Vol 6 No 1 (2024) *e*-ISSN: 2682-8383

Journal of Engineering Technology and Applied Physics

# Investigation of Flexible Job Shop with Machine Availability Constraints

Nurul Izah Anuar[1,*], Muhammad Hafidz Fazli Md Fauadi[2] and Adi Saptari[3]
[1]*Faculty of Engineering and Technology, Multimedia University, Ayer Keroh, Melaka, Malaysia.*
[2]*Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia.*
[3]*Department of Industrial Engineering, President University, Cikarang Baru, Bekasi, Indonesia.*
[*]*Corresponding author*: nurulizah.anuar@mmu.edu.my, *ORCiD*: 0000-0002-0168-0963

https://doi.org/10.33093/jetap.2024.6.1.8
*Manuscript Received: 11 August 2023, Accepted: 27 September 2023, Published: 15 March 2024*

***Abstract* — In real-world scheduling applications, machines may be unavailable during certain time periods for deterministic and stochastic reasons. This situation does not pose any problems if the jobs always have more than one machine available for processing. However, it becomes an issue if the only available machine is the one which more than one job needs for processing. Thus, the investigation of limited machine availability, along with the practical requirement to handle this feature of scheduling problems, are of huge significance. This paper examined a flexible job shop environment of a manufacturing firm to optimize different performance criteria related to makespan, due dates, priorities and penalties by using a metaheuristic approach, taking into account the precedence constraints. The work investigated the case in which all machines are available for processing and another case in which some of the machines are known in advance to be unavailable. From the results, the best schedules are analysed and the perspectives of the findings to decision-makers are discussed with the purpose of achieving high machine utilization, cost reduction and customer satisfaction.**

***Keywords—Flexible job shop, Machine availability constraints, Production scheduling, Manufacturing, Metaheuristic optimization***

## I. INTRODUCTION

Flexible job shop environments take place mainly in industries where each customer order has specific characteristics and order sizes are relatively small. A typical example of a real-life flexible job shop environment is wafer fabrication in the semiconductor industry in which an order usually implies a batch of a certain type of item that has to go through the manufacturing facility according to a given route with specific processing times. A flexible job shop can be considered as a general case of the parallel machine and the job shop environments. Instead of *m* machines in series, the machines are divided into a number of work centres that have to be scheduled. A work centre may consist of a single machine or a bank of identical machines in parallel. An operation of a job in the flexible job shop can be scheduled on several machines in a work centre [1]. Hence, every operation has a pre-determined list of machines that would be able to process that particular operation. Recirculation may occur in a flexible job shop when a job on its course through the shop visits a work centre more than once.

In real-world manufacturing environments, the operations of a job being scheduled would have different processing times and different processing orders on machines. The jobs would also have agreed delivery dates, and the solution procedure differs as the goal of the scheduling varies [2]. From the practical standpoint, the decision-makers have many concerns about what and how certain measures can improve: (1) throughput rate, (2) customer satisfaction, particularly, on-time deliveries, and (3) cost reduction, for instance, inventory holding cost. The second and third criteria are in line with the Just-in-Time (JIT) principle of making a product specifically when it is needed. This decreases lateness fines (tardiness) and storage costs (earliness) [3].

In addition, the scheduling environments are very complex with the existence of parallel machines and are due to many constraints such as machine unavailability and machine conflict that can heavily impact day-to-day operations [4]. Parallel machines are commonly needed in a real-life setting to prevent the system from being deterred by the unavailability (e.g. breakdown) of a single machine [5]. These parallel machines will not pose any problems to the

Journal of Engineering Technology and Applied Physics (2024) 6, 1, 8:55-66
https://doi.org/10.33093/jetap.2024.6.1

Published by MMU PRESS. URL: https://journals.mmupress.com/index.php/jetap/index

decision-makers if the jobs always have more than one machine to choose from for processing. However, when the alternative machines become unavailable, which can be due to breakdowns, busy (occupied), repair or maintenance, it will become an issue if the only available machine is the one which more than one job contends for. Machine conflict happens if more than one job competes for one machine [6].

Most literature regarding production scheduling predominantly adopt the assumption that all machines are available continually for processing and that all jobs can be scheduled at any available machine throughout the whole planning span [7]. These assumptions may be justified in several cases but they may not always hold in some practical environment, since most of the real-world problems of production planning are dynamic, such as machine preventive maintenance and machine breakdowns that can occur during any instant, causing one or a number of machines to become not available for job processing. In addition, machines could be halted to execute planned maintenance activities, such as washing or control procedures. This would mean the input data should be frequently updated during the planning horizon [5].

Machines may be unavailable during certain time periods for deterministic and stochastic (random) causes. In deterministic cases, machines would be subject to preventive maintenance where starting periods and durations are known in advance. In general, preventive maintenance is planned to preserve the equipment and enhance overall accessibility. In stochastic cases, the machines would be subject to unpredictable breakdowns that would disturb the production activities until the machines are restored [8]. Therefore, this research work considers the case in which all machines are assumed to be available and another case in which some of the machines are known in advance to be unavailable. The rest of the paper is structured as follows: The mathematical model of the flexible job shop problem, covering the model constraints and assumptions are provided in Section II, followed by the justification of the objective functions implemented in this study and the metaheuristic approach applied. Section III presents a case study of the flexible job shop model taken from a real manufacturing system, including the machine availability constraints. In Section IV, the investigation and analysis of the case study from the standpoint of four scenarios subject to two different cases of machine availability are carried out, and the results are discussed before summarizing the findings. Finally, the conclusion is presented in Section V.

## II. PROBLEM DESCRIPTION

### A. *Mathematical Model*

Mathematically, a flexible job shop problem can be described in the following manner: There are $c$ work centres, where at every work centre, there are several identical machines in parallel. There are $n$ jobs, each to be scheduled on $m$ machines one at a time; job $i$ requires processing on only one machine at every work centre and any machine would do. The jobs are assigned pre-determined machine sequences and have specific processing times; however, the machine orders are random from job to job [1]. The operations are scheduled to be processed by particular machines, observing a pre-determined sequence, named the precedence constraints, where the machine orders are different in each job. The precedence constraints, that prescribe particular arrangements of operations, impose certain complexity on the flexible job shop.

The mathematical model of the problem can be formulated as: minimize $f_b(s)$, where given a feasible schedule $s$, set $S$ is a finite set of all feasible schedules such that $s \in S$. $f_b(s)$ is the $b^{\text{th}}$ objective function, where $b$ = 1, 2, 3 and 4, as described below:

1) Minimization of makespan, $C_{max}$ ($b$ = 1):

$$f_1(s){:}\ C_{max} = \max_{1 \le i \le n}(C_i) \tag{1}$$

, where $C_i$ is the completion time of job $i$.

2) Minimization of sum of earliness and tardiness, $E + T$ ($b$ = 2):

$$f_2(s){:}\ (E + T) = \textstyle\sum_{i=1}^{n}(E_i + T_i) \tag{2}$$

, where $E_i = max(d_i - C_i, 0)$ denotes the earliness of job $i$ (when job $i$ is completed before its due date), $T_i = max(C_i - d_i, 0)$ indicates the tardiness of job $i$ (when job $i$ is finished after its due date) and $d_i$ stands for the due date for job $i$.

3) Minimization of total weighted tardiness, TWT ($b$ = 3):

$$f_3(s){:}\ \text{TWT} = \textstyle\sum_{i=1}^{n} \omega_i T_i \tag{3}$$

, where $\omega_i$ is the weight of job $i$.

4) Minimization of sum of earliness and tardiness penalties, ET penalties ($b$ = 4):

$$f_4(s){:}\ \text{ET penalties} = \textstyle\sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i) \tag{4}$$

, where $\alpha_i$ is the earliness penalty and $\beta_i$ is the tardiness penalty, where both are in units of RM/time unit.

