A Flow Shop Batch Scheduling Model with Part Deterioration and Operator Learning-Forgetting Effects to Minimize Total Actual Flow Time

Dwi Kurniawan
Industrial Engineering Department
Institut Teknologi Nasional
Bandung 40124, Indonesia
dwi_kurniawan@itenas.ac.id

Rinto Yusriski
Industrial Engineering Department
Universitas Jenderal Achmad Yani
Bandung 40285, Indonesia
rinto.yusriski@lecture.unjani.ac.id

Mohammad Mi'radj Isnaini[1], Anas Ma'ruf[2], Abdul Hakim Halim[3]
Industrial Engineering Department
Institut Teknologi Bandung
Bandung 40132, Indonesia
[1]isnaini@itb.ac.id, [2]maruf@itb.ac.id, [3]ahakimhalim@itb.ac.id

Abstract

This paper proposes a batch scheduling model where two time-changing effects simultaneously occur, i.e., parts deteriorating and operator learning-forgetting effects. A mathematical model is proposed for the problem where the decision variables are the number of batches, batch sizes, and a schedule of the resulting batches to minimize total actual flow time. A proposed algorithm is developed by trying different numbers of batches, starting from one and then increasing it one by one until the objective function value does not improve anymore. Numerical examples show that the faster operators learn, the lower the optimal number of batches and the lower the optimal total actual flow time. Second, the faster parts deteriorate, the higher the optimal number of batches and the higher the optimal total actual flow time. Finally, the model divides parts into small batches to prevent the processing time from increasing due to part deterioration.

1. Introduction

This paper develops a flow shop batch scheduling model where processing time changes over time caused by the occurrence of time-changing effects (Strusevich and Rustogi, 2017). Two types of time-changing effects can occur, i.e., a learning effect causing the processing time to decrease, and part deterioration causing the processing time to increase. A learning effect occurs when an operator operates the same task repeatedly, and when the operation is interrupted, a forgetting effect takes place directly (Yusriski et al., 2015b). The learning effect was conceptualized in the first place by Wright (1936) who discovered that the processing time of a production process reduces by a constant proportion when the total number of produced units doubles. Jaber and Bonney (1996) proposed a mathematical model for learning and forgetting that occur alternately from a production lot to the next. Nembhard and Uzumeri (2000) highlighted that understanding the learning effect is essential in industries with manual operation to set standard times, estimating labour costs and scheduling. After conducting an experiment in a car factory, Sebrina et al. (2011) identified that learning rate is influenced by some factors including product complexity and takt time, and that a higher learning rate can cause the defect rate to increase.

Part deterioration can occur, for example, in a steel slab rolling process to produce a steel plate or coil (Cheng et al., 2004). The rolling process is performed after a preheating process for a particular time affected by the initial temperature of the slab. When the rolling process is performed, the slab temperature naturally decreases, causing the rolling time to be longer. This situation shows that the processing time is longer when the process starts at a later time. This phenomenon is defined as deteriorating jobs (Wang and Xia, 2006, Jeng and Lin, 2004).

The literature shows that batch scheduling models with learning and forgetting effects have been proposed in Yusriski et al. (2015b) for single-stage systems and in Kurniawan et al. (2020) for flow shops. However, these papers did not consider part deterioration. On the contrary, Sukoyo et al. (2010) proposed a multi-item batch scheduling model considering part deterioration, but did not consider learning and forgetting effects, and assumed a uniform processing time of parts in a batch. Batch production is a common practice in industries (Baker and Jia, 1993) as it gives a better makespan, flow time and inventory level (Kalir and Sarin, 2000). There are two problems in batch scheduling, i.e., finding the optimal number of batches and batch sizes, and scheduling the resulting batches (Halim et al., 1994). Additionally, actual flow time is an important performance criterion to ensure due date fulfilment and to minimize the amount of inventory in the shop floor (Halim et al., 1994). It is necessary to integrate the batch scheduling model with learning and forgetting effects in Kurniawan et al. (2020) and the model considering part deterioration in Sukoyo et al. (2010), and consider the fluctuation of processing time among parts. The impact of different learning and deterioration rates to batch schedules is also necessary to investigate.

