OmniaScience

Journal of Industrial Engineering and Management
JIEM, 2020 – 13(3): 546-564 – Online ISSN: 2013-0953 – Print ISSN: 2013-8423
https://doi.org/10.3926/jiem.3153

A Flow Shop Batch Scheduling and Operator Assignment Model with Time-Changing Effects of Learning and Forgetting to Minimize Total Actual Flow Time

Dwi Kurniawan, Andi Cakravastia Raja, Suprayogi Suprayogi, Abdul Hakim Halim

Institut Teknologi Bandung (Indonesia)

dwikur77@students.itb.ac.id, andi@mail.ti.itb.ac.id, yogi@mail.ti.itb.ac.id, ahakimhalim@mail.ti.itb.ac.id

Received: March 2020
Accepted: July 2020

Abstract:

Purpose: This paper aims to investigate simultaneous problems of batch scheduling and operator assignment with time-changing effects caused by learning and forgetting.

Design/methodology/approach: A mathematical model was developed for the problems, and the decision variables of the model were operator assignment, the number of batches, batch sizes and the schedule of the resulting batches. A proposed algorithm worked by trying different number of batches, starting from one, and increasing the number of batches one by one until the objective function value did not improve anymore.

Findings: We mathematically and numerically show that the closest batch to the due date always became the largest batch in the schedule, and the faster operators learn, the larger the difference between the closest batch to the due date and the other batches, the lower optimal number of batches, and the lower the total actual flow time.

Originality/value: Previous papers have considered the existence of alternative operators but have not considered learning and forgetting, or have considered learning and forgetting but only in a single-stage system and without considering alternative operators.

Keywords: batch scheduling, operator assignment, time-changing effect, learning-forgetting, flow shop, actual flow time

To cite this article:

Kurniawan, D., Raja, A.C., Suprayogi, S., & Halim, A.H. (2020). A flow shop batch scheduling and operator assignment model with time-changing effects of learning and forgetting to minimize total actual flow time. *Journal of Industrial Engineering and Management*, 13(3), 546-564. https://doi.org/10.3926/jiem.3153

1. Introduction

Scheduling is an activity of assigning a set of resources (machines, vehicles or people) to perform a number of tasks over time in order to achieve one or more objectives while satisfying a set of constraints (Baker, 1974). Some objectives frequently used in scheduling are flow time, lateness, tardiness and the number of tardy jobs (Bedworth & Bailey, 1987). These measures can satisfy either the need to reduce inventory level or the need to meet due date; however, they cannot do both simultaneously. To reduce inventory level and to meet due date simultaneously, the so called actual flow time defined in Halim and Ohta (1993) as the interval between starting time of production and due date was proposed. Using the actual flow time as the performance measure, parts may arrive at the shop floor when they are needed instead of arriving at time zero simultaneously, and parts will be delivered to the customer at their due date (Halim & Ohta, 1993). Batch scheduling models to minimize actual flow time have been studied such as in Halim, Miyazaki and Ohta (1994) and Yusriski, Sukoyo, Samadhi and Halim (2015).

The batch scheduling model in Halim et al. (1994) assumed fixed processing times. In practical situations, processing times can vary, such as from the existence of alternative operators studied in Costa, Cappadonna and Fichera (2013) and from time-changing effects of learning and forgetting studied in Jaber and Bonney (1996) and Yusriski et al. (2015). Nevertheless, Costa et al. (2013) and Yusriski et al. (2015) only considered single-machine systems, while Jaber and Bonney (1996) have not considered batch scheduling.

This paper extends the batch scheduling model in Halim et al. (1994) by considering the existence of alternative operators to perform each operation proposed in Costa et al. (2013) and time-changing effects of learning and forgetting studied in Jaber and Bonney (1996) and Yusriski et al. (2015). The effects of learning and forgetting to the objective function and decision variables will also be studied.

2. Literature Review

According to Baker (1974), scheduling is the allocation of resources to perform a set of tasks over time. Scheduling is performed in various activities of manufacturing system such as production (Mattfeld, 2013), maintenance (Garg, Rani & Sharma, 2013) and logistics (Chang, Wu, Lee & Shen, 2014). Scheduling also plays important roles in service industries such as in health care (Aickelin & Dowsland, 2004), transportation (Abbink, Fischetti, Kroon, Timmer & Vromans, 2005) and energy (Patwal, Narang & Garg, 2018).

In manufacturing systems, machine and operator are among the most considered resources, and they are often studied separately (Pinedo, 2002). To represent real systems better, machine and operator need to be considered simultaneously in scheduling (Van den Bergh, Beliën, De Bruecker, Demeulemeester & De Boeck, 2013), such as in Mencía, Sierra, Mencía and Varela (2015) and Frihat, Sadfi and Hadj-Alouane (2014). Mencía et al. (2015) proposed a simultaneous scheduling model of machine and operator, considering operation sequence to minimize makespan in job shop production systems, while Frihat et al. (2014) proposed scheduling model to minimize personnel cost, considering time lag between operations and due date. In both papers, processing times were assumed to be fixed

In industry, operation's processing time may not be fixed. Kellerer and Strusevich (2008) and Grigoriev and Uetz (2005) proposed a process acceleration by allocating additional tools. Processes can also be accelerated by assigning additional operator as proposed in Aftab, Umer and Ahmad (2012) to minimize makespan, in Chaudhry (2010) to minimize flow time, and in Chaudhry and Drake (2009) to minimize total tardiness. Processing times can also vary due to different skill level among operators. This occurs in manufacturing systems where there are alternative operators to perform each operation, and different operator assignments may lead to different processing times. This situation has been investigated by Costa et al. (2013) who proposed a job scheduling model in a single-stage system with different operator's set up times to minimize makespan.

Different with skill levels, processing times can change over time because of the so-called time-changing effects (Strusevich & Rustogi, 2017). There are two distinct forms of time-changing effects: a deterioration effect causes the processing time to increase over time, while a learning effect causes the processing time to decrease over time. In industries with manual operations, understanding the learning effect is important for setting time standards, estimating labour costs and scheduling (Nembhard & Uzumeri, 2000). The learning effect occurs when an operator performs a particular operation repeatedly, and will be followed by a forgetting effect starting straight away when the operation stops (Yusriski et al., 2015). The learning effect phenomenon was firstly reported by Wright (1936) who showed that repetition in production process led to a constant decrease in processing time every time the cumulative produced quantity doubled. Learning and forgetting effects occur differently among operators (Jaber & Bonney, 1996). Nembhard and Uzumeri (2000) conducted an experiment in textile and automotive manufacturers and found that operators who learned more gradually tended to reach higher productivity in long term, while those who learned faster tended to experience more rapid forgetting during break times. This finding suggests managers to assign fast learning operators to short batch productions and not to assign them to secondary tasks (as they will easily forget their primary task), and to assign gradual learning operators to long batch productions and may assign them to secondary tasks. Additionally, through an experiment in an automotive manufacturer, Sebrina and Cakravastia (2011) found that learning rates were influenced by product complexity and takt times, and that a faster learning effect might lead to a higher defect rate.

To accelerate the processing of a job, it is common in industries to divide parts being produced into several batches or sublots (Baker & Jia, 1993). Through this, parts processing can overlap, thus reducing machine idle time and allowing completion time. The benefits of batch scheduling in flow shops include better makespan, mean flow time and average work-in-process level (Kalir & Sarin, 2000). The more batches made for processing parts, the more setups will be required, but in the same time, the more overlaps are allowed for parts. Thus, the problems in batch scheduling are to find the optimal number of batches and batch sizes, and to determine the schedule and the sequence of the resulting batches (Halim et al., 1994).

3. Model Development

Parameters and variables used in this paper are shown as follows.

Parameters:

n = number of parts to be processed (units),

b = number of operators (people),

c = number of machines (units),

d = due date, calculated from $t = 0$,

$s_{m,o}$ = set up time per batch on machine m if performed by operator o (time-unit),

$t_{m,o}$ = processing time of the first unit on machine m if performed by operator o (time-unit),

δ_o = learning rate of operator o,

ℓ_o = learning gradient of operator o.