### B. *Model Constraints*

Given $n$ jobs of $J_i$ ($i = 1, 2, \ldots, n$) to be scheduled on a set of $m$ machines $M_j$ ($j = 1, 2, \ldots, m$) in a given order. Each $J_i$ consists of a sequence of $n_i$ operations. Each operation $O_{ij}$ ($i$ = 1,2,...,$n$; $j$ = 1,2,..., $n_i$) of $J_i$ can be processed on any subset $M_{i,j} \subseteq M_j$ of compatible machines. The constraints of the flexible job shop model are provided as follows:

$$s_{ij} + p_{ij} \le s_{ij'} \qquad \forall\, i, j, j' \tag{5}$$

$$s_{ij} + p_{ij} \le s_{ij'} + L \cdot (1 - y_{ii'j}) \quad \forall\, i, i', j \tag{6}$$

$$s_{ij} + p_{ij} \leq s_{ij'} + L \cdot y_{ii'j} \qquad \forall\, i,\, i',\, j \quad (7)$$

$$s_{ij} \geq r_i \geq 0 \qquad \forall\, i,\, j \quad (8)$$

, where $i, i' = \{1, 2, \dots n\}$, $j, j' = \{1, 2, \dots, m\}$, $s_{ij}$ and $p_{ij}$ are the corresponding start time and processing time of job $i$ on machine $j$, $r_i$ and $d_i$ are the corresponding release time and due date of job $i$, $L$ is a large positive number and $y_{ii'j}$ is a decision variable defined as $y_{ii'j} = 1$ if job $i$ precedes job $i'$ on machine $j$, or 0 otherwise. Eq. (5) is the precedence constraint to ensure that each operation of a job is processed in sequence. Eqs. (6) and (7) are the resource constraints to make sure that each machine can only process one operation at a time. Eq. (8) ensures that no job starts before its release time.

### *C. Model Assumptions*

The main assumptions of the presented model are explained as follows:

1) All jobs are available for processing at time zero.
2) Move times among operations and machine setup times are negligible.
3) Jobs are independent of each other.
4) Machines are independent of each other.
5) A job must not be processed on more than one machine at a time. A machine cannot process more than one job at a time.
6) Every job visits every machine exactly once (no recirculation).
7) Process pre-emption is not permitted. When an operation begins on a machine, it must not be disturbed.
8) Jobs can only be scheduled on machines in specific sequences. An operation of a job must not be processed until its previous operations are finished.
9) There are no limiting resources other than machines/work centres.
10) The machines of different work centres are not identical and perform different operations.

### *D. Objective Functions*

The aim behind production scheduling is to find a good assignment of operations to machines to obtain a schedule which optimizes certain pre-defined objective functions. The objective function most often used in production scheduling problems is the minimization of makespan. Makespan is described as the completion time of the last job leaving the system [1]. It is denoted by $C_{max}$ and can be explained by Eq. (1).

Decision-makers in production lines may be concerned with this objective as makespan characterises a good performance measure in production scheduling; a schedule having minimum makespan is an indication of high machine utilization and high throughput rate (output rate) which are particularly significant in production lines as machines are expensive to acquire and operate. Reducing the makespan is linked to reducing the idle time of machines as well [9].

Though minimizing the makespan is frequently the objective in numerous situations, nevertheless, the common view in production scheduling is that the reduction of production cost is supposed to be the objective of manufacturing optimization as well. Minimizing the makespan would not reduce the scheduling costs; it is established that the minimization of the earliness and tardiness of jobs would be an important objective in many real-world circumstances, specifically concerning the reduction of inventory or holding fees, contractual penalty charges of delayed deliveries and customer goodwill [10]. Hence, the objective of minimizing the production cost has become a critical objective, more than minimizing the makespan.

The earliness-tardiness (ET) objective reflects the JIT principle which emphasizes that earliness and tardiness are undesirable; an ideal schedule is when jobs are completed on their assigned due dates [3]. It aims to eliminate inventories as well as tardy jobs, which directly diminishes inventory holding cost and customer penalties, which conforms to the concept of zero inventory which is widely implemented in the manufacturing environment [11]. The sum of earliness and tardiness is denoted by ($E + T$) and can be formulated as Eq. (2).

The earliness or tardiness is dependent on the due dates of jobs, by which they are normally decided by the customers. Decision-makers can foresee the likelihood of jobs being early, on time or tardy under different settings of due dates: restricted (tight) and unrestricted (loose). They then will be able to find out which jobs can be finished early or on time, and which jobs might be finishing late and by how many days or weeks so that the customers can be informed. Furthermore, if the existing or agreed due date setting does not fulfil the JIT requirements, decision-makers may propose much tighter due date constraints so that jobs can be completed near their designated due dates.

The first two objectives consider every job of the production line as equally important. If every job (customer order) has unequal priority, the jobs will be assigned dissimilar weights. This priority is multiplied by the amount of time the job is tardy. Hence, the objective is to minimize the total weighted tardiness (TWT) in order for more important jobs to be less tardy than the less important ones, as shown by Eq. (3).

The priorities denote the comparative importance of jobs, set by decision-makers depending on the types of customers. Certain customers would be considered more valuable than others; the significance of customers to the company would be subject to several aspects, for instance, the duration of business relationships, the frequency of their orders, the size of their orders with respect to the company's capacity and the potential for future sales [12]. The importance levels differ significantly between every customer and their orders, and the job priorities are manifested in the weights. It is essential for decision-makers to manifest these priorities in their scheduling decisions. A schedule which minimizes the TWT indicates important deliveries are proceeding on schedule, while a TWT value which is higher than necessary indicates a lot of important orders are not delivered as planned.

The ET penalties are also known as JIT-related penalties since they fit the concept of just-in-time deliveries similar to the second objective function in Eq. (2), plus they penalize jobs that are either early or tardy from their promised completion dates. The objective is, thus to minimize the sum of ET penalties to reduce the inventory holding fee plus the contractual penalty charge, as given by Eq. (4).

A job completed before the designated due date could impose additional insurance or storage costs, or even cause product deterioration. Meanwhile, a job completed after the designated due date could incur tardiness penalties due to potential loss of reputation, contractual charges for delayed delivery and customer dissatisfaction [3]. Generally, the consequences of finishing the jobs before their due dates would not necessarily be similar to completing the jobs afterwards; the earliness penalty is considered to be less than the tardiness penalty due to the belief that being tardy is less appropriate than being early [13]. Under the circumstances that certain jobs could not be completed in time, decision-makers need to decide which jobs could be scheduled early or be delayed and at the same time, weigh the expense of deferring some jobs against moving forward the rest.

### *E. Metaheuristic Approach*

There are various metaheuristic approaches reported in the literature for solving the flexible job shop scheduling problem, which include Simulated Annealing [14], Tabu Search [15], Artificial Bee Colony algorithm [16], Ant Colony Optimization [17], Genetic Algorithm [18] and Particle Swarm Optimization (PSO) [19].

This study applied PSO as the metaheuristic technique to solve the flexible job shop scheduling problem as it is capable of finding near-optimal solutions at the expense of very low computational costs and it has performed satisfactorily in an extensive range of applications. PSO has been initially proposed as a metaheuristic optimization technique in the continuous domain [20], nevertheless, it could be adapted to optimize problems in the discrete domain, such as combinatorial optimization problems that involve sequencing or permutation, for instance, the production scheduling problems. This is performed by applying an indirect solution mapping which could encode the position of the particle in the continuous search space into the solution of the discrete problem [21]. The PSO algorithm implemented in this case study applied the solution representation based on the smallest position value (SPV) rule and the random key representation described in [22].

In the PSO algorithm, each potential solution is called a ‘particle’, where each particle with $N$ dimensions corresponds to a permutation sequence consisting of 1 to $N$ operations of a schedule in the flexible job shop. According to [22], every operation of a job in a schedule will be initially ordered in accordance with the continuous position values of the particle which is ranked in ascending order. Each operation is scheduled successively afterwards on each machine in this new sequence by observing the precedence constraints of the flexible job shop. Thus, any permutation of this representation will lead to a viable schedule. During the course of constructing the schedule, every operation is scheduled at the earliest possible starting time such that all constraints are fulfilled, without having to shift operations formerly scheduled.

## III. CASE STUDY

The study considers a flexible job shop system of a manufacturing firm, in which three jobs are required to be scheduled over two to four departments or work centres. Each work centre employs one to four machines for processing, taking into account the precedence constraints. From the data furnished by the manufacturing firm, the authors derived six parameters that are crucial towards designing the flexible job shop model, whereby the lack of at least one parameter will deter the whole process of model development. These parameters are shown in Table I.