2. Literature Review

According to Baker (1974), scheduling is the allocation of resources to perform a set of tasks over time. Fogarty et al. (1991) more simply defined scheduling as scheduling activities, both Master Production Schedule (MPS), schedules on the factory floor, maintenance schedules, and so on. Meanwhile, Morton and Pentico (1993) defined scheduling as a process for organizing, selecting, and determining the time of resource usage in producing a number of outputs at a certain time by meeting the constraints of time and resource availability.

Fogarty et al. (1991) stated that scheduling can be divided into job scheduling and batch scheduling. Job scheduling is performed on a number of jobs with known sizes, while batch scheduling is performed by dividing parts into several groups and performing one set up for each group. Scheduling can be done on a flow shop production system, which is a system that processes products with a uniform process sequence, or on a job shop system, which is a system that processes products with a varied process sequence.

Batch scheduling models that consider machines as the resource have been developed for single-stage systems to minimize costs (Wang et al., 2016b), makespan (Liang and Hui, 2016), flow time (Yin et al., 2014, Ji et al., 2015), actual flow time (Arizono et al., 1992, Yusriski et al., 2015a, Halim et al., 1994), the number of tardy jobs (Li et al., 2015, Parsa et al., 2017), and total tardiness (Wang et al., 2016a). Flow shop batch scheduling models have also been proposed to minimize makespan (Arroyo and Leung, 2017), flow time (Bukchin et al., 2002), and actual flow time Halim and Ohta (1993) dan Hidayat et al. (2016). Additionally, job shop batch scheduling models have been discussed, such as in Mosheiov and Oron (2008) to minimize makespan and flow time.

The batch scheduling models to minimize total actual flow time have been studied such as Yusriski et al. (2015a), Hidayat et al. (2016), Maulidya et al. (2018) dan Yusriski et al. (2019), and have been proven to be effective in producing schedules that meet due dates and minimize inventory levels. The models assumed that processing times are fixed, while in practical situations, processing times can change over time due to learning and forgetting (Wright, 1936, Jaber and Bonney, 1996) and part deterioration (Cheng et al., 2004, Sukoyo et al., 2010). The batch scheduling model needs to consider the time-changing effect due to the learning-forgetting and part deterioration, so that the resulting schedule is more accurate and easier to implement in industries.

3. Model Development

Indices, parameters and variables used in this paper are shown as follows.

Indices:

m = index of machines ($m = 1, \ldots, k$),
i = index of batches, sequenced backward from the due date ($i = 1, \ldots, N$).

Parameters:

n = number of parts to be processed,
k = number of machines,
d = due date, calculated from $t = 0$,
s_m = setup time per batch at machine m,
t_m = part processing time at machine m before learning, forgetting and part deterioration,
δ = part deterioration rate,
ℓ_m = learning gradient of operator at machine m.

Variables:

F = total actual flow time of all parts,
N = number of batches,
$Q_{[i]}$ = number of parts in batch i,
$B_{m,[i]}$ = starting time of the batch i at machine m,
$f_{m,[i]}$ = forgetting gradient of batch i at machine m,
$\alpha_{m,[i]}$ = equivalent number of parts of retained learning experience at the beginning of batch i at machine m,
$\beta_{m,[i]}$ = equivalent number of parts of accumulated part deterioration at the beginning of batch i at machine m,
$I_{m,[i]}$ = length of process interruption between batch i and *batch* i+1 at machine m,
$J_{m,[i]}$ = length of process interruption for batch i between machine m–1 and machine m,
$t_{[p]}$ = processing time of the x-th part as a learning function,
$\hat{t}_{[x]}$ = processing time of the x-th as a forgetting function,
$T_{m,[i]}$ = processing time of all parts in batch i at machine m.

In manufacturing systems, a learning effect occurs when an operator performs a particular production process repeatedly, resulting a decreasing part processing time as the number of repetition increases (Jaber and Bonney, 1996). According to Wright (1936), the learning effect occurs following a learning function shown in Eq. (1):

$$L(x) = tx^{-\ell} \tag{1}$$

where t is processing time before learning, x is the part sequence and ℓ is a learning gradient of 0 or more (a higher ℓ means a faster learning, $\ell = 0$ means no learning).