Variables:

F = total actual flow time of all parts (time-unit),

N = number of batches,

$Q_{[i]}$ = batch size, number of parts in the batch sequenced in position i from the due date (units),

$B_{m,[i]}$ = process start time of the batch sequenced in position i from the due date at machine m (time-unit),

$X_{m,o}$ = binary variable that equals to 1 if operator o is assigned to machine m, equals to 0 if not,

$f_{m,o,[i]}$ = forgetting gradient of all parts in the batch sequenced in position i from the due date at machine m by operator o,

$\alpha_{m,o,[i]}$ = equivalent number of part of retained learning experience when starting the batch sequenced in position i from the due date at machine m if performed by operator o,

$T_{[p]}$ = processing time of the p-th part as a learning function,

$\hat{T}_{[p]}$ = processing time of the p-th as a forgetting function,

$T_{m,o,[i]}$ = processing time of all parts in the batch sequenced in position i from the due date at machine m by operator o,

W = set of operators assigned to machine 1 to b.

According to Jaber and Bonney (1996), learning effect occurs when a person performs an activity repeatedly. In manufacturing systems, the learning effect experienced by operators results in decreasing unit processing time as the number of repetition increases, following a learning function (Wright, 1936):

$$T_{[p]} = T_{[1]}p^{-\ell} \tag{1}$$

where $\ell = -{}^{2}\log\delta$ (δ has a value of 0.7 to 0.9 in manufacturing system; smaller value means faster learning).

The fluctuation of part processing time caused by learning and forgetting effects in a batch production are shown in Fig. 1. Suppose that a learning effect occurs for $T_{m,o,[i+1]}$ during the processing of batch i+1 and followed by a forgetting effect occurs for T_R. Suppose also that during $T_{m,o,[i+1]}$, $Q_{[i+1]}$ parts are processed and that during T_R, R parts can be processed if there is no interruption. ACD curve in Fig. 1 represents the learning function in Equation (1), while BCE curve shows a forgetting function shown in Equation (2).

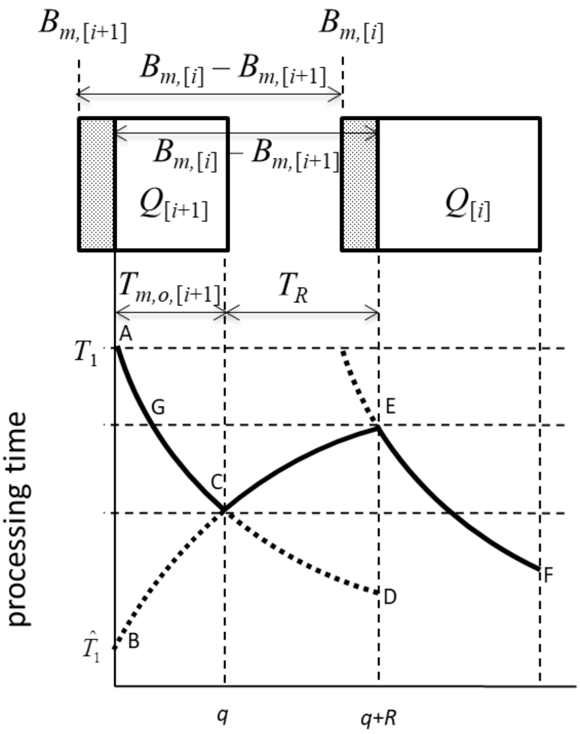

Figure 1. Learning and forgetting effects during two consecutive batches

$$\hat{T}_{[p]} = \hat{T}_{[1]}p^{f} \tag{2}$$

At C, the value of $\hat{T}_{[p]}$ will be equal to $T_{[p]}$, or $T_{[1]}Q_{[i+1]}^{-\ell} = \hat{T}_{[1]}Q_{[i+1]}^{f}$, which can be used to find $\hat{T}_{[1]}$, that is:

$$\hat{T}_{[1]} = T_{[1]}Q_{[i+1]}^{-(\ell+f)} \tag{3}$$

Substituting Equation (3) to Equation (2) we obtain:

$$\hat{T}_{[p]} = T_{[1]}Q_{[i+1]}^{-(\ell+f)} \cdot p^{f} \tag{4}$$

To find T_R, i.e. the time required to produce R parts (if interruption does not occur), the learning function in Equation (1) is integrated along this interval:

$$T_R = \int_{Q_{[i+1]}}^{Q_{[i+1]}+R} T_{[1]} p^{-\ell} dp = \frac{T_{[1]}}{1-\ell}\left[\left(Q_{[i+1]}+R\right)^{1-\ell} - Q_{[i+1]}{}^{1-\ell}\right] \tag{5}$$

After some algebraic operations on Equation (5) to find $(Q_{[i+1]} + R)$ we obtain:

$$\left(Q_{[i+1]}+R\right) = \left[Q_{[i+1]}^{1-\ell} + (1-\ell)\frac{T_R}{T_{[1]}}\right]^{1/(1-\ell)} \tag{6}$$

After experiencing a forgetting effect during CE, $Q_{[i+1]}$ units of operator's learning experience obtained when processing batch i+1 will be reduced to α units, where α is the equivalent retained number of parts of the learning experience after forgetting ($0<\alpha<Q_{[i+1]}$). The value of α can be found by equating $\hat{T}_{[p]}$ at E (i.e. $\hat{T}_{Q_{[i+1]}+R}$) to $T_{[p]}$ at G (because $T_G = T_E$), i.e.:

$$T_{[1]}\alpha^{-\ell} = T_{[1]}Q_{[i+1]}^{-(\ell+f)}\left(Q_{[i+1]}+R\right)^{f},$$

which results:

$$\alpha = Q_{[i+1]}{}^{(\ell+f)/\ell}\left(Q_{[i+1]}+R\right)^{-f/\ell} \tag{7}$$

Substituting Equation (6) to Equation (7) and after some algebraic operations we obtain:

$$\alpha = Q_{[i+1]}^{(\ell+f)/\ell}\left(Q_{[i+1]}^{1-\ell} + (1-\ell)\frac{T_R}{T_{[1]}}\right)^{\frac{-f}{\ell(1-\ell)}} \tag{8}$$

According to Figure 1, the T_R value is given by $B_{m,[i]}-B_{m,[i+1]}-T_{m,o,[i+1]}$, while the value of $T_{[1]}$ is $t_{m,o}$. Thus, Equation (8) can be rewritten as follows:

$$\alpha = Q_{[i+1]}^{(\ell+f)/\ell}\left(Q_{[i+1]}^{1-\ell} + (1-\ell)\frac{B_{m,[i]}-B_{m,[i+1]}-T_{m,o,[i+1]}}{t_{m,o}}\right)^{\frac{-f}{\ell(1-\ell)}} \tag{9}$$

We should remember that the Equation (9) occurs when the operator experiences partial forgetting after batch i+1, or when $T_E < T_1$. If the operator experiences a total forgetting after batch i+1, then $T_E = T_1$ and $\alpha = 0$. Therefore, Equation (9) needs to be rewritten as follows:

$$\alpha = \max\left(0, Q_{[i+1]}^{(\ell+f)/\ell}\left(Q_{[i+1]}^{1-\ell} + (1-\ell)\frac{B_{m,[i]}-B_{m,[i+1]}-T_{m,o,[i+1]}}{t_{m,o}}\right)^{\frac{-f}{\ell(1-\ell)}}\right) \tag{10}$$

Furthermore, the value of the forgetting gradient f is determined at the beginning of the total forgetting, i.e. when the value of $\hat{T}_{[p]}$ becomes equal to $T_{[1]}$. This situation is given by $T_{[1]}Q_{[i+1]}^{-(\ell+f)}\left(Q_{[i+1]}+R\right)^{f} = T_{[1]}$, which can be used to find f:

$$f = \frac{\ell \log Q_{[i+1]}}{\log\left(Q_{[i+1]}+R\right) - \log Q_{[i+1]}} .$$

Using Equation (6) to substitute $(Q_{[i+1]} + R)$ and substituting T_R and T_1 with their values in Equation (9) we obtain:

$$f = \frac{\ell(1-\ell)\log Q_{[i+1]}}{\log\left(Q_{[i+1]}^{1-\ell} + (1-\ell)\frac{B_{m,[i]} - B_{m,[i+1]} - T_{m,o,[i+1]}}{t_{m,o}}\right) - \log Q_{[i+1]}} \tag{11}$$