Table I. Parameters derived from data obtained from manufacturing firm.

| **No.** | **Parameters** | **Values** |
|---|---|---|
| 1 | Number of jobs, $n$ | 3 jobs |
| 2 | Number of work centres, $c$ | 2 to 4 work centres |
| 3 | Number of machines in use in the work centres, $m$ | 1 to 4 machines |
| 4 | Number of operations within each job | 2 to 4 operations |
| 5 | Processing times to complete each operation for each job on each machine (in minutes), $p_{ij}$ | *-refer to Table II-* |
| 6 | Machining sequence for operations within each job (precedence constraints) | *-refer to Table II-* |

The fifth and sixth parameter values could not be put inside the table since they involved a variety of values that are not possible to be portrayed using the format of the table. Instead, they will be presented in the subsequent table, Table II.

### *A. Flexible Job Shop Model*

Table II then is developed which provides the scheduling details of the flexible job shop model. In Table II, there are three jobs (A, B and C) which need to be scheduled over two to four work centres (M, CM, GS and W). One of the work centres utilizes only one machine (W6), whereas others make use of three to four machines. Each grey shade in Table II represents a set of parallel, identical machines stationed at work centres named CM (1, 3, 6), GS (15, 21, 26), CM (2, 6) and M (5, 19).

Table II. Schedule of flexible job shop model.

| **Job** | **Sequence of machines (processing time in minutes)** | | | |
|---|---|---|---|---|
| **A** | M7 (100) | CM1 (70) | GS15 (60) | W6 (360) |
| | | CM3 (70) | GS21 (60) | |
| | | CM6 (70) | GS26 (60) | |
| **B** | CM2 (120) | M5 (60) | GS15 (120) | |
| | CM6 (120) | M19 (60) | | |
| **C** | M7 (45) | GS16 (60) | | |

From Table II, there are three rows that designate three jobs with processing times $p_{ij}$ in parentheses (fifth parameter values: processing times (in minutes) to complete each operation for each job on each machine). The rows contain permutations of machine IDs denoting the machine order which the jobs go through (sixth parameter values: machining sequence for operations within each job (precedence constraints)). For instance, Job A must be initially scheduled on M7 for 100 minutes; afterwards it needs to be processed at CM work centre, where there is a set of three parallel machines (CM1, CM3 and CM6) with an identical processing time of 70 minutes; later it requires GS work centre, where there is also a set of three parallel machines (GS15, GS21 and GS26) with an identical processing time of 60 minutes; and finally, it is scheduled on W6 for 360 minutes.

Thus, there are alternate machine routings due to the existence of four sets of parallel machines at three work centres (CM, GS and M). For example, at CM work centre, there are three parallel machines with an identical processing time of 70 minutes. Job A from machine M7 can be processed on any of these three machines, namely CM1, CM3 and CM6. Afterwards, Job A could be scheduled on any of the three machines in GS work centre, namely GS15, GS21 and GS26, before finishing at machine W6. Overall, there are four operations that belong to Job A, three operations belong to Job B and two operations belong to Job C.

In terms of machine availability, the flexible job shop model presented thus far assumes all the machines to be available. This includes the four sets of identical and parallel machines, where the jobs may be processed on any available machines and some machines handle different jobs.

### B. *Machine Availability Constraints*

The scheduling detail of the flexible job shop model shown in Table II is modified to consider the machine availability constraints, as shown in Table III. Here, it is assumed that all alternative machines of the parallel sets in CM and GS work centres are unavailable for processing; the only available machines of the parallel sets are the ones where more than one job contends for (machine conflict), namely, CM6 and GS15, respectively. In Table III, the only set of parallel machines left is M5 and M19, which execute Job B only. Job A will then have two sets of machine conflict operations with Job B, which are on CM6 and GS15, as denoted in Table III in yellow and blue colours, respectively. It is noted that there is already a machine conflict on M7 between Job A and Job C, as indicated in green colour. Thus, overall there are three sets of machine conflict operations, including the single machine M7 between operations of Job A and Job C, where M7 is not identical or in parallel with other machines.

Table III. Schedule with machine availability constraints.

| **Job** | **Sequence of machines (processing time in minutes)** | | | |
|---|---|---|---|---|
| **A** | M7 (100) | CM6 (70) | GS15 (60) | W6 (360) |
| **B** | CM6 (120) | M5 (60) | GS15 (120) | |
| | | M19 (60) | | |
| **C** | M7 (45) | GS16 (60) | | |

## IV. RESULTS AND DISCUSSIONS

The initial solution for the case study in this paper is obtained from the original sequence in which the jobs are entered into the system, i.e. the starting order of jobs is based on the sequence in Table II, where Job A will be scheduled first, followed by Job B and lastly Job C.

In generating the due dates, the formula proposed by [23] is adopted which is as follows:

$$d_i = t\sum_{l=1}^{L_i} p_i^l \tag{9}$$

, where $t$ is the control parameter for the due dates, $L_i$ is the number of operations for job $i$ and $p_i^l$ is the $l^{th}$ operation's processing time of job $i$. The setting for $t$, as suggested by [23], is given below, where $t$ = 1.8 indicates the most restrictive or tight and $t$ = 3.5 denotes the least restrictive or loose:

- For cases with tight due date constraints: $t$ = 1.8, 1.9
- For cases with loose due date constraints: $t$ = 2.5, 3.5

This study intends to apply the data provided by the firm to establish a case study from the standpoint of four scenarios that reflect the real-world flexible job shop scheduling problem with the aim to obtain and analyse the best sequence of operations i.e. the best schedule. The four scenarios are based on typical objectives widely analysed and evaluated in flexible job shop environments as performance criteria for sequencing the jobs: 1) makespan, 2) due dates, 3) priorities, and 4) penalties. The four scenarios are subjected to two different cases: Case 1) all machines are assumed to be constantly available for production; therefore, there is no machine unavailability, and Case 2) the alternative machines of the parallel sets are known in advance to be unavailable throughout the entire production period.

The experimental design of the case study is provided in Table IV, while the descriptions of each of the two cases are given as follows:

1) Case 1: All alternative machines are available

For Case 1, all machines are available for processing; for the sets of parallel machines, the jobs may be processed on any available one. Thus, there are a total of three jobs, a combination of four single machines and four sets of parallel machines, as well as nine operations. Based on Table II, at the beginning of the schedule, there is already a machine conflict on M7 between Job A and Job C, i.e. both Job A and Job C are ready to be processed and both require processing on the same machine, M7.

2) Case 2: All alternative machines are unavailable

For Case 2, the authors assumed that the alternative machines of the parallel sets become unavailable, which can be due to breakdowns, busy (occupied), repair or maintenance; the only available machine of the parallel sets is the one in which more than one job contends for (machine conflict). This happens on CM6 and GS15 between Job A and Job B. Thus, the total number of single machines/sets of parallel machines is reduced to six; the number of jobs and operations remains, which are three jobs and nine operations, respectively. It is noted that there is already a machine conflict on M7 between Job A and Job C. Hence, there are three sets of machine conflict operations overall.

Table IV. Experimental design of the case study.

| **Cases**<br>**Scenarios** | **Case 1: All alternative machines are available** | **Case 2: All alternative machines are unavailable** |
|---|---|---|
| Scenario 1: Makespan Analysis | ✓ | ✓ |
| Scenario 2: Earliness and Tardiness (ET) Analysis | ✓ | ✓ |
| Scenario 3: Weighted Tardiness Analysis | ✓ | ✓ |
| Scenario 4: ET Penalties Analysis | ✓ | ✓ |

After each of the best schedules is obtained and analysed, the perspectives of each of the findings to decision-makers are discussed in an effort to aid them in making informed decisions and taking necessary actions to achieve high machine utilization, cost reduction and customer satisfaction.

## *A. Scenario 1: Makespan Analysis*

The first scenario is the makespan analysis. In this scenario, the objective is to obtain a minimum makespan, i.e. minimum completion time of the last job leaving the system.

1) Case 1: All alternative machines are available

Objective: Minimum makespan, $C_{max}$

As shown in Table II previously, Job A is on the critical path, which takes the longest time to complete. Since makespan is the completion time of the last job leaving the system, in this case, the makespan is equal to the total operation times for Job A, and it cannot be minimized further: $C_{max} = 100 + 70 + 60 + 360 = 590$ minutes. Other jobs do not affect the value of makespan since they have shorter processing times.

