to target
        check done of mhe
    end while
end while
return Job schedule in each mhe and machine
```

Pseudo Code for improvement heuristic

```
import Job schedule from initial solution
x = true
while x = true
    #Call Intra route
    if has a better solution
        return a better tour to be an optimal solution
        call Intra-route (10,000 times)
    else :
        break and terminate Intra route
    end if

    #Call Inter route
    if has a better solution
        return a better tour to be an optimal solution
        call Inter route (10,000 times)
    else :
        break and terminate  Inter route
    end if

    #Call Shake
    if has a better solution
        return a better tour to be an optimal solution
        call shake again
    else :
        break and terminate Shake
        x = False(terminate)
end while
return  best-found minimizing makespan also delay and balance operating time of any machines
```

Pseudo Code for improvement heuristic : Intra route (insertion)

```
x = Job schedule from initial solution
temp = []
        while temp is not feasible
        temp = x
        m = random machine from list temp machine
        for p in machine m.work_list
                for i in range(len(m.work_list))
                        if m.work_list[i] != p
                                remove p from m.worklist
                                m.work_list[i] = p
                        if m.worklist is better and feasible
                                x = temp
                        else :
                                temp = x
end while
return x
```

Pseudo Code for improvement heuristic : Inter route (exchange)

```
x = Job schedule from initial solution
temp = []
while temp is not feasible
        temp = x
        m1 = random machine from list temp machine
        m2 = random machine from list temp machine
        for p in machine m.work_list
                if m2 can work on process p
                        for i in range(len(m.work_list))
                                remove p from m.worklist
                                m.work_list[i] = p
                                if temp is better and feasible
                                        x = temp
```

```
                    else :
                            temp = x
end while
return x
```

4.4 Implementation and optional

The researchers thought that by comparing the heuristic to the factory operations, we thought that the results may indicate that such heuristic can reduce time off and The number of delays that can occur but the results may vary from case to case depending on the product catalog and the number of items that have been produced in the weekly order. This heuristic can be used to decide production planning. Adaptation of this heuristic may be suitable for the factory as follows.

1) No setup time
2) Non-batch mode production
3) Hurry due date
4) Long processing time
5) Difference processing time in each machine
6) Some machine can not do some job

Chapter 5

Result & Discussion

Performance of the proposed model is evaluated based on some benchmark problems from Optimum Loading Problems in which the raw data of this problem is available. The proposed heuristics algorithm has been coded in python and run on a PC, cpu intel core i5-7500 CPU@3.40GHz . The initial solution and number of iterations are the most critical parameters affecting the performance of algorithms.

5.1 Operational time and response quality

The result of the improvement heuristic is compared to the number of jobs in terms of program run time and quality of the result.

Table 5.1 program run time in heuristics vary by number of order

No of order	Proposed Heuristic		Improvement
	EDD	FIFO	
	Avg time (second)		Avg time
80	4.085	4.294	96.75
160	7.173	7.193	163.87
240	9.462	9.602	239.59

In terms of calculation time, from table 5.1 showed that the number of orders had small sample size,the number of orders were smaller than 80 orders, the program run time of proposed heuristics were not significantly different. However in a bigger sample size, there was more program run time to be processed. In the program run time of improvement heuristics, there were significantly much more than the run time of proposed heuristics. When we increased the number of orders, it had more effect on program run time. Thus from this problem, it is not possible to find answers from mathematical models.

5.2 Result discussion

From finding initial solutions by using the proposed heuristics and data set of optimum loading instance, the first outcome from EDD was shown in figure 5.2.1. The outcome showed that machine no.1 and no.4 had significantly larger non operating time than machine no.2 and no.3. This is the result of comparing in condition of the dataset because not every machine can operate every job. Moreover, machine no.2 and no.3 had more finished goods than machine no. 1 and no. 4 and the result of utilization was a mean of 64.862% and delay of every piece which was 3,658 minutes. When we compared this solution with the solution of the relaxation constraint (EDD adjust) as shown in table 5.2.1 below which the result of utilization was a mean of 82.598% and delay of every piece which was 2821 minute , the difference utilization and delay of them are 17.735% and 22.881% , respectively. So, EDD adjust was represented as a comparison of quality among solutions. In this case, the solution from EDD proposed heuristics was not good enough.