Meanwhile, part deterioration occurs because the material changes over time, causing the processing time to be longer when the process starts at a later time. We assume a linear deterioration, i.e., the processing time of a part increases constantly by proportion δ of the initial processing time, resulting from a deterioration function in Eq. (2):

$$D(x) = t(1 + \delta x) \tag{2}$$

where t is an initial processing time (before deterioration), x is the part sequence and δ is a deterioration rate of 0 or more (a higher δ means a faster deterioration, $\delta = 0$ means no deterioration).

When the production process is performed, operator learning and part deterioration coincide. The processing time can be written as $L(x)$ with deterioration, $L(x)(1+\delta x)$, or $D(x)$ with learning, $D(x)x^{-\ell}$. Both formulas result in function $t_{[x]}$ written in Eq. (3).

$$t_{[x]} = t(1+\delta x)x^{-\ell} \tag{3}$$

Figure 1 shows the fluctuation of processing time based on Eq. (3). The processing time of the x-th part decreases to a minimum point at the $x_{\min}$-th part, then increases again, and reaches its initial value at the r-th part. Before $x_{\min}$, learning is more dominant than deterioration, while after $x_{\min}$, deterioration is more dominant. Theorem 1 and 2 postulate when the processing time reaches a minimum and reaches its initial value again.

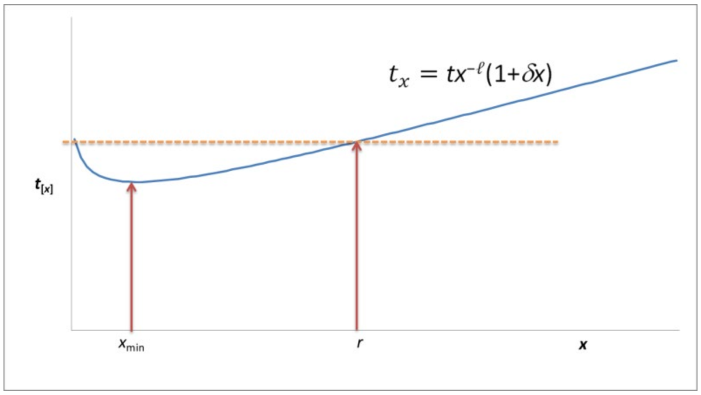

Figure 1. Fluctuation of processing time along parts

Theorem 1. Function $t_{[x]} = t(1+\delta x)x^{-\ell}$ reaches a minimum at the $x_{\min}$-th part, where

$$x_{\min} = \frac{\ell}{\delta(1-\ell)}.$$

Proof. Function $t_{[x]} = t(1+\delta x)x^{-\ell}$ reaches minimum when $t'_{[x]} = t\delta x^{-\ell} - t\ell x^{-1-\ell}(1+\delta x) = 0$ or $t\delta x^{-\ell} = t\ell x^{-1-\ell}(1+\delta x)$, which results $x = \dfrac{\ell}{\delta(1-\ell)}$, as notated in this theorem.■

Theorem 2. After reaching minimum, function $t_{[x]} = t(1+\delta x)x^{-\ell}$ increases and reaches t again at the r-th part, where r is the bigger root of polynom $r^{\ell} - \delta r = 1$.

Proof. The situation in this theorem is given by $t_{[x]} = t(1+\delta x)x^{-\ell} = t$, which can be rewritten as $r^{-\ell}(1+\delta r) = 1$. After some algebraic operations we get $r^{\ell} - \delta r = 1$, which can be solved when ℓ and δ are known. We take the bigger root as r, since the smaller root is usually negative. ■

Figure 2 shows learning-forgetting and part deterioration in a three-machine, three-batch system. Parts arrive at the beginning of each batch at machine 1 in undeteriorated condition, and the deterioration occurs continuously from machines 1 to 2 and 3. Meanwhile, an operator at each machine starts learning at the beginning of batch 3 at the machine, and experiences forgetting during process interruptions between two consecutive batches. Forgetting starts directly when the learning effect stops and takes place during setups and machine idle times (Kurniawan et al., 2020).

Figure 3 shows a more detailed analysis of the processing time fluctuation in a batch schedule. Suppose that the processing time before part deterioration and operator learning is t, and suppose that the process of batch i is started when the operator has had $\alpha_{[i]}$ unit-equivalent of the learning experience ($\alpha_{[i+1]} = 0$ at A). Learning and deterioration coincide along AC curve for $T_{m,[i+1]}$, following the function in Eq. (3). When the process is interrupted, forgetting occurs along CE curve during interval I, following a forgetting function in Eq. (4) that imaginarily started at B. Suppose that $Q_{[i+1]}$ parts are processed during $T_{m,[i+1]}$, and suppose that R parts can be processed if the process is not interrupted during interval I, and learning will continue along CD curve following the learning function in Eq. (1).