The scheduling solution, i.e. the best sequence of operations can be visually represented in the form of a Gantt chart that displays the operation processing by each machine (listed in the vertical axis) over a period of time (horizontal axis). In this case, the best sequence of operations is shown in the Gantt chart of the best schedule provided in Fig. 1.

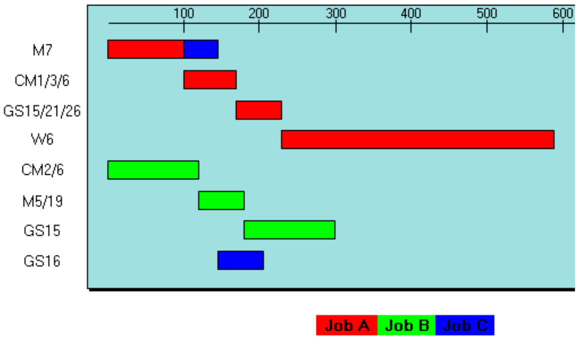


Fig. 1. Gantt chart of best schedule for minimum makespan (Case 1).

There is a machine conflict on M7 between Job A and Job C, i.e. both Job A and Job C are ready to be processed and both require processing on the same machine, M7. Since the starting order of jobs is based on the sequence in Table II, where Job A will be scheduled first and finally Job C, any delay in Job A may lead to Job C being processed on M7 first instead. Job A will then need to queue behind Job C when it is time for Job A to resume processing on M7. This situation will result in the increase of makespan value, since the makespan will then equal the total operation times for Job A plus the total operation times for Job C on M7.

Therefore, the makespan value if there is any delay in processing Job A on M7: $C_{max}$ = Total operation times for Job A + total operation times for Job C on M7 = $(100 + 70 + 60 + 360) + 45 = 635$ minutes.

Decision-makers may wish to take into serious consideration this increase in makespan value if they ever need to suspend Job A under any circumstances, since it will have a direct impact on throughput rate and machine utilization; a schedule having a high makespan value indicates a low throughput rate (output rate) and low machine utilization.

2) Case 2: All alternative machines are unavailable

Objective: Minimum makespan, $C_{max}$

As shown in Table III previously, Job A is on the critical path, which takes the longest time to complete. As stated in Case 1 earlier, since makespan is the completion time of the last job leaving the system, in this case, the makespan then equals the total operation times for Job A, and it cannot be minimized further. Other jobs do not affect the value of makespan since they have shorter processing times. The best sequence of operations is shown in the Gantt chart of the best schedule provided in Fig. 2.

There is already a machine conflict on M7 between Job A and Job C, which is similar to Case 1, i.e. both Job A and Job C are ready to be processed and both require processing on the same machine, M7. In Case 2, another two machine conflicts happen on CM6 and

GS15 between Job A and Job B. Thus, there are a total of three operations of Job A that are in conflict with either Job B or Job C. Since the starting order of jobs is based on the sequence in Table III, where Job A will be scheduled first and finally Job C, any delay in Job A may lead to Job C to be processed on M7 first instead. Job A will then need to queue behind Job C when it is time for Job A to resume processing on M7. The same situation will happen between Job A and Job B, i.e. any delay in Job A may lead to Job B to be processed on CM6 and GS15 first. Job A will need to queue behind Job B when it is time for Job A to resume processing on CM6 and GS15 later. This situation will result in an increase in makespan value.

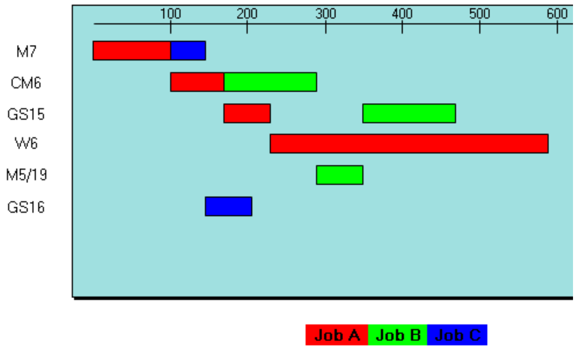


Fig. 2. Gantt chart of best schedule for minimum makespan (Case 2).

As similarly stated in Case 1 earlier, decision-makers may wish to take into serious consideration this increase in makespan value if they ever need to suspend Job A under any circumstances, since it will have a direct impact on throughput rate and machine utilization; a schedule having a high makespan value indicates low throughput rate (output rate) and low machine utilization. This is particularly true in Case 2 since there are now three operations of Job A that are in conflict with Job B and Job C. If Job A needs to be suspended, its three operations may need to queue behind Job B and Job C when it is time to resume processing. This will for certain raise the makespan value.

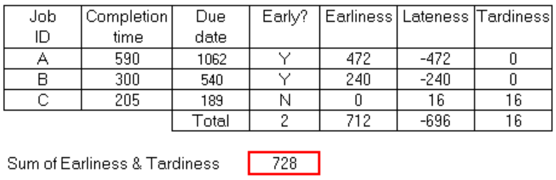
### B. *Scenario 2: Earliness and Tardiness (ET) Analysis*

The second scenario is the ET analysis. In this scenario, the objective is to achieve a minimum sum of ET, $(E + T)_{\min}$. As stated earlier, there are four different settings of due dates, depending on the control parameter, $t$ ($t = 1.8, 1.9, 2.5, 3.5$), which can be grouped into either tight or loose due date constraints. The looser the due dates are, the higher the values of the sum of ET.

1) Case 1: All alternative machines are available

Objective: Minimum sum of ET, $(E + T)_{\min}$

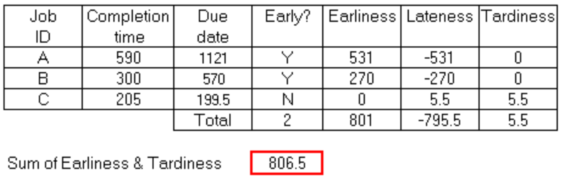
The goal to be achieved in this scenario is for jobs to finish closer to their due dates, i.e. in line with the JIT principle of making a product precisely when it is needed. Nevertheless, from the computational results of $(E + T)_{\min}$ shown in Fig. 3, even under tight due date constraints ($t = 1.8, 1.9$), the sum of ET are quite high. This means jobs are not finishing nearer to their due dates; they either finish processing too early or too tardy. This may be due to certain jobs like Job B, which does not have machine conflict, can straightaway be scheduled without delay, and thus it is able to finish way ahead of its due date. In addition, this also means that the strict due date settings applied are not tight enough in this case such that Job A and Job B manage to finish way ahead of their due dates, resulting in higher values of earliness.



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 590 | 1062 | Y | 472 | -472 | 0 |
| B | 300 | 540 | Y | 240 | -240 | 0 |
| C | 205 | 189 | N | 0 | 16 | 16 |
| | | Total | 2 | 712 | -696 | 16 |

Sum of Earliness & Tardiness 728

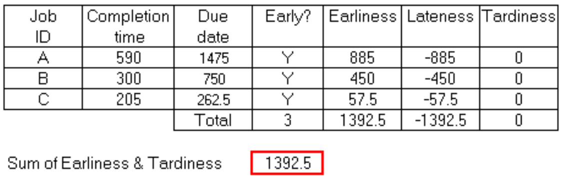
(a) $t = 1.8$



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 590 | 1121 | Y | 531 | -531 | 0 |
| B | 300 | 570 | Y | 270 | -270 | 0 |
| C | 205 | 199.5 | N | 0 | 5.5 | 5.5 |
| | | Total | 2 | 801 | -795.5 | 5.5 |

Sum of Earliness & Tardiness 806.5

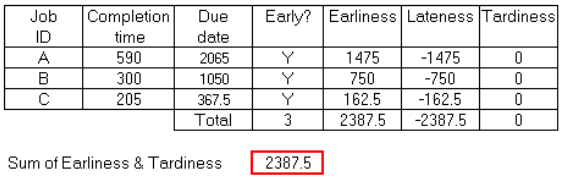
(b) $t = 1.9$



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 590 | 1475 | Y | 885 | -885 | 0 |
| B | 300 | 750 | Y | 450 | -450 | 0 |
| C | 205 | 262.5 | Y | 57.5 | -57.5 | 0 |
| | | Total | 3 | 1392.5 | -1392.5 | 0 |

Sum of Earliness & Tardiness 1392.5

(c) $t = 2.5$



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 590 | 2065 | Y | 1475 | -1475 | 0 |
| B | 300 | 1050 | Y | 750 | -750 | 0 |
| C | 205 | 367.5 | Y | 162.5 | -162.5 | 0 |
| | | Total | 3 | 2387.5 | -2387.5 | 0 |

Sum of Earliness & Tardiness 2387.5

(d) $t = 3.5$

Fig. 3. Computational results of $(E + T)_{\min}$ (Case 1).