Table 5.2.1 result of utilization and delays in EDD proposed heuristic and EDD adjust

	EDD	EDD adjust	%difference
Utilization	64.863%	82.598%	17.735%
Delays(minute)	3658	2821	22.881%

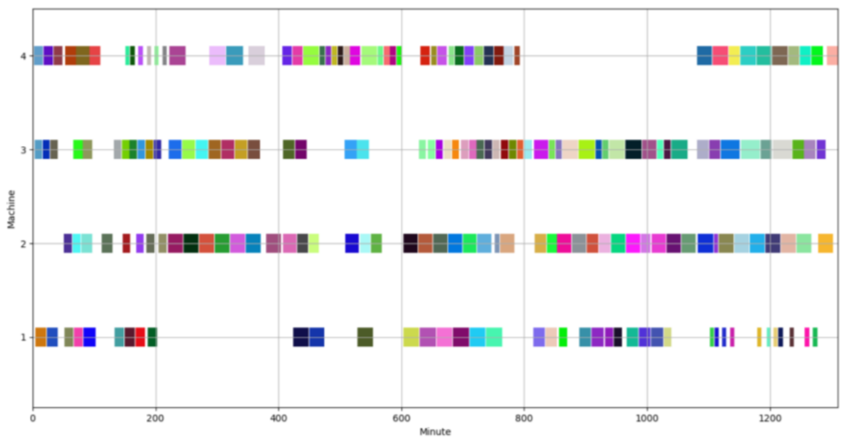

Figure 5.2.1 Gantt chart of job sequence by using EDD

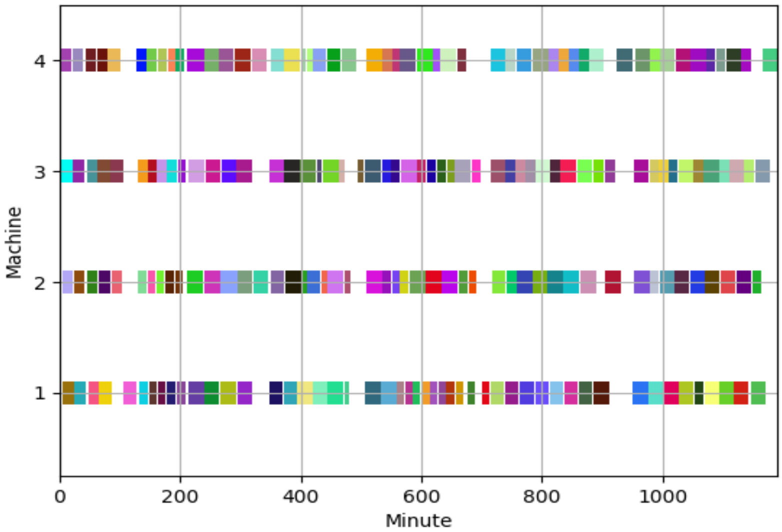

Figure 5.2.2 Gantt chart of job sequence by using EDD adjust

Then, the results from FIFO were obtained as shown in the table below. For the basic FIFO, the utilization was found to be at 64.831%, while the delay could be identified as 4,218 minutes. On the other hand, the utilization of FIFO adjust was 82.688% and the delay was 2,944 minutes. The differences between those two utilizations and delay was calculated to be 17.837% and 14.557%, respectively.