Since this study considers multiple operator, machine and batch, ℓ will be different for each operator, while f and a will be different for each machine, operator and batch. Therefore, notations of ℓ, f and a are replaced by ℓ_o, $a_{m,o,[i]}$ and $f_{m,o,[i]}$, which is formulated as:

$$\ell_o = -{}^2\log\delta_o \tag{12}$$

$$\alpha_{m,o,[i]} = \max\left(0, Q_{[i+1]}^{(\ell_o + f_{m,o,i})/\ell_o}\left[Q_{[i+1]}^{1-\ell_o} + (1-\ell_o)\frac{B_{m,[i]} - B_{m,[i+1]} - T_{m,o,[i+1]}}{t_{m,o}}\right]^{\frac{-f_{m,o,[i]}}{\ell_o(1-\ell_o)}}\right) \tag{13}$$

$$f_{m,o,[i]} = \frac{\ell_o(1-\ell_o)\log Q_{[i+1]}}{\log\left(Q_{[i+1]}^{1-\ell_o} + (1-\ell_o)\frac{B_{m,[i]} - B_{m,[i+1]} - T_{m,o,[i+1]}}{t_{m,o}}\right) - \log Q_{[i+1]}} \tag{14}$$

For batch production, the batch processing time is determined by integrating part processing time, $T_{[x]}$, from 0 to $Q_{[N]}$ for batch N, and from $a_{m,o,[i]}$ to $a_{m,o,[i]}+Q_{[i]}$ for batch i = 1, ..., N-1. The calculation of batch processing time is given in Equation (15) and (16).

$$T_{m,o,[N]} = t_{m,o}\int_0^{Q_N} p^{-\ell_o}\,dp = t_{m,o}\frac{Q_{[N]}^{1-\ell_o}}{1-\ell_o} \tag{15}$$

$$T_{m,o,[i]} = t_{m,o}\int_{\alpha_{m,o,[i]}}^{\alpha_{m,o,[i]}+Q_{[i]}} p^{-\ell_o}\,dp = \frac{t_{m,o}}{1-\ell_o}\left[\left(\alpha_{m,o,[i]} + Q_{[i]}\right)^{1-\ell_o} - \alpha_{m,o,[i]}^{1-\ell_o}\right] \tag{16}$$

4. Model and Algorithm

This research extends the flow shop batch scheduling model in Halim and Ohta (1993) by introducing the existence of alternative operators to perform each operation studied in Costa et al. (2013), and by considering time-changing effects of learning and forgetting studied in Jaber and Bonney (1996). We will look at the effects of learning and forgetting effects to number of batches, batch sizes and the schedule of the resulting batches.

The problem studied in this research is described as follows. There are n parts will be processed in N batches, and each batch i will be processed through b operations with uniform routing. Each operation can be performed by one of c alternative operators with different set up times $s_{m,o}$ and initial processing times $t_{m,o}$. Each operator o experiences a learning effect at a rate of δ_o. All operations must be finished no later than a due date d. The decision variables in the model are assignment of operator o to machine m ($X_{m,o}$), the number of batches N, batch sizes $Q_{[i]}$, and the schedule of operation m in batch i ($B_{m,[i]}$). This study develops a model of simultaneous batch scheduling and operator assignment to minimize actual flow time in flow shops.

Assumptions used in this study are:

1. All parts, machines and operators are ready (can be scheduled) at t = 0.
2. Interruption of an operation are not allowed.
3. Each machine and operator cannot perform more than one operation at a time.

4. Machines are always available during the scheduling time horizon.
5. Set up and process of a particular operation are performed by the same operator.
6. An operator can be assigned to at most one machine.

Based on the analysis explained in Section 3, the flow shop batch scheduling and operator assignment problem to minimize actual flow time was formulated in Model 1 as follows.

Model 1.

Minimize

$$F = \sum_{i=1}^{N} \left(d - B_{1,[i]}\right) Q_{[i]} \tag{17}$$

subject to

$$B_{c,[i]} = d - \sum_{o=1}^{b} X_{c,o} \left(i s_{c,o} + \sum_{j=1}^{i} T_{c,o,[j]} \right), \; i = 1, \ldots, N \tag{18}$$

$$B_{m,[1]} = B_{m+1,[1]} - \sum_{o=1}^{b} X_{m,o} \left(s_{m,o} + T_{m,o,[1]}\right), \; m = 1, \ldots, c{-}1 \tag{19}$$

$$B_{m,[i]} \leq B_{m,[i-1]} - \sum_{o=1}^{b} X_{m,o} \left(s_{m,o} + T_{m,o,[i]}\right), \; m = 1, \ldots, c{-}1, \; i = 2, \ldots, N \tag{20}$$

$$B_{m,[i]} \leq B_{m+1,[i]} - \sum_{o=1}^{b} X_{m,o} \left(s_{m,o} + T_{m,o,[i]}\right), \; m = 1, \ldots, c{-}1, \; i = 2, \ldots, N \tag{21}$$

$$B_{1,[N]} > 0 \tag{22}$$

$$T_{m,o,[N]} = t_{m,o} \frac{{Q_{[N]}}^{1-\ell_o}}{1 - \ell_o}, \; m = 1, \ldots, c, \; o = 1, \ldots, b \tag{23}$$

$$T_{m,o,[i]} = \frac{t_{m,o}}{1 - \ell_o} \left[\left(\alpha_{m,o,[i]} + Q_{[i]}\right)^{1-\ell_o} - \alpha_{m,o,[i]}^{1-\ell_o} \right], \quad m = 1, \ldots, c, \; o = 1, \ldots, b, \; i = 1, \ldots, N\text{-}1 \tag{24}$$

$$\alpha_{m,o,[i]} = \max\left(0, Q_{[i+1]}^{(\ell_o + f_{m,o,[i]})/\ell_o} \left[Q_{[i+1]}^{1-\ell_o} + \left(1 - \ell_o\right) \frac{B_{m,[i]} - B_{m,[i+1]} - T_{m,o,[i+1]}}{t_{m,o}} \right]^{\frac{-f_{m,o,[i]}}{\ell_o (1-\ell_o)}} \right), \quad m = 1, \ldots, c, \; o = 1, \ldots, b, \; i = 1, \ldots, N\text{-}1 \tag{25}$$

$$\ell_o = -{}^{2}\log \delta_o, \; o = 1, \ldots, b \tag{26}$$

$$f_{m,o,[i]} = \frac{\ell_o \left(1 - \ell_o\right) \log Q_{[i+1]}}{\log\left(Q_{[i+1]}^{1-\ell_o} + \left(1 - \ell_o\right) \frac{B_{m,[i]} - B_{m,[i+1]} - T_{m,o,[i+1]}}{t_{m,o}} \right) - \log Q_{[i+1]}}, \quad m = 1, \ldots, c, \; o = 1, \ldots, b, \; i = 1, \ldots, N-1 \tag{27}$$

$$\sum_{i=1}^{N} Q_{[i]} = n, \tag{28}$$

$$\sum_{o=1}^{b} X_{m,o} = 1,\ m = 1, \ldots, c \tag{29}$$

$$\sum_{m=1}^{c} X_{m,o} \le 1,\ o = 1, \ldots, b \tag{30}$$

$$X_{m,o} \in \{0, 1\},\ \forall m, o \tag{31}$$

$$Q_{[i]} > 0,\ 1 < N < n,\ i = 1,\ldots, N. \tag{32}$$

Objective function (17) is minimization of total actual flow time, i.e. the sum of batch actual flow time multiplied by its size. Batch actual flow time is the time spent by the batch on the shop floor, i.e. the time length from the batch arrival at machine 1 to the due date. The starting time calculation of each batch at each machine is expressed in Constraint (18) to (21). Since this study performs a backward scheduling, the batch starting time is calculated from the last machine. Constraint (18) states that the starting time of batch *i* at machine *m* is the due date less by the sum of set up and processing time of batch 1 to batch *i*. Constraint (19) states that the starting time of batch 1 at machine *m* is calculated backward from machine *c* to machine *m*. Constraint (20) and (21) state that the starting time of other than batch 1 at other than machine *c* is the earliest between the starting time of the previous batch at the same machine and the starting time of the batch on the next machine, less by its set up and processing time. Set up and processing times in Constraint (18) to (21) are multiplied by operator assignment variable, because each operator has different set up and processing times at each machine. Constraint (22) states that the starting time of batch *N* at machine 1 must be non-negative.