Under tight due date constraints ($t = 1.8, 1.9$), Job C will never finish on time and always be tardy, compared to Job A and Job B that will always be finishing ahead of their due dates. This is because jobs' due date computations are in terms of their total processing times. The shorter the total processing times, the shorter their due dates will be. Besides, Job C has a machine conflict on M7 with Job A.

All four settings of the control parameter, $t$ result in a similar sequence of operations since the sum of ET are proportional to the due dates. Thus, the best sequence of operations is shown in the Gantt chart of the best schedule provided in Fig. 4.

Decision-makers will be able to notify the customers about the completion dates of their orders, especially if their orders will be running late. On the other hand, by customer request, decision-makers may offer other due date settings that are more loose or less restricted, so that the customer orders may be finished early or on time, rather than late. Alternatively, in order to closely observe the JIT requisite, decision-makers may as well suggest much tighter due dates (e.g. $t$ = 1.8), if for instance, the existing due-date setting (e.g. $t$ = 1.9) results in a high amount of inventories.

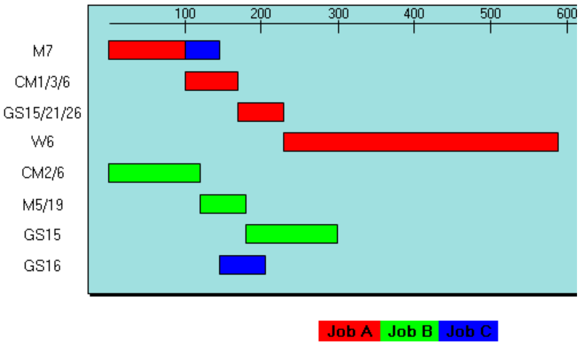
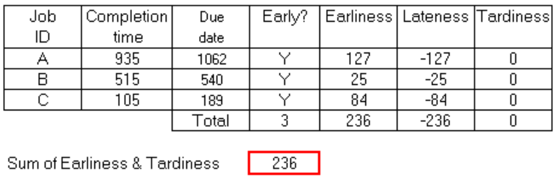


Fig. 4. Gantt chart of best schedule for $(E + T)_{\min}$ (Case 1).

2) Case 2: All alternative machines are unavailable

Objective: Minimum sum of ET, $(E + T)_{\min}$

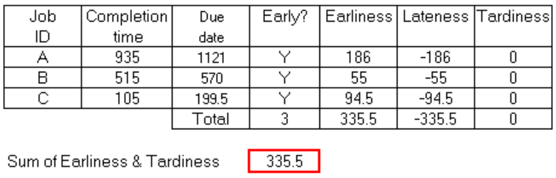
From the computational results of $(E + T)_{\min}$ shown in Fig. 5, all jobs manage to finish early with no tardy jobs. The sum of ET are much lesser for all four settings, contrary to Case 1; part of it is because only earliness values are left. Another reason is the existence of the three sets of machine conflict operations that cause certain jobs to be delayed until they are scheduled much nearer to their due dates. In fact, under tight due date constraints of $t$ = 1.8, they manage to finish closer to their due dates, compared to when there is no machine unavailability. The loose and tight due date settings result in a dissimilar sequence of operations.



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 935 | 1062 | Y | 127 | -127 | 0 |
| B | 515 | 540 | Y | 25 | -25 | 0 |
| C | 105 | 189 | Y | 84 | -84 | 0 |
| | | Total | 3 | 236 | -236 | 0 |

Sum of Earliness & Tardiness 236

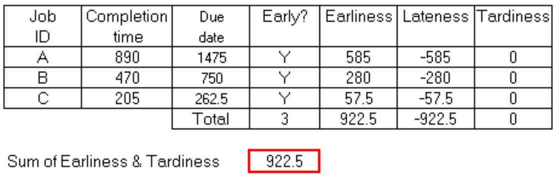
(a) $t$ = 1.8



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 935 | 1121 | Y | 186 | -186 | 0 |
| B | 515 | 570 | Y | 55 | -55 | 0 |
| C | 105 | 199.5 | Y | 94.5 | -94.5 | 0 |
| | | Total | 3 | 335.5 | -335.5 | 0 |

Sum of Earliness & Tardiness 335.5

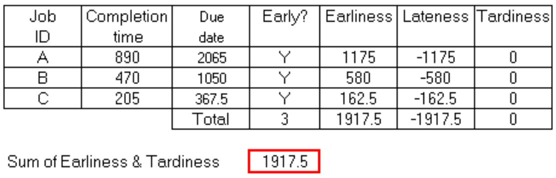
(b) $t$ = 1.9



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 890 | 1475 | Y | 585 | -585 | 0 |
| B | 470 | 750 | Y | 280 | -280 | 0 |
| C | 205 | 262.5 | Y | 57.5 | -57.5 | 0 |
| | | Total | 3 | 922.5 | -922.5 | 0 |

Sum of Earliness & Tardiness 922.5

(c) $t$ = 2.5



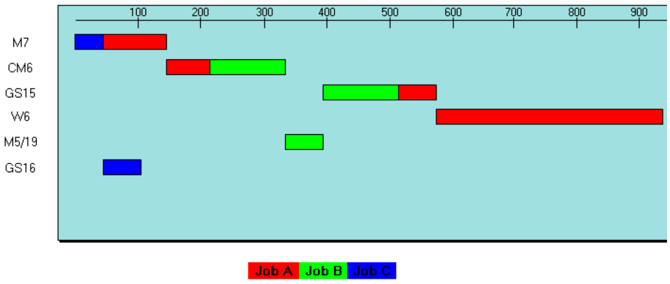
| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness |
|---|---|---|---|---|---|---|
| A | 890 | 2065 | Y | 1175 | -1175 | 0 |
| B | 470 | 1050 | Y | 580 | -580 | 0 |
| C | 205 | 367.5 | Y | 162.5 | -162.5 | 0 |
| | | Total | 3 | 1917.5 | -1917.5 | 0 |

Sum of Earliness & Tardiness 1917.5

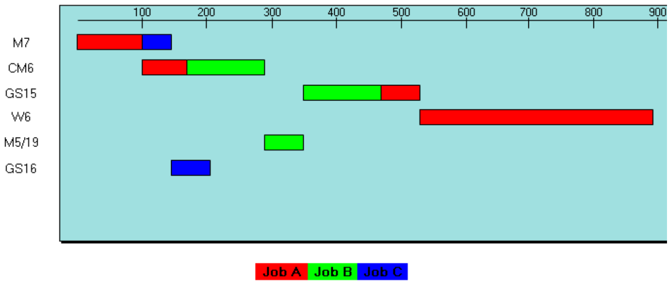
(d) $t$ = 3.5

Fig. 5. Computational results of $(E + T)_{\min}$ (Case 2).

The best sequence of operations are shown in the Gantt chart of the best schedules provided in Fig. 6.



(a) $t$ = 1.8 and $t$ = 1.9



(b) $t$ = 2.5 and $t$ = 3.5

Fig. 6. Gantt chart of best schedules for $(E + T)_{\min}$ (Case 2).

Decision-makers will be able to notify the customers regarding the completion dates of their orders. Alternatively, since the goal is to realize the JIT concept, decision-makers may investigate the settings of due-date constraints that result in jobs being completed nearest to their promised delivery dates; if for instance, the established setting results in a high amount of inventories.