Table 5.2.2 result of utilization and delays in FIFO proposed heuristic and FIFO adjust

	FIFO	FIFO adjust	%difference
Utilize	64.831%	82.668%	17.837%
Delays(minute)	4218	2944	30.203%

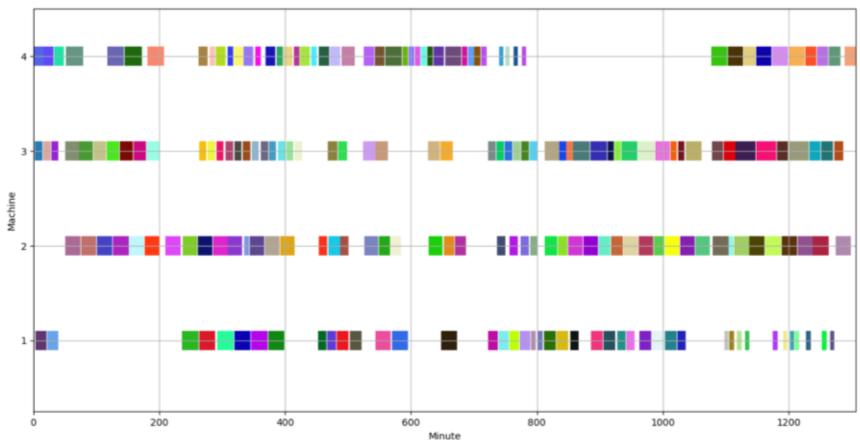

Figure 5.2.3 Gantt chart of job sequence by using FIFO

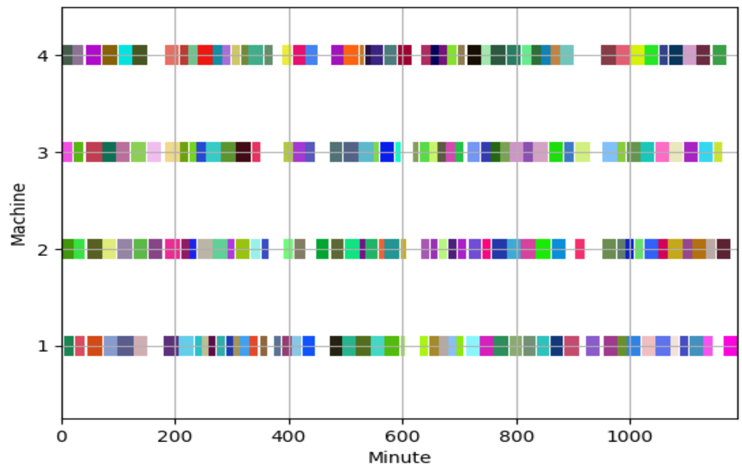

Figure 5.2.4 Gantt chart of job sequence by using FIFO adjust

Number of iterations or parameter n was the main factor which had a huge effect on the improvement of the solution because increasing the number of iterations could expand more neighbourhood change. So, it is possible to discover broader solutions and a new best solution as shown in table 5.2.3 below. When we improved the initial solutions (EDD and FIFO) , the result of 10 iterations showed that every value was better than the initial solution. Then we moved to 20 iterations, it was a result that made every value much better as we can see the lower of makespans and delay but higher utilization. In this case, we collectำก improving solutions from n = 10, 20, 30, 40, 50 because time of running a program more than 1 hour could cause difficulty in experiment.

Table 5.2.3 The result of improving initial solutions vary by number of iterations

No of iteration	EDD		FIFO	
	best Delays (minute)	best utilization	best Delays	best utilization
10	3326 (65.36)	65.570%(3600)	3779 (65.13)	67.30 (4110)
20	3237 (66.850)	66.850% (3237)	3678 (66.25)	66.250 %(3686)
30	3179 (66.864)	67.231% (3361)	3451 (67.3)	68.59% (3547)
40	2969 (68.887)	68.887% (2969)	3403 (64.51)	68.36% (3620)
50	3040 (68.65)	68.903% (3088)	3593 (67.85)	68.24% (3647)

As illustrated in figure 5.2.5 and 5.2.6, these were the solutions after improving the initial solutions with more than 50 iterations. Each point in the scatter plot is a feasible solution which is screened from improvement heuristics. In Y axis means delay and X axis means utilization of the overall machines. The more the points tend to go down the lower right line, the better. We found that the solution of EDD after improvement was better than the solution of FIFO after improvement in the range of 50 - 100 iterations. In the utilization aspect, the best solution of EDD after improvement was 68.903% of utilization and 3,088 minutes of delay. In the delay aspect, the best solution of EDD after improvement was 68.65% of utilization and 3,040 minutes of delay. Thus, the best value of utilization and the best value of delay were not the same point.