Since parts in a batch have different processing times, we use batch processing times instead of part processing times in Constraint (18) to (21). The batch processing times are defined in Constraint (23) and (24), which are taken from Equation (15) and (16) in Section 2. The $a_{m,o,[i]}$ variables are the equivalent retained number of parts of learning experience from batch *i*+1 after forgetting effect, which is influenced by the previous batch learning result, interruption duration, initial processing time, learning gradient and forgetting gradient. The value of $a_{m,o,[i]}$ will be zero if the operator experiences a total forgetting, i.e. loses all learning experience in the previous batch. Constraint (26) shows that the learning gradient is influenced by the learning rate, while Constraint (27) indicates that the forgetting gradient is influenced by learning experience in the previous batch, the interruption duration and the learning gradient.

Constraint (28) states that the number of parts in all batches must be equal to the total number of parts. Constraint (29) states that exactly one operator must be assigned to each machine, while Constraint (30) states that an operator can be assigned to maximum one machine. Constraint (31) states that operator assignment variables are binary numbers, and Constraint (32) states that batch sizes must be positive, and the batch number must be at least one and at most equal to the total number of parts.

The existence of learning and forgetting effects experienced by operators is important to consider in the model since learning and forgetting affect the objective function and batch sizes. The effects of learning and forgetting to batch scheduling are explained in the following propositions.

Proposition 1. *The size of the closest batch to the due date is larger than any other batches in the schedule.*

Proposition 2. *The faster operators learn, the larger the difference between the closest batch to the due date and the any other batches in the schedule.*

Proposition 3. *The faster operators learn, the lower number of batches in the optimal solution.*

Proposition 4. *The faster operators learn, the lower the value of the objective function F and $Q_{[1]}$ will be larger than $Q_{[i]}$ not more by*

$$\left[\frac{(1-\ell_c)(1-\ell_1)(t_c+t_1)}{t_c(1-\ell_1)+t_1(1-\ell_c)}\right]^{\ell_c}.$$

Proofs. See Appendix.

The problem formulated in Model 1 cannot be solved when N is unknown. It is necessary to relax N from a decision variable to a parameter, so that Model 1 becomes solvable in $O(bc2^{[n/\varepsilon]-1})$ time (where ε ∈ {1, 0.1, 0.01, …} is the desired accuracy of $Q_{[i]}$ value). The problem formulated in Model 1 is NP-hard since it can be reduced to a flow shop batch scheduling problem without operator assignment and without learning and forgetting effects which is classified as NP-hard in Mortezaei and Zulkifli (2013).

To find the optimal solution for the problem, we need to try all possible N values ($1 < N < n$) and find the best objective value. However, this is time inefficient as the computation time will increase rapidly as N increases. Therefore, using the convexity of F as conjectured in Bukchin, Tzur and Jaffe (2002), a solution method for the problem will try several N values, starting from one, and increase it one-by-one until the value of F does not improve anymore. If the objective function F does not improve at an N value, we stop the computation and set the last improving F as our solution. The solution method for Model 1 is described in Algorithm 1.

Algorithm 1.

Step 1. Set parameters n, d, b, c, $s_{m,o}$, $t_{m,o}$, δ_o, ℓ_o and d. Go to Step 2.

Step 2. Set $N = 1$. Go to Step 3.

Step 3. Solve Model 1, determine F, the best solution for the current N. Go to Step 4.

Step 4. If $F < F^*$ or F^* has not been set, set F as F^*, go to Step 5. If not, go to Step 6.

Step 5. If $N = n$, go to Step 6. If not, set $N = N + 1$, and return to Step 3.

Step 6. Stop. The current F^* is the optimal solution.

Note that in Step 1, parameters b and c are given by the system, while n and d are obtained from customer order data. Additionally, $s_{m,o}$ and $t_{m,o}$ are the standard setup and processing times, and δ_o is obtained by measuring the change of processing time over the time for each operator.

In Algorithm 1, we solved Model 1 after setting a value for N. We used Lingo to solve Model 1 in Step 3, as Lingo is the most comprehensive optimization tool, capable to solve small or large models efficiently, and capable to solve linear or non-linear models (Goodarzi, Ziaei & Hosseinipour, 2014).

5. Results and Discussion

The proposed model and algorithm were implemented to 18 data sets, consisting of 2 x 3 (2 machines and 3 operators), 2 x 4, 3 x 4, 3 x 6, 4 x 8, 4 x 4, 6 x 6 and 8 x 8 data sets. Data Set 1 to 8 examined a situation where more operators were available than machines, thus one or more operators were unassigned, while Data Set 9 to 16 examined a situation where the number of operators was equal to the number of machines. Additionally, Data Set 17 and 18 were specially developed to perform a sensitivity analysis. Parameters of the data sets are shown in Table 1.

Implementing Algorithm 1 to the data sets, the results are shown in Table 2. For example, the optimal solution for Data Set 1 was obtained when parts were split into 3 batches, the batch sizes were 97.6, 1.1 and 1.3 respectively (remember that batch 1 is the closest batch to the due date), operator 1 and 3 were assigned to machine 1 and 2 respectively and the objective function value was 21678.6. If the optimal solution for a data set is achieved at an N^* value, then the algorithm must try N values from 1 to N^*+1. Set of operator W^* shows the operator assignment in the optimal solution, e.g. $W^* = 13$ for Data Set 1 means that operator 1 and 3 are assigned to machine 1 and 2 respectively.

A calculation example using the algorithm is given for Data Set 12 in Table 3. We tried some N values, started from $N = 1$. By increasing N value one-by-one, the objective function value improved until $N = 6$, and then it stopped improving at $N = 7$. The calculation was then stopped and the objective function value at $N = 6$ was set as the optimal solution. The resulting Gantt-charts for Data Set 1, 5 and 14 are shown in Figure 2.

Data Set	n	c	b	d	$s_{m,o}$	$t_{m,o}$	δ_o (x 100)
1	100	2	3	3000	o=1 o=2 o=3 m=1 32 30 34 m=2 48 50 45	o=1 o=2 o=3 m=1 4 6 5 m=2 6 5 3	70 80 75
2	100	2	3	3000	49 78 82 75 89 72	13 9 14 13 10 11	73 79 87
3	100	2	4	3000	41 55 47 67 99 80 98 48	15 10 5 14 7 8 7 5	85 88 78 88
4	100	2	4	3000	49 74 92 80 78 55 53 47	7 11 12 11 14 12 14 6	83 72 84 88
5	100	3	4	3000	41 55 47 67 99 80 98 48 79 58 74 82	15 10 5 14 7 8 7 5 14 7 6 9	85 88 78 88
6	100	3	4	3000	49 74 92 80 78 55 53 47 89 80 51 57	7 11 12 11 14 12 14 6 15 8 12 6	83 72 84 88
7	100	3	6	3000	47 83 73 50 53 54 83 72 63 86 60 98 63 66 71 77 81 71	12 11 9 6 9 8 11 12 7 12 13 7 11 7 14 9 7 13	77 73 71 73 79 70
8	100	4	8	3000	84 48 88 83 55 71 51 82 93 60 57 88 68 86 92 71 74 66 96 73 66 68 81 46 76 67 78 91 96 48 62 68	9 6 13 14 14 8 8 12 7 10 13 13 8 13 13 7 9 6 10 12 10 10 15 14 10 11 9 12 13 7 13 6	77 82 71 87 74 85 77 87
9	100	4	4	3000	84 48 88 83 93 60 57 88 74 66 96 73 76 67 78 91	5 8 13 10 5 6 10 14 7 15 5 11 7 14 10 7	77 82 71 87
10	100	4	4	3000	55 71 51 82 68 86 92 71 66 68 81 46 96 48 62 68	11 8 14 15 15 8 8 10 7 12 14 10 5 13 9 14	74 85 77 87
11	100	4	4	3000	47 83 73 50 83 72 63 86 63 66 71 77 73 53 59 78	9 6 13 14 7 10 13 13 9 6 10 12 10 11 9 12	78 81 80 73
12	100	4	4	3000	53 54 44 89 60 98 78 61 81 71 69 86 55 68 80 51	14 8 8 12 8 13 13 7 10 10 15 14 13 7 13 6	86 76 81 75
13	100	4	4	3000	41 93 49 68 93 63 87 90 46 43 61 65 46 43 44 63	12 11 9 6 11 12 7 12 11 7 14 9 6 8 8 10	77 73 71 73