### *C. Scenario 3: Weighted Tardiness Analysis*

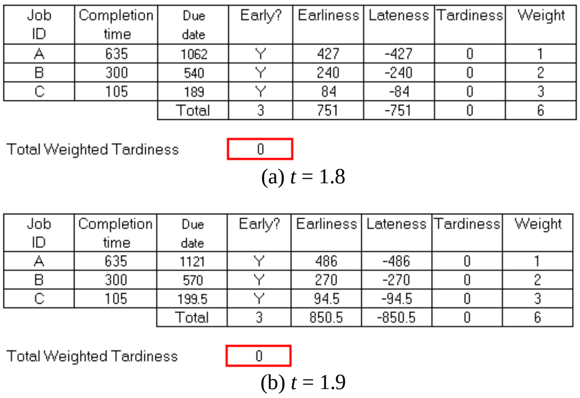
The third scenario is the weighted tardiness analysis. The objective in this scenario is to achieve minimum TWT. To demonstrate the way the priority weight operates, the weights for Job A, B and C are assumed to be 1, 2 and 3, respectively. Job C has the highest importance or priority, followed by Job B and Job A.

1) Case 1: All alternative machines are available

Objective: Minimum TWT

As can be seen from Case 1 in Scenario 2, under tight due date constraints (t = 1.8, 1.9), only Job C never finishes on time and is always tardy, compared to Job A and Job B that always manage to finish ahead of their due dates.

From the computational results of minimum TWT shown in Fig. 7, when Job C is given the highest priority under tight due date constraints, it produces the sequence of operations which prioritizes Job C to be completed first instead of Job A, since both jobs have machine conflict on M7. This occurs so as to reduce the TWT. The best sequence of operations is shown in the Gantt chart of the best schedule provided in Fig. 8.



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness | Weight |
|---|---|---|---|---|---|---|---|
| A | 635 | 1062 | Y | 427 | -427 | 0 | 1 |
| B | 300 | 540 | Y | 240 | -240 | 0 | 2 |
| C | 105 | 189 | Y | 84 | -84 | 0 | 3 |
| | | Total | 3 | 751 | -751 | 0 | 6 |

Total Weighted Tardiness 0

(a) $t$ = 1.8

| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness | Weight |
|---|---|---|---|---|---|---|---|
| A | 635 | 1121 | Y | 486 | -486 | 0 | 1 |
| B | 300 | 570 | Y | 270 | -270 | 0 | 2 |
| C | 105 | 199.5 | Y | 94.5 | -94.5 | 0 | 3 |
| | | Total | 3 | 850.5 | -850.5 | 0 | 6 |

Total Weighted Tardiness 0

(b) $t$ = 1.9

Fig. 7. Computational results of minimum TWT (Case 1).

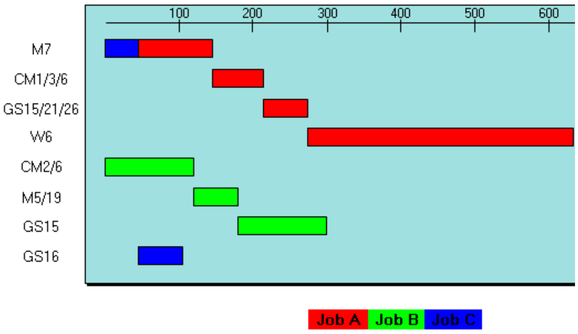


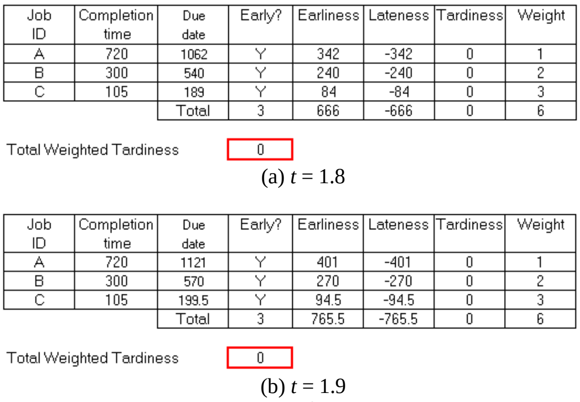
Fig. 8. Gantt chart of best schedule for minimum TWT (Case 1).

With no priority weights, the expected sequencing of jobs are such that Job A will be processed first along with Job B, and finally followed by Job C, as in Case 1 of Scenario 2. Thus, by applying the TWT objective function, decision-makers are capable of prioritizing specific jobs in order for the critical jobs to be delivered as scheduled while the less critical ones are delayed to some acceptable extent.

2) Case 2: All alternative machines are unavailable

Objective: Minimum TWT

From the computational results of minimum TWT shown in Fig. 9, when Job C is assigned the highest priority under tight due date constraints, it produces the sequence of operations which prioritizes Job C to be completed first instead of Job A, since both jobs have machine conflict on M7. This occurs so as to reduce the TWT. This is true for Job B as well since it has the second highest priority and both Job B and A have machine conflict on CM6 and GS15. The best sequence of operations is shown in the Gantt chart of the best schedule provided in Fig. 10.



| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness | Weight |
|---|---|---|---|---|---|---|---|
| A | 720 | 1062 | Y | 342 | -342 | 0 | 1 |
| B | 300 | 540 | Y | 240 | -240 | 0 | 2 |
| C | 105 | 189 | Y | 84 | -84 | 0 | 3 |
| | | Total | 3 | 666 | -666 | 0 | 6 |

Total Weighted Tardiness 0

(a) $t$ = 1.8

| Job ID | Completion time | Due date | Early? | Earliness | Lateness | Tardiness | Weight |
|---|---|---|---|---|---|---|---|
| A | 720 | 1121 | Y | 401 | -401 | 0 | 1 |
| B | 300 | 570 | Y | 270 | -270 | 0 | 2 |
| C | 105 | 199.5 | Y | 94.5 | -94.5 | 0 | 3 |
| | | Total | 3 | 765.5 | -765.5 | 0 | 6 |

Total Weighted Tardiness 0

(b) $t$ = 1.9

Fig. 9. Computational results of minimum TWT (Case 2).

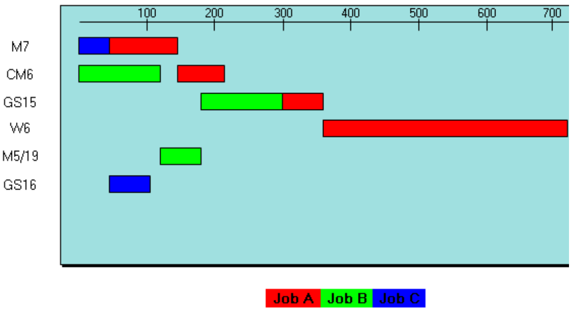


Fig. 10. Gantt chart of best schedule for minimum TWT (Case 2).

As similarly stated in Case 1 earlier, without the priority weights, the ordering or sequencing of jobs are such that Job A may be processed first along with Job B, and finally followed by Job C. Thus, by applying the TWT objective function, decision-makers are capable to prioritize specific jobs in order for the critical jobs to be delivered on time while the less critical ones are postponed to some reasonable extent.

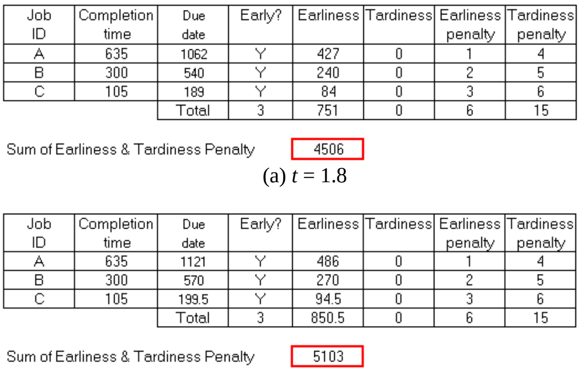
### *D. Scenario 4: ET Penalties Analysis*

The fourth scenario is the analysis of ET penalties. In this scenario, the objective is to minimize the sum of ET penalties. To illustrate the way the penalty cost operates, the earliness penalties are given as 1, 2 and 3 and tardiness penalties are given as 4, 5 and 6 to Job A, B and C, respectively (in units of RM/time unit). The earliness penalty is considered to be less than the tardiness penalty due to the belief that being tardy is less appropriate than being early.