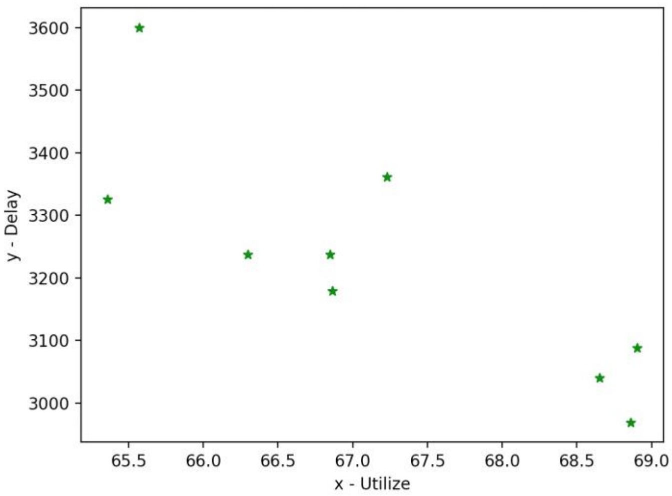

Figure 5.2.5 Scatter plot of EDD after improvement

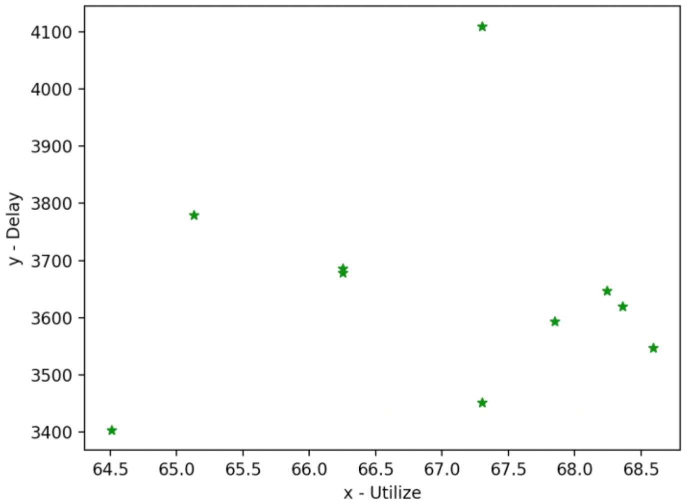

Figure 5.2.6 Scatter plot of FIFO after improvement

Next, we experimented that the number of material handling equipment had an effect on utilization and delay. In this case, we set the condition of EDD as an initial solution, 10 iterations and 4 machines. Then, we varied the number of MHE to get the result.

Table 5.2.4 The best known answer of varying number of MHE in terms of utilization of machine and delay by using EDD as an initial solution

No of MHE	EDD	
	Best delays (Utilization of)	Best utilization (Delay of …)
1	3651 (60.901%)	61.103% (3665)
2	3353 (64.912%)	65.061% (3425)
3	3387 (64.272%)	64.272%(3387)
4 (base)	3365(65.302%)	65.302%(3365)
5	3291(66.261%)	66.261%(3291)
6	3193(65.529%)	65.529%(3193)
7	3369 (64.305%)	65.280% (3508)

The results were illustrated in table 5.2.4, when we compared another number of MHE with base case of 4 MHE (65.302% of utilization and 3,365 minutes of delay), there were not significant differences from base case, in case of 2, 3, 5, 6 and 7 MHE. However, if we lowered the number of MHE to 1, the utilization of the best known answer reduced to 61.103% and delay increased to 3,651 minutes.