Data Set	n	c	b	d	$s_{m,o}$	$t_{m,o}$	δ_o (x 100)
14	100	4	4	3000	52 85 60 70 90 81 42 69 58 56 64 55 41 81 74 42	9 8 5 12 13 7 10 10 7 13 14 13 9 9 7 8	79 70 80 90
15	100	6	6	3000	47 83 73 50 53 54 83 72 63 86 60 98 63 66 71 77 81 71 73 53 59 78 55 68 41 93 49 68 52 85 93 63 87 90 90 81	12 11 9 6 9 8 11 12 7 12 13 7 11 7 14 9 7 13 6 8 8 10 9 9 9 6 13 14 14 8 7 10 13 13 8 13	77 73 71 73 79 70
16	100	8	8	3000	84 48 88 83 55 71 51 82 93 60 57 88 68 86 92 71 74 66 96 73 66 68 81 46 76 67 78 91 96 48 62 68 47 83 73 50 53 54 44 89 83 72 63 86 60 98 78 61 63 66 71 77 81 71 69 86 73 53 59 78 55 68 80 51	9 6 13 14 14 8 8 12 7 10 13 13 8 13 13 7 9 6 10 12 10 10 15 14 10 11 9 12 13 7 13 6 5 8 13 10 11 8 14 15 5 6 10 14 15 8 8 10 7 15 5 11 7 12 14 10 7 14 10 7 5 13 9 14	77 82 71 87 74 85 77 87
17	100	4	4	3000	53 54 44 89 60 98 78 61 81 71 69 86 55 68 80 51	14 8 8 12 8 13 13 7 10 10 15 14 13 7 13 6	76 66 71 65
18	100	4	4	3000	53 54 44 89 60 98 78 61 81 71 69 86 55 68 80 51	14 8 8 12 8 13 13 7 10 10 15 14 13 7 13 6	96 86 91 85

Table 1. Parameter of data sets

After implementing the model and algorithm to the data sets, we can see that the model worked properly for 2 to 8-machine flow shops (see Table 2). Batch 1 (the closest batch to the due date) was always larger than any other batches in the schedule. This confirms the relationships stated in Proposition 1, and is consistent with the single-machine model proposed in Yusriski et al. (2015).

To investigate this behaviour further, Data Set 1 and 12 were modified to a minimum learning by setting $\delta_o = 1$ for all *o* (the modified data sets were named as Data Set 1a and 12a). The results, as seen in Table 4, showed that a minimum learning has made batch 1 not anymore larger than the other batches in the modified data sets. This confirms that the behaviour (batch 1 is larger than the other batches) is caused by the existence of learning and forgetting effects, and that the faster operators learn, the larger the difference between batch 1 and the other batches, as stated in Proposition 2.

The behaviour that batch 1 is much larger than the other batches can be explained as follows. Please remember that batch 1 is the last batch processed in the schedule. The operator who processes a large batch will experience a long learning effect, which will be followed by a rapid forgetting effect. By scheduling the largest batch to be processed last in the schedule (as batch 1), this rapid forgetting effect will not affect any other batches in the schedule. This is why the model schedules the largest batch as batch 1.

Data Set	N*	$Q_{[i]}$	W*	F*
1	3	97.6; 1.1; 1.3	13	21678.6
2	4	89.4; 1.2; 4.6; 4.8	21	68622.8
3	4	88.4; 3; 4.4; 4.2	34	47950.5
4	6	86.5; 1.3; 3.2; 3.3; 4; 1.7	12	59083.9
5	6	79.9; 2.9; 4.4; 4.3; 4.4; 4.1	342	84192.4
6	4	85.9; 2.7; 5.7; 5.7	142	86858.8
7	4	92.7; 1.2; 2.9; 3.2	432	57344.9
8	2	98.4; 1.6	7523	103462.3
9	5	81.9; 3.7; 5.7; 5; 3.7	1234	106954.0
10	3	86.6; 6.5; 6.9	2341	39443.1
11	4	90.3; 1.4; 4; 4.3	4123	120404.1
12	6	82.5; 2.7; 3.6; 3.6; 3.7; 3.9	3124	117576.6
13	5	89.5; 1.3; 3.5; 4; 1.7	4321	79231.2
14	8	75.3; 3.5; 3.6; 3.6; 3.6; 3.7; 4.3; 2.4	3214	110016.3
15	3	91.3; 4.1; 4.6	432165	135391.8
16	10	54.6; 2.9; 5.3; 5.3; 5.3; 5.3; 5.3; 5.4; 5.2; 5.4	75286134	202723.3
17	2	98.8; 1.2	3124	78139.8
18	14	35.6; 4.9; 4.9; 5; 4.9; 4.9; 4.9; 5; 5; 4.9; 4.9; 4.9; 5; 5.2	3124	135923.6

Table 2. Computation results

N	$Q_{[i]}$	F	W
1	100	130385.0	3124
2	96.2; 3.8	124913.5	3124
3	92.7; 3.5; 3.8	122667.3	3124
4	89.2; 3.2; 3.7; 3.9	120614.3	3124
5	86.8; 3.1; 4.1; 2; 3.9	119470.4	3124
6	82.5; 2.7; 3.6; 3.6; 3.7; 3.9	117576.6*	3124
7	80.5; 3.6; 3.6; 3.6; 3.7; 4; 1	118743.0	3124

* Optimal

Table 3. Calculation example for Data Set 12

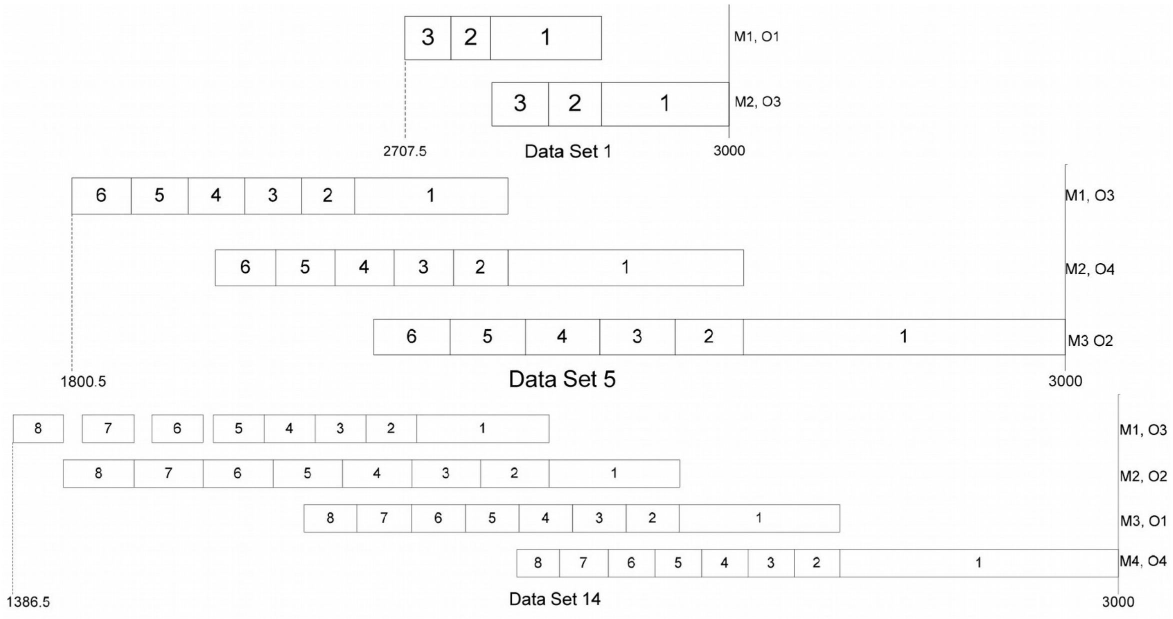

Figure 2. Gantt-charts for Data Sets 1, 5 and 14

Data Set	Parameter			N*	$Q_{[i]}$	W*	F*
	n	m	o				
1	100	2	3	3	97.6; 1.1; 1.3	13	21678.6
1a#	100	2	3	6	17.1; 18.5; 20.3; 22.7; 14.7; 6.7	13	43503.2
12	100	4	4	6	82.5; 2.7; 3.6; 3.6; 3.7; 3.9	3124	117576.6
12a#	100	4	4	9	8.3; 10.3; 12.2; 14; 15.7; 17; 12.3; 7.5; 2.7	3214	112462.7