1) Case 1: All alternative machines are available

Objective: Minimum sum of ET penalties

As can be seen from Case 1 in Scenario 2, under tight due date constraints ($t$ = 1.8, 1.9), only Job C never finishes on time and is always tardy, compared to Job A and Job B that always manage to finish ahead of their due dates. From the computational results of the minimum sum of ET penalties shown in Fig. 11, when Job C is given the highest ET penalties under tight due date constraints, it produces the sequence of operations which schedules Job C to be completed first instead of Job A, since both of the jobs have machine conflict on M7. This occurs so as to reduce the sum of ET penalties. The best sequence of operations is shown in the Gantt chart of the best schedule provided in Fig. 12.



| Job ID | Completion time | Due date | Early? | Earliness | Tardiness | Earliness penalty | Tardiness penalty |
|---|---|---|---|---|---|---|---|
| A | 635 | 1062 | Y | 427 | 0 | 1 | 4 |
| B | 300 | 540 | Y | 240 | 0 | 2 | 5 |
| C | 105 | 189 | Y | 84 | 0 | 3 | 6 |
|  |  | Total | 3 | 751 | 0 | 6 | 15 |

Sum of Earliness & Tardiness Penalty 4506

(a) $t$ = 1.8

| Job ID | Completion time | Due date | Early? | Earliness | Tardiness | Earliness penalty | Tardiness penalty |
|---|---|---|---|---|---|---|---|
| A | 635 | 1121 | Y | 486 | 0 | 1 | 4 |
| B | 300 | 570 | Y | 270 | 0 | 2 | 5 |
| C | 105 | 199.5 | Y | 94.5 | 0 | 3 | 6 |
|  |  | Total | 3 | 850.5 | 0 | 6 | 15 |

Sum of Earliness & Tardiness Penalty 5103

(b) $t$ = 1.9

Fig. 11. Computational results of minimum sum of ET penalties (Case 1).

Without the ET penalties, the expected sequencing of jobs are such that Job A will be processed first along with Job B, and finally followed by Job C as in Case 1 of Scenario 2. Hence, decision-makers are capable to deliberate the expense of deferring certain jobs as compared to moving the other jobs ahead, in the event that certain jobs could not be completed as scheduled.

It is vital to comprehensively assess this circumstance as it has a clear effect on the tardiness cost (contractual penalty charge, etc.) and the earliness cost (inventory holding fee, etc.).

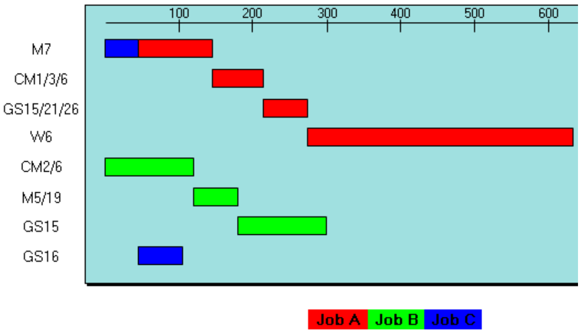
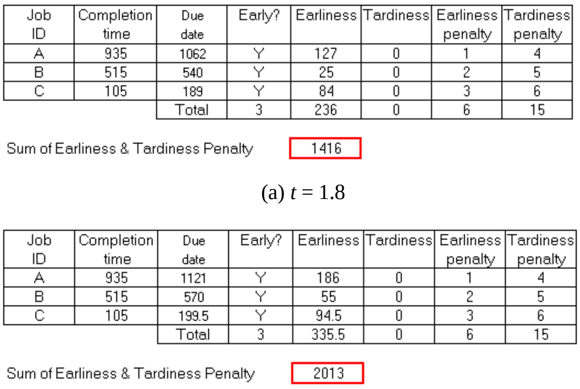


Fig. 12. Gantt chart of best schedule for minimum sum of ET penalties (Case 1).

2) Case 2: All alternative machines are unavailable

Objective: Minimum sum of ET penalties

From the computational results of the minimum sum of ET penalties shown in Fig. 13, when Job C is assigned the highest ET penalties under tight due date constraints, it produces the sequence of operations that schedules Job C to be completed first instead of Job A, since both of the jobs have machine conflict on M7. This occurs so as to reduce the sum of ET penalties. This is true for Job B as well since it has the second-highest ET penalties, and both Job B and A have machine conflicts on CM6 and GS15. The best sequence of operations is shown in the Gantt chart of the best schedule provided in Fig. 14.



| Job ID | Completion time | Due date | Early? | Earliness | Tardiness | Earliness penalty | Tardiness penalty |
|---|---|---|---|---|---|---|---|
| A | 935 | 1062 | Y | 127 | 0 | 1 | 4 |
| B | 515 | 540 | Y | 25 | 0 | 2 | 5 |
| C | 105 | 189 | Y | 84 | 0 | 3 | 6 |
|  |  | Total | 3 | 236 | 0 | 6 | 15 |

Sum of Earliness & Tardiness Penalty 1416

(a) $t$ = 1.8

| Job ID | Completion time | Due date | Early? | Earliness | Tardiness | Earliness penalty | Tardiness penalty |
|---|---|---|---|---|---|---|---|
| A | 935 | 1121 | Y | 186 | 0 | 1 | 4 |
| B | 515 | 570 | Y | 55 | 0 | 2 | 5 |
| C | 105 | 199.5 | Y | 94.5 | 0 | 3 | 6 |
|  |  | Total | 3 | 335.5 | 0 | 6 | 15 |

Sum of Earliness & Tardiness Penalty 2013

(b) $t$ = 1.9

Fig. 13. Computational results of minimum sum of ET penalties (Case 2).

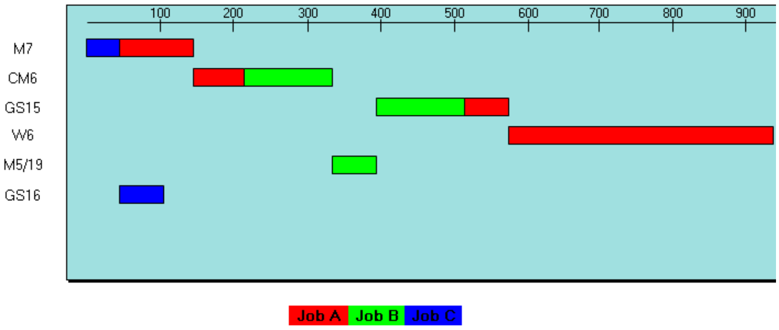


Fig. 14. Gantt chart of best schedule for minimum sum of ET penalties (Case 2).

As similarly stated in Case 1 earlier, without the ET penalties, the expected sequencing of jobs are such that Job A may be processed first along with Job B, and finally followed by Job C. Decision-makers are capable to deliberate the expense of deferring certain jobs as compared to moving the other jobs ahead, in the event that certain jobs could not be completed as scheduled. It is vital to comprehensively assess this circumstance as it has a clear effect on the tardiness cost (contractual penalty charge, etc.) and the earliness cost (inventory holding fee, etc.).

### *E. Summary of Findings*

According to the results obtained from Case 1 and Case 2 under the four different scenarios, both Scenarios 1 and 3 achieve the same solution for both cases, i.e. 590 and 0, respectively. This is due to the nature of the performance measures since, for the makespan-oriented objective, it is subject to the length of processing times of jobs. A job having the longest processing time will always be on the critical path, which takes the longest time to complete. Since, in both cases, the processing times do not change, Job A, which has the longest processing time, is the last job leaving the system in both cases, which contributes to the consistent value of minimum makespan in Case 1 and Case 2. For the TWT objective, it is subject to the weights assigned to every job. Since the starting order of jobs is based on the sequence in Table II, where Job A will be scheduled initially and finally Job C, without the weights, Job C is processed last. Moreover, the tight due date setting will cause Job C not finishing on time and always be tardy. But, when Job C is assigned the highest priority, it produces the sequence of operations which prioritizes Job C to be completed first. Since, in both cases, Job C is assigned the highest priority, Job C will be completed first in both cases, which contributes to the consistent minimum TWT in Case 1 and Case 2.