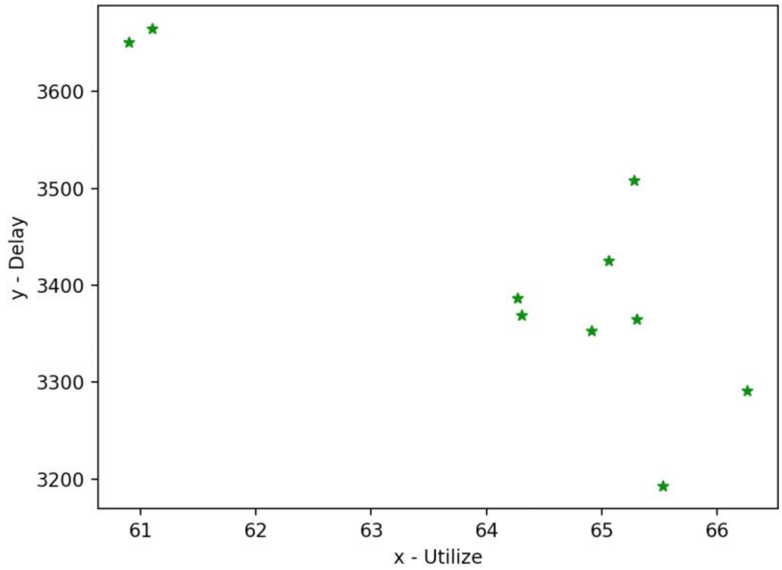

Figure 5.2.7 Scatter plot of the best known answer in terms of utilization of machine and delay by using EDD as an initial solution and varying the number of MHE

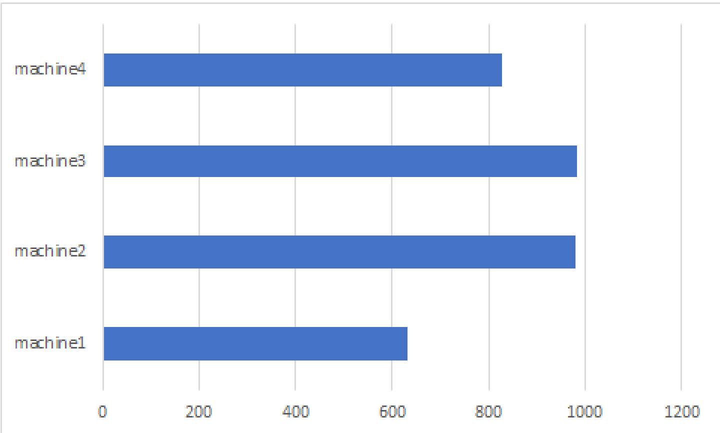

Figure 5.2.8 Bar chart of total operating time on each machine of best known utilization point

We analysed the best known utilization point which was 68.903% of utilization and 3,088 minutes of delay. As shown in figure 5.2.8, machine no.1, no.2, no.3, no.4 had 630, 979, 982, 828 minutes of total operating time, respectively.