#Data Set 1a and 12a are Data Set 1 and 12 without learning and forgetting

Table 4. Impact of learning and forgetting elimination to Data Set 1 and 12

A sensitivity analysis was performed to identify the effect of changing the parameters to the solution. The sensitivity analysis was performed by changing the learning rate, while the other parameter values were kept constant. Data Set 12 was taken as a reference. The learning effect in Data Set 12 was accelerated by decreasing the learning rate (δ_0) by 0.1 (became Data Set 17), and was decelerated by increasing the learning rate (δ_0) by 0.1 (became Data Set 18). The result showed that the faster operators learn, the lower the number of batches in the optimal solution, as stated in Proposition 3. This occurred because operators with faster learning experienced a faster forgetting, so the model reduced the number of batches to reduce the forgetting effects between batches. This finding is consistent with the finding in Nembhard and Uzumeri (2000) through an empirical experiment which suggests that faster learning operators will also experience faster forgetting effect. Another finding of the sensitivity analysis was that the faster operators learn, the lower total actual flow time in the optimal solution as stated in Proposition 4. This occurred because the faster operators learned, the shorter time they needed to process batches. The effects of the learning rate changes are shown in Figure 3.

Since the objective function F is a unimodal function (i.e. to have exactly one extreme point) of N according to Bukchin et al. (2002), we can stop computation when F stops improving. Based on Bukchin et al. (2002), if the value of F* does not increase at an N value, then it will never improve again when N is increased. Thus, the last lowest F* value before it increases is the optimal solution for the problem.

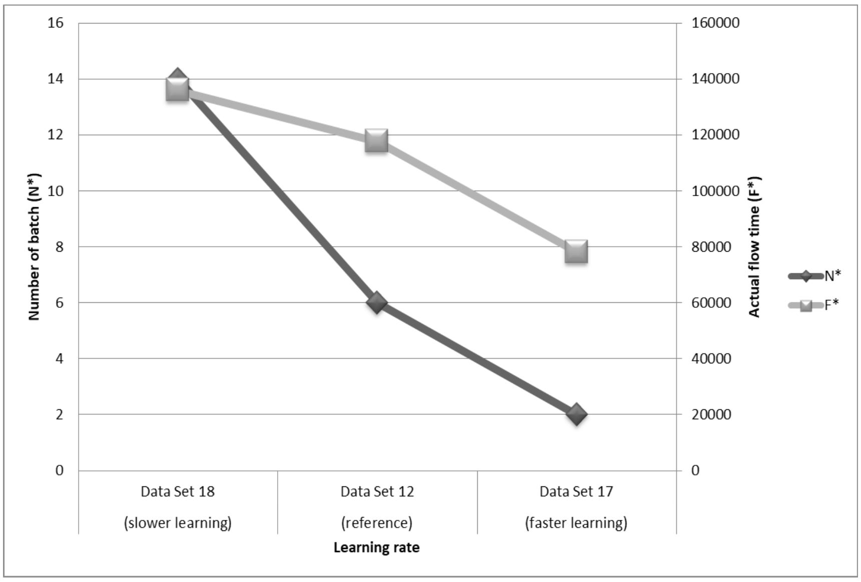

Figure 3. The result of sensitivity analysis on learning rate

The advantage of considering learning and forgetting in batch scheduling can be explained as follows. As we can see in Table 4, ignoring the effects of learning and forgetting results more batches in the schedule and a higher objective function value. More batches means more setup costs spent by the system than required, while a higher actual flow time brings a higher inventory cost. Both brings out inefficiency due to poor scheduling. As stated in Nembhard and Uzumeri (2000), understanding the learning effect is important for setting time standards, estimating labour costs and scheduling.

In addition to Algorithm 1, future studies can use evolutionary algorithm heuristics to find the solution for Model 1, such as those presented in Garg (2016) and Garg (2019). The last two mentioned papers provided a beneficial approach for solving constrained nonlinear optimization problems. The solution and the computation time obtained from these heuristics can be compared to those obtained from Algorithm 1.

6. Concluding Remarks

This research developed a flow shop batch scheduling and operator assignment model with learning and forgetting effects to minimize total actual flow time. To solve the model, an algorithm was developed, and the algorithm worked by trying different number of batches, starting from one, then increasing the number of batches one-by-one until the objective function value did not improve anymore. We mathematically and numerically showed that the model scheduled the largest batch to be processed last in the schedule, and the faster operators learn, the larger the difference between the closest batch to the due date and the other batches, the lower number of batches in the optimal solution, and the lower the objective function. Considering the effects of learning and forgetting in a batch scheduling brings a better decision on batch sizes and a better calculation on the objective function value. Future works need to utilize evolutionary algorithm heuristics as an effort to improve the solution as well as the computation time.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

References

Abbink, E., Fischetti, M., Kroon, L., Timmer, G., & Vromans, M. (2005). Reinventing crew scheduling at Netherlands Railways. *Interfaces,* 35, 393-401. https://doi.org/10.1287/inte.1050.0158

Aftab, M.T., Umer, M., & Ahmad, R. (2012). Jobs scheduling and worker assignment problem to minimize makespan using ant colony optimization metaheuristic. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering,* 6, 2823-2826.

Aickelin, U., & Dowsland, K.A. (2004). An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research,* 31, 761-778. https://doi.org/10.1016/S0305-0548(03)00034-0

Baker, K.R. (1974). *Introduction to sequencing and scheduling*. John Wiley & Sons.

Baker, K.R., & Jia, D. (1993). A comparative study of lot streaming procedures. *Omega,* 21, 561-566. https://doi.org/10.1016/0305-0483(93)90024-F

Bedworth, D.D., & Bailey, J.E. (1987). *Integrated production control system: Management, analysis, design*.

Bukchin, J., Tzur, M., & Jaffe, M. (2002). Lot splitting to minimize average flow-time in a two-machine flow-shop. *IIE transactions,* 34, 953-970. https://doi.org/10.1080/07408170208928925

Chang, F.-S., Wu, J.-S., Lee, C.-N., & Shen, H.-C. (2014). Greedy-search-based multi-objective genetic algorithm for emergency logistics scheduling. *Expert Systems with Applications,* 41, 2947-2956. https://doi.org/10.1016/j.eswa.2013.10.026

Chaudhry, I.A. (2010). Minimizing flow time for the worker assignment problem in identical parallel machine models using GA. *The International Journal of Advanced Manufacturing Technology,* 48, 747-760. https://doi.org/10.1007/s00170-009-2323-1

Chaudhry, I.A., & Drake, P.R. (2009). Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms. *The International Journal of Advanced Manufacturing Technology,* 42, 581-594. https://doi.org/10.1007/s00170-008-1617-z

Costa, A., Cappadonna, F., & Fichera, S. (2013). A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology,* 69, 2799-2817. https://doi.org/10.1007/s00170-013-5221-5

Frihat, M., Sadfi, C., & Hadj-Alouane, A.B. (2014). Optimization of integrated employee timetabling and Hybrid Job Shop Scheduling under time lag constraints. *International Conference on Control, Decision and Information Technologies (CoDIT)*, 282-287. https://doi.org/10.1109/CoDIT.2014.6996907

Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation,* 274, 292-305. https://doi.org/10.1016/j.amc.2015.11.001

Garg, H. (2019). A hybrid GSA-GA algorithm for constrained optimization problems. *Information Sciences,* 478, 499-523. https://doi.org/10.1016/j.ins.2018.11.041

Garg, H., Rani, M., & Sharma, S. (2013). Preventive maintenance scheduling of the pulping unit in a paper plant. *Japan journal of industrial and applied mathematics,* 30, 397-414. https://doi.org/10.1007/s13160-012-0099-4

Goodarzi, E., Ziaei, M., & Hosseinipour, E.Z. (2014). *Optimization analysis using LINGO and MATLAB. Introduction to Optimization Analysis in Hydrosystem Engineering*. Springer. https://doi.org/10.1007/978-3-319-04400-2_5