On another hand, for Scenarios 2 and 4, Case 2 achieves better solutions. Once again, this is because of the nature of the performance measures since, for earliness-tardiness objectives, they depend on the due dates and tightness levels ($t$ = 1.8, 1.9, 2.5, 3.5). The closer the completion time of jobs from the due dates, the smaller the ET values will be. On the other hand, the lower the tightness level, the higher the ET values will be. However, based on individual results of Scenarios 2 and 4, the strict due date settings ($t$ = 1.8, 1.9) applied are not tight enough such that certain jobs manage to finish way ahead of their due dates. Therefore, in Case 1, with all alternative machines are available, the jobs may be scheduled on any available ones, such that they can straightaway be scheduled without delay, thus they are able to finish way ahead of their due dates, resulting in higher values of earliness. While in Case 2, when there are machine conflicts that lead to certain jobs having to be scheduled on the same machine, selected jobs have to be processed later on, which means much closer to the due dates, resulting in smaller values of earliness. Hence, besides due dates and tightness levels, the machine conflicts also affect the result of Scenarios 2 and 4, i.e. the machine conflicts force certain jobs to be processed much nearer to their due dates, resulting in smaller values of earliness.

## V. CONCLUSION

In this research, a case study of a real-life flexible job shop comprises of three jobs to be scheduled over a maximum of four work centres with machine availability constraints is investigated. The best sequence of operations i.e. the best schedules that correspond to makespan, due dates, priorities and penalties, whereby they were subjected to cases of machine availability and unavailability, have been determined. The best schedules have been analysed and interpreted to provide helpful insight to decision-makers to aid them in making informed decisions and taking necessary actions from the standpoint of four scenarios to achieve high machine utilization, cost reduction and customer satisfaction. Based on the outcome of this research, the machine availability constraints that exist in this case study have no impact on some of the performance criteria and a positive effect on the others. Nevertheless, the above conclusion is subject to the shop configuration in the firm and the variables assumed in this setting, such as the due date settings, priority and penalty values. Different configurations and variable settings may result in dissimilar outcomes. One of the future research opportunities involves the testing of the proposed approach in solving larger flexible job shop configurations, i.e. consisting of a higher number of work centres, machines, jobs or operations. Moreover, different variable settings, such as different due date constraints, priority and penalty values can be used to model the manufacturing systems. It would also be worthwhile to consider adding into the model considerations such as non-identical release times of jobs and machine setup times (which may be sequence-dependent).

## REFERENCES

[1] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 5th ed. Springer, 2016.

[2] C. Laroque, M. Leißau, P. Copado, C. Schumacher and J. Panadero, “A Biased-Randomized Discrete-Event Algorithm for the Hybrid Flow Shop Problem with Time Dependencies and Priority Constraints,” *Algorithms*, vol. 15, no. 2. pp. 1–14, 2022.

[3] J. Rezaeian, S. Hosseini-Kia and I. Mahdavi, “The Preemptive Just-in-Time Scheduling Problem in a Flow Shop Scheduling System,” *J. Optim. Ind. Eng.*, vol. 12, no. 2, pp. 79–92, 2019.

[4] J. Shen, Y. Shi, J. Shi, Y. Dai and W. Li, “An Uncertain Permutation Flow Shop Predictive Scheduling Problem with Processing Interruption,” *Phys. A Stat. Mech. Its Appl.*, vol. 611, pp. 1–15, 2023.

[5] M. M. Wocker, F. F. Ostermeier, T. Wanninger, R. Zwinkau and J. Deuse, “Flexible Job Shop Scheduling with Preventive Maintenance Consideration,” *J. Intell. Manuf.*, 2023.

[6] D. Lei, “A Pareto Archive Particle Swarm Optimization for Multi-Objective Job Shop Scheduling,” *Comput. Ind. Eng.*, vol. 54, no. 4, pp. 960–971, 2008.

[7] M. Geurtsen, J. B. H. C. Didden, J. Adan, Z. Atan and I. Adan, “Production, Maintenance and Resource Scheduling: A Review,” *Eur. J. Oper. Res.*, vol. 305, no. 2, pp. 501–529, 2023.

[8] T. Xia, Y. Ding, Y. Dong, Z. Chen, M. Zheng, E. Pan and L. Xi, “Collaborative Production and Predictive Maintenance

Scheduling for Flexible Flow Shop with Stochastic Interruptions and Monitoring Data," *J. Manuf. Syst.*, vol. 65, pp. 640–652, 2022.

[9] R. Bürgy and K. Bülbül, "The Job Shop Scheduling Problem with Convex Costs," *Eur. J. Oper. Res.*, vol. 268, no. 1, pp. 82–100, 2018.

[10] J. G. Kim, H. B. Jun, J. Y. Bang, J. H. Shin and S. H. Choi, "Minimizing Tardiness Penalty Costs in Job Shop Scheduling under Maximum Allowable Tardiness," *Processes*, vol. 8, no. 11, pp. 1–15, 2020.

[11] D. Guo, Z. Lyu, W. Wu, R. Y. Zhong, Y. Rong, G. Q. Huang, "Synchronization of Production and Delivery with Time Windows in Fixed-Position Assembly Islands under Graduation Intelligent Manufacturing System," *Robot. Comput. Integr. Manuf.*, vol. 73, pp. 1–14, 2022.

[12] H. Güçdemir and H. Selim, "Dynamic Dispatching Priority Setting in Customer-Oriented Manufacturing Environments," *Int. J. Adv. Manuf. Technol.*, vol. 92, no. 5–8, pp. 1861–1874, 2017.

[13] S. Shi, H. Xiong and G. Li, "A No-Tardiness Job Shop Scheduling Problem with Overtime Consideration and the Solution Approaches," *Comput. Ind. Eng.*, vol. 178, pp. 1–16, 2023.

[14] F. M. Defersha, D. Obimuyiwa and A. D. Yimer, "Mathematical Model and Simulated Annealing Algorithm for Setup Operator Constrained Flexible Job Shop Scheduling Problem," *Comput. Ind. Eng.*, vol. 171, pp. 1–22, 2022.

[15] M. Hajibabaei and J. Behnamian, "Flexible Job-Shop Scheduling Problem with Unrelated Parallel Machines and Resources-Dependent Processing Times: A Tabu Search Algorithm," *Int. J. Manag. Sci. Eng. Manag.*, vol. 16, no. 4, pp. 242–253, 2021.

[16] Y. Li, C. Liao, L. Wang, Y. Xiao, Y. Cao and S. Guo, "A Reinforcement Learning-Artificial Bee Colony Algorithm for Flexible Job-Shop Scheduling Problem with Lot Streaming," *Appl. Soft Comput.*, vol. 146, pp. 1–13, 2023.

[17] M. Huang, D. Guo and F. Guo, "An Improved Ant Colony Algorithm for Multi-Objective Flexible Job-Shop Scheduling Problem," in *2022 IEEE 10th Int. Conf. Comp. Sci. and Netw. Technol.*, pp. 1–3, 2022.

[18] X. Luo, Q. Qian and Y. F. Fu, "Improved Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem," *Procedia Comput. Sci.*, vol. 166, pp. 480–485, 2020.

[19] Y. J. Gao, Q. X. Shang, Y. Y. Yang, R. Hu and B. Qian, "Improved Particle Swarm Optimization Algorithm Combined with Reinforcement Learning for Solving Flexible Job Shop Scheduling Problem," in *Adv. Intell. Comput. Technol. and Appl.*, D. S. Huang, P. Premaratne, B. Jin, B. Qu, K. H. Jo and A. Hussain, Eds., Singapore: Springer, pp. 288–298, 2023.

[20] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. 1995 IEEE Int. Conf. Neur. Netw.*, pp. 1942–1948, 1995.

[21] M. A. Zeidan, M. A. Al-Abaji and M. H. Ahmed, "Improved Particle Swarm Algorithm for Permutation Flow Shop Scheduling Problems," *Investig. Operacional*, vol. 42, no. 2, pp. 165–203, 2021.

[22] M. F. Tasgetiren, Y. C. Liang, M. Sevkli and G. Gencyilmaz, "A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in the Permutation Flowshop Sequencing Problem," *Eur. J. Oper. Res.*, vol. 177, no. 3, pp. 1930–1947, 2007.

[23] H. Ohta and T. Nakatani, "A Heuristic Job-Shop Scheduling Algorithm to Minimize the Total Holding Cost of Completed and in-Process Products Subject to No Tardy Jobs," *Int. J. Prod. Econ.*, vol. 101, no. 1, pp. 19–29, 2006.