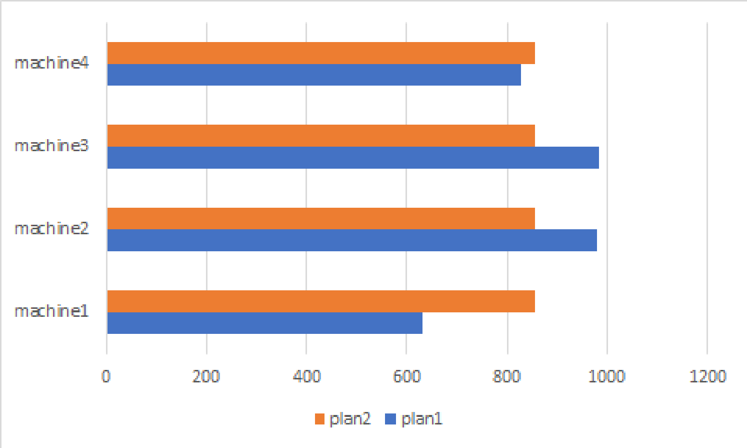

Figure 5.2.9 Bar chart of total operating time on each machine of best known utilization point (Blue) and benchmark instance(orange)

From optimum machine loading article (Prakash et al., 2008), we referenced this research as an benchmark instance. We measured the value of balancing in the machine by using the System Balance equation or SU. This equation is the average of the total operating time of the overall machine. In this case, SU is 854.75 minus the over or under utilization of each machine as in equation (1).

$$SU = |854.75 - 630| + |854.75 - 979| + |854.75 - 982| + |854.75 - 828| \quad (1)$$

$$SU = 224.75 + 124.25 + 127.25 + 26.75 \quad (2)$$

$$SU = 503 \quad (3)$$

When we researched the benchmark instance in MLP problem 1 found that the researcher named Prakash, Mulkhopadhyay, Nargarjuna calculated the value of SU as 318, 122 and 122 minutes, respectivetive. To compare with our experiment, our value of SU was 503 minutes, our SU were significantly more than the benchmark instance because there are more constraints in multiobjective which we were interested in.

Table 5.2.5 Comparing the best known solution with EDD adjust and FIFO adjust in term of Utilization and delay

	EDD adjust	FIFO adjust	Best Known solution	%difference from EDD and best known	%difference from FIFO and best known
Utilization	82.598%	82.668%	68.903%	13.695 %	13.765 %
Delays(minute)	2821	2944	2969	5.018%	0.842%

In table 5.2.5, we compared EDD adjust, FIFO adjust and Best known solution, we found that the best known solution had lower utilization from both of initial adjustments which was 13.695% and 13.756%, respectively. In case of delay, the best known solution had higher from both of initial adjustments which was 5.018% and 0.842%, respectively. Both initial adjustments represent a lower bond in order to approximately estimate the percent gap of the best known solution and optimal solution.

Chapter 6
Conclusion

This paper presents a mathematical programming model for solving FMS scheduling problems and machine loading problems in flexible manufacturing systems. .This model has a lot of conditions so we cannot find a solution by using a math model. So, we use a two-phase heuristics approach that generates initial solution by basic scheduling (EDD&FIFO) and generates a better solution by improving heuristic that calls VNS through the three steps of improvement intra-route, inter-route and shaking. The proposed model takes into account many practical parameters including number of machines, number of material handling equipment, requirement of each work and number of iteration. An effective solution approach that we found based on multi objective and variable neighbourhood search. Some benchmark problems adopted from the literature are solved by proposed heuristic. Computational results indicate that the proposed model provides the promising solution. Even Though, compare some solutions with those available in the literature was not good enough and It is worth noting that application of the proposed model is limited to certain cases where there are sufficient numbers of material handling in the shop floor. The work may be extended further by imposing constraints on the availability of these resources.

Solution of the attempted problem identifies assignment of machines to different operations of the selected jobs. This solution should then be used to determine the sequence of jobs on different conditions ex. machine, workload, capacity. This problem is addressed as the scheduling in the literature. This issue can be attempted in future to yield more efficient utilization of resources.

In this research, we aim to study the Flexible manufacturing system problem and use of Logistics algorithms to find optimal solutions. In limited time and crisis situations, some results are not significantly better than the benchmark instance. We had some problems with the structure of the computational method so it takes a lot of time to fix the structure and recreate. So far, we could find a better solution by increasing the number of iterations and it will take a lot of computer processing time to find a better one. The results of research do not clearly solve all problems but the issue can be attempted in future by using another heuristics based and creating another condition for improvement phase.