Grigoriev, A., & Uetz, M. (2005). Scheduling parallel jobs with linear speedup. *International Workshop on Approximation and Online Algorithms*, 203-215. https://doi.org/10.1007/11671411_16

Halim, A.H., Miyazaki, S., & Ohta, H. (1994). Batch-scheduling problems to minimize actual flow times of parts through the shop under JIT environment. *European Journal of Operational Research,* 72, 529-544. https://doi.org/10.1016/0377-2217(94)90421-9

Halim, A.H., & Ohta, H. (1993). Batch-scheduling problems through the flowshop with both receiving and delivery just in time. *The International Journal Of Production Research,* 31, 1943-1955. https://doi.org/10.1080/00207549308956833

Jaber, M.Y., & Bonney, M. (1996). Production breaks and the learning curve: the forgetting phenomenon. *Applied Mathematical Modelling,* 2, 162-169. https://doi.org/10.1016/0307-904X(95)00157-F

Kalir, A.A., & Sarin, S.C. (2000). Evaluation of the potential benefits of lot streaming in flow-shop systems. *International Journal of Production Economics,* 66, 131-142. https://doi.org/10.1016/S0925-5273(99)00115-2

Kellerer, H., & Strusevich, V.A. (2008). Scheduling parallel dedicated machines with the speeding-up resource. *Naval Research Logistics (NRL),* 55, 377-389. https://doi.org/10.1002/nav.20292

Mattfeld, D.C. (2013). *Evolutionary search and the job shop: investigations on genetic algorithms for production scheduling.* Springer Science & Business Media.

Mencía, R., Sierra, M.R., Mencía, C., & Varela, R. (2015). Genetic algorithms for the job shop scheduling problem with operators. *Applied Soft Computing,* 34, 94-105. https://doi.org/10.1016/j.asoc.2015.05.004

Mortezaei, N., & Zulkifli, N. (2013). Integration of lot sizing and flow shop scheduling with lot streaming. *Journal of Applied Mathematics,* 2013. https://doi.org/10.1155/2013/216595

Nembhard, D.A., & Uzumeri, M.V. (2000). Experiential learning and forgetting for manual and cognitive tasks. *International journal of industrial ergonomics,* 25, 315-326. https://doi.org/10.1016/S0169-8141(99)00021-9

Patwal, R.S., Narang, N., & Garg, H. (2018). A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy,* 142, 822-837. https://doi.org/10.1016/j.energy.2017.10.052

Pinedo, M. (2002). *Scheduling: theory, algorithms, and systems.* Prentice Hall.

Sebrina, L.D., & Cakravastia, A. (2011). Learning curves in automobile assembly line. *International Conference on Industrial Engineering.* Stuttgart, Germany.

Strusevich, V.A., & Rustogi, K. (2017). *Scheduling with time-changing effects and rate-modifying activities.* Springer. https://doi.org/10.1007/978-3-319-39574-6

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research,* 226, 367-385. https://doi.org/10.1016/j.ejor.2012.11.029

Wright, T.P. (1936). Factors affecting the cost of airplanes. *Journal of the aeronautical sciences,* 3, 122-128. https://doi.org/10.2514/8.155

Yusriski, R., Sukoyo, S., Samadhi, T.A., & Halim, A.H. (2015). Integer batch scheduling problems for a single-machine with simultaneous effects of learning and forgetting to minimize total actual flow time. *International Journal of Industrial Engineering Computations,* 6, 365. https://doi.org/10.5267/j.ijiec.2015.2.005

Appendix

A.1. Proof of Proposition 1: The size of the closest batch to the due date is larger than any other batches in the schedule

Let there be a flow shop batch schedule as shown in Fig. A.1, showing batch 1, which is scheduled to be completed at the due date, and batch *i*, which is scheduled before batch 1. Each batch will be processed in machine 1 and machine *c*. Assume that one operator is assigned for each machine. The value of objective function *F* in Equation (17) can be rewritten as in Equation (A.1).

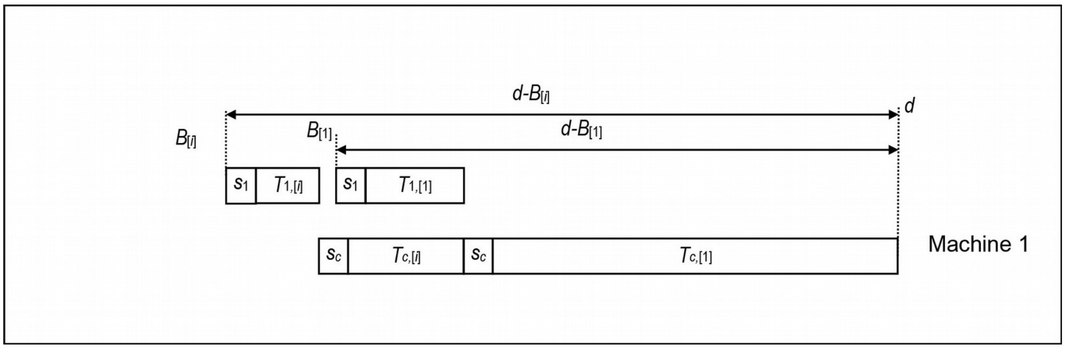

Fig. A.1. A flow shop batch schedule

$$F = \left(s_c + T_{c,[1]} + s_1 + T_{1,[1]}\right)Q_{[1]} + \left(2s_c + T_{c,[1]} + T_{c,[i]} + s_1 + T_{1,[i]}\right)Q_{[i]} \tag{A.1}$$

Using Equation (23) and Equation (24) to determine $T_{m,[j]}$ ($m \in \{1, c\}, j \in \{1, i\}$), we obtain Equation (A.2):

$$\begin{aligned} F = & \left(s_c + \frac{t_c\left[\left(\alpha_c + Q_{[1]}\right)^{1-\ell_c} - \alpha_c^{1-\ell_c}\right]}{1-\ell_c} + s_1 + \frac{t_1\left[\left(\alpha_1 + Q_{[1]}\right)^{1-\ell_1} - \alpha_1^{1-\ell_1}\right]}{1-\ell_1}\right)Q_{[1]} \\ & + \left(2s_c + \frac{t_c\left[\left(\alpha_c + Q_{[1]}\right)^{1-\ell_c} - \alpha_c^{1-\ell_c}\right]}{1-\ell_c} + \frac{t_c Q_{[i]}^{1-\ell_c}}{1-\ell_c} + s_1 + \frac{t_1 Q_{[i]}^{1-\ell_1}}{1-\ell_1}\right)Q_{[i]} \end{aligned} \tag{A.2}$$

where ℓ_m is the learning rate of operator assigned to machine m ($m \in \{1, c\}$).

Let us denote F_A as the objective function value when $Q_{[1]}$ is much bigger than $Q_{[i]}$, i.e. $Q_{[1]} = v$ and $Q_{[i]} = 0$ (v is the number of parts in batch 1 and batch i). Please remind that if $Q_{[i]} = 0$ (no learning during batch i), then no learning experience will be retained ($a_m = 0$, $m \in \{1, c\}$). Based on Equation (A.2), F_A can be determined in Equation (A.3).

$$F_A = \left(s_c + \frac{t_c v^{1-\ell_c}}{1-\ell_c} + s_1 + \frac{t_1 v^{1-\ell_1}}{1-\ell_1}\right)v \tag{A.3}$$

Let us also denote F_B as the objective function value when $Q_{[1]}$ is much smaller than $Q_{[i]}$, i.e. $Q_{[1]} = 0$ and $Q_{[i]} = v$. The value of F_B is given in Equation (A.4).

$$F_B = \left(2s_c + \frac{t_c v^{1-\ell_c}}{1-\ell_c} + s_1 + \frac{t_1 v^{1-\ell_1}}{1-\ell_1}\right)v \tag{A.4}$$

The difference between F_A and F_B is given in Equation (A.5).

$$F_A - F_B = -s_c v \tag{A.5}$$

Since s_c and v are positive, then $F_A - F_B$ is negative, or $F_A < F_B$. This proves that the objective function F is lower when batch 1 is larger than any other batches in the schedule.