Reference

Browne, J., Dubois, D., Rathmill K., Sethi, SP. & Stecke, K.E. (1984). Classification of flexible manufacturing systems. *The FMS Magazine 2,* 2, 114.

Buzacott, J.A. (1982). The fundamental principles of flexibility in manufacturing systems. Conf. on Flexible Manufacturing Systems.Brighton, U.K.

Buzacott, J.A. & Yao, D.D.W. (1982). A review of models, Flexible manufacturing systems. Ontario, Canada.

Stecke, K.E. (1985). Useful models to address FMS operating problems, Advances in Production Management Systems.Budapest, Hungary. Material Flow 2, 2, 179.

Stecke, K.E. & Browne, J. (1985). Variations in flexible manufacturing systems according to the relevant types of automated materials handling. Material Flow 2*,* 2/3, 179

Suri, R. (1984). An overview of evaluative models for flexible manufacturing systems. ORSA/TIMS Special Interest Conf on Flexible Manufacturing Systems: Operations Research Models and Applications. Ann Arbor, Michigan.

Suri, D. An overview of evaluative models for flexible manufacturing systems, Proc. 1st ORSA/TIMS Special Interest Conf on Flexible Manufacturing Systems: Operations Research Models and Applications. Ann Arbor, Michigan.

Osullivan, D. (1999). Unit15 flexible manufacturing systems. Retrieved from http://www.nuigalway.ie/staff-sites/david_osullivan/documents/unit_15_flexible_manufacturing_systems.pdf

Inman, R.A. (1991). flexible manufacturing : issues and implementation. Retrieved from Flexible Manufacturing Systems: Issues and Implementation" by Inman, R. Anthony - Industrial Management, Vol. 33, Issue 4, July/August 1991 | Online Research Library: Questia Reader

Lugaresi, G., Lanzarone, E., Frigerio, N. & Matta, A. (2020). A robust cardinality-constrained model to address the machine loading problem. Retrieved from Science Direct website : https://www.sciencedirect.com/science/article/pii/S0736584518301157

Candan, G. & Yazgan, H.R. (2015). Genetic algorithm parameter optimization using Taguchi method for a flexible manufacturing system scheduling problem. Retrieved from scopus website : https://www.scopus.com/record/display.uri?eid=2-s2.0-84912523404&origin=inward&txGid=b64de0d5fae88e25a7c3f87575eb78f7

Gamila, A.M. & Motavalli, S. (2003). A modeling technique for loading and scheduling problems in FMS. Retrieved from Science Direct website : https://www.sciencedirect.com/science/article/abs/pii/S0736584502000613

Das, R.S. & canel, C. (2005). An algorithm for scheduling batches of parts in a multi-cell flexible manufacturing system. Retrieved from Science Direct website : https://www.sciencedirect.com/science/article/pii/S0925527304003196

Prakash, A., Chan, T.S. F. & Deshmukh, S.G. (2011). FMS scheduling with knowledge based genetic algorithm approach. Retrieved from Science Direct website : https://www.sciencedirect.com/science/article/pii/S0957417410009061

Dong-Kyu, L., Jeong-Hoon, S. & Dong-Ho, L. (2020). Operations scheduling for an advanced flexible manufacturing system with multi-fixturing pallets. Retrieved from Science Direct website : https://www.sciencedirect.com/science/article/pii/S0360835220302308

Rifai, P.A., Nquyen, H.T., Aoyama, H., Dawal, M.Z. & Masruroh, A.N. (2018). Non-dominated sorting biogeography-based optimization for bi-objective reentrant flexible manufacturing system scheduling. Retrieved from Science Direct website : https://www.sciencedirect.com/science/article/pii/S1568494617306592#bib0035

Naruemitwong, W. & Chutima, P. (2013). Multi-Objective Sequencing Problem on Mixed-Model Two-sided Assembly Lines under Learning Effect. The journal of KMUTNB, 23(1), 115 - 128.