A.2. Proof of Proposition 2: The faster operators learn, the larger the difference between the closest batch to the due date and any other batches in the schedule

Consider again Equation (A.3), where F_A is the objective function value when $Q_{[1]}$ is bigger than $Q_{[i]}$ by v. The value of $\partial F_A/\partial v$ is given in Equation (A.6).

$$\frac{\partial F_A}{\partial v} = s_c + s_1 + \frac{\left(2-\ell_c\right)t_c v^{1-\ell_c}}{1-\ell_c} + \frac{\left(2-\ell_1\right)t_1 v^{1-\ell_1}}{1-\ell_1} \tag{A.6}$$

The optimal value of v occurs when $\partial F_A / \partial v = 0$. We need to find v at minimum and maximum learning rates, i.e. 0 and 0.5 respectively (the maximum observable learning rate is 0.514, rounded to 0.5 to simplify the computation). Setting $\partial F_A / \partial v = 0$ and the learning rates to 0.5, we can rewrite Equation (A.6) as shown in Equation (A.7):

$$\frac{1.5 t_c v^{0.5}}{0.5} + \frac{1.5 t_1 v^{0.5}}{0.5} = -s_c - s_1 \tag{A.7}$$

which after some algebraic operations can be used to find v as shown in Equation (A.8).

$$v = \left[\frac{-s_c - s_1}{3(t_c + t_1)}\right]^2 \tag{A.8}$$

When the learning rates are zero, we substitute the learning rates in Equation (A.6) with zero and set $\partial F_A / \partial v = 0$ to obtain Equation (A.9) which can be used to find v' (i.e. v at zero learning rates) as shown in Equation (A.10).

$$1.5 t_c v' + 1.5 t_1 v' = -s_c - s_1 \tag{A.9}$$

$$v' = \frac{-s_c - s_1}{1.5(t_c + t_1)} \tag{A.10}$$

The difference between v and v' is given in Equation (A.11):

$$v - v' = \left[\frac{-s_c - s_1}{3(t_c + t_1)}\right]^2 - \frac{-s_c - s_1}{1.5(t_c + t_1)} = \frac{(-s_c - s_1)^2 + 6(t_c + t_1)(s_c + s_1)}{9(t_c + t_1)^2} \tag{A.11}$$

which is always positive, meaning that the difference between $Q_{[1]}$ and $Q_{[i]}$ is larger when operators learn faster.

A.3. Proof of Proposition 3: The faster operators learn, the lower number of batches in the optimal solution

Consider a schedule of batch i ($i = 1, \ldots, N$). Batch 1, the closest batch to the due date, is larger than any other batches in the schedule (Proposition 1), and the difference v between $Q_{[1]}$ and $Q_{[i]}$ ($i > 1$) is larger when operators learn faster (Proposition 2). This means that the faster operators learn, the more parts will be processed in batch 1, and in turn, this will cause one or more batches (other than batch 1) to have zero size, bringing a lower number of batches in the optimal solution.

A.4. Proof of Proposition 4: The faster operators learn, the lower the value of the objective function F and $Q_{[1]}$ will be larger than $Q_{[i]}$ not more by $[(1 - \ell_c)(1 - \ell_1)(t_c + t_1)]^{\ell_c} / [t_c(1 - \ell_c) + t_1(1 - \ell_c)]^{\ell_c}$

Consider again Equation (A.2) showing the objective function F for the flow shop batch schedule shown in Fig. A.1. Decreasing the learning rates ℓ_m ($m \in \{1, c\}$) to zero will change F to F', i.e. the objective function at zero learning rates, as given in Equation (A.12).

$$F' = (s_c + t_c Q_{[1]} + s_1 + t_1 Q_{[1]})Q_{[1]} + (2s_c + t_c Q_{[1]} + t_c Q_{[i]} + s_1 + t_1 Q_{[i]})Q_{[i]} \tag{A.12}$$

The difference between F and F' is given in Equation (A.13):

$$\begin{aligned} F' - F = {} & \left(t_c\left[(\alpha_c + Q_1) - \alpha_c\right] + t_1\left[(\alpha_1 + Q_1) - \alpha_1\right] - \frac{t_c\left[(\alpha_c + Q_1)^{1-\ell_c} - \alpha_c^{1-\ell_c}\right]}{1-\ell_c} - \frac{t_1\left[(\alpha_1 + Q_1)^{1-\ell_1} - \alpha_1^{1-\ell_1}\right]}{1-\ell_1} \right) Q_{[1]} \\ & + \left(t_c\left[(\alpha_c + Q_1) - \alpha_c\right] + t_c Q_{[i]} + t_1 Q_{[i]} - \frac{t_c\left[(\alpha_c + Q_1)^{1-\ell_c} - \alpha_c^{1-\ell_c}\right]}{1-\ell_c} - \frac{t_c Q_{[i]}{}^{1-\ell_c}}{1-\ell_c} - \frac{t_1 Q_{[i]}{}^{1-\ell_1}}{1-\ell_1} \right) Q_{[i]}, \end{aligned} \tag{A.13}$$

which can be rewritten in Equation (A.14):

$$F'-F=\left(\frac{Q_{[1]}(1-\ell_c)(1-\ell_1)(t_c+t_1)-t_c(1-\ell_1)\left[(\alpha_c+Q_1)^{1-\ell_c}-\alpha_c^{1-\ell_c}\right]-t_1(1-\ell_c)\left[(\alpha_1+Q_1)^{1-\ell_1}-\alpha_1^{1-\ell_1}\right]}{(1-\ell_c)(1-\ell_1)}\right)Q_{[1]}$$
$$+\left(\frac{(1-\ell_c)(1-\ell_1)\left[t_cQ_{[1]}+Q_{[i]}(t_c+t_1)\right]-t_c(1-\ell_1)\left[(\alpha_c+Q_1)^{1-\ell_c}-\alpha_c^{1-\ell_c}\right]-t_c(1-\ell_1)Q_{[i]}{}^{1-\ell_c}-t_1(1-\ell_c)Q_{[i]}{}^{1-\ell_1}}{(1-\ell_c)(1-\ell_1)}\right)Q_{[i]} \tag{A.14}$$

Proposition 4 is proven if $F' > F$, or if $F'–F$ is positive. Since denominators in Equation (A.14) are positive, the proof requires numerators in Equation (A.14) to be also positive, or:

$$\begin{aligned}&Q_{[1]}{}^2(1-\ell_c)(1-\ell_1)(t_c+t_1)-t_cQ_{[1]}(1-\ell_1)\left[(\alpha_c+Q_1)^{1-\ell_c}-\alpha_c^{1-\ell_c}\right]\\&-t_1Q_{[1]}(1-\ell_c)\left[(\alpha_1+Q_1)^{1-\ell_1}-\alpha_1^{1-\ell_1}\right]+Q_{[i]}(1-\ell_c)(1-\ell_1)\left[t_cQ_{[1]}+Q_{[i]}(t_c+t_1)\right]\\&-t_cQ_{[i]}(1-\ell_1)\left[(\alpha_c+Q_1)^{1-\ell_c}-\alpha_c^{1-\ell_c}\right]-t_c(1-\ell_1)Q_{[i]}{}^{2-\ell_c}-t_1(1-\ell_c)Q_{[i]}{}^{2-\ell_1}>0\end{aligned} \tag{A.15}$$

Proposition 2 states that the faster operators learn, the larger the difference between the closest batch to the due date and any other batches in the schedule. The situation in Proposition 2 also applies in Proposition 4, meaning that faster learning rates increase the difference between the closest batch to the due date and any other batches, and decrease the objective function simultaneously. Let Q[1] be bigger than Q[i] by v, so we replace Q[1] and Q[i] in Expression (A.15) with v and 0 respectively to obtain Expression (A.16).

$$v^2(1-\ell_c)(1-\ell_1)(t_c+t_1)-t_cv^{2-\ell_c}(1-\ell_1)-t_1v^{2-\ell_1}(1-\ell_c)>0 \tag{A.16}$$

which can be used to find v in Expression (A.17):

$$v<\left[\frac{(1-\ell_c)(1-\ell_1)(t_c+t_1)}{t_c(1-\ell_1)+t_1(1-\ell_c)}\right]^{\ell_c} \tag{A.17}$$

Now it is proven that the faster operators learn, the lower the value of the objective function F, and $Q_{[1]}$ will be bigger than $Q_{[i]}$ by not more than v as given in Expression (A.17).