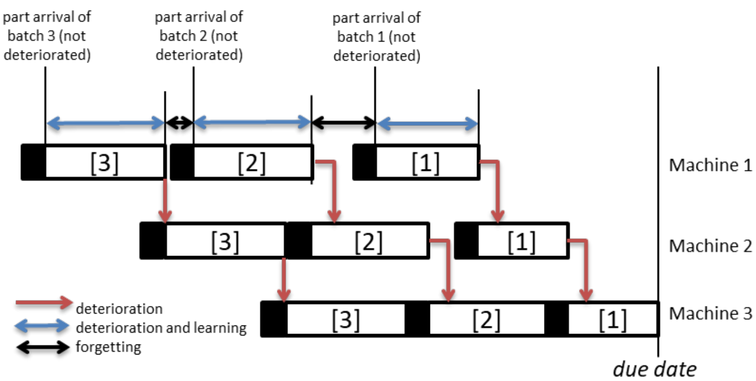

Figure 2. Learning-forgetting and part deterioration in a backward schedule

$$\hat{t}_{[x]} = \hat{t}x^{f} \tag{4}$$

At C, the value of $\hat{t}_{[x]}$ is equal to $t_{[x]}$, or $t(1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))(\alpha_{[i+1]}+Q_{[i+1]})^{-\ell} = \hat{t}(\alpha_{[i+1]}+Q_{[i+1]})^{f}$, which can be used to find $\hat{t}$, an imaginary processing time at B where $\hat{t}_{[x]}$ starts, i.e.:

$$\hat{t} = t(1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))(\alpha_{[i+1]}+Q_{[i+1]})^{-(f+\ell)}. \tag{5}$$

Substituting Eq. (5) to Eq. (4) we obtain:

$$\hat{t}_{[x]} = t(1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))(\alpha_{[i+1]}+Q_{[i+1]})^{-(f+\ell)}x^{f}. \tag{6}$$

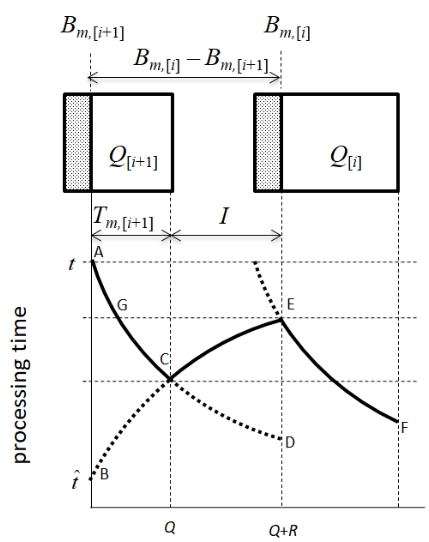

Figure 3. Learning and forgetting effects during two sequential batches

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

$$I = \int_{\alpha_{[i+1]}+Q_{[i+1]}}^{\alpha_{[i+1]}+Q_{[i+1]}+R} tx^{-\ell}dx = \frac{t}{1-\ell}\left[\left(\alpha_{[i+1]}+Q_{[i+1]}+R\right)^{1-\ell} - \left(\alpha_{[i+1]}+Q_{[i+1]}\right)^{1-\ell}\right] \tag{7}$$

After some algebraic operations on Eq. (7) to find $\alpha_{[i+1]}+Q_{[i+1]}+R$ we obtain Eq. (8).

$$\alpha_{[i+1]}+Q_{[i+1]}+R = \left[\frac{I(1-\ell)}{t} + \left(\alpha_{[i+1]}+Q_{[i+1]}\right)^{1-\ell}\right]^{1/(1-\ell)} \tag{8}$$

After forgetting during CE, $\alpha_{[i+1]}+Q_{[i+1]}$ unit-equivalent of operator learning experience obtained during AC will be reduced to $\alpha_{[i]}$ units. The value of $\alpha_{[i]}$ can be found by equating $\hat{t}_{[x]}$ at E (i.e. $\hat{t}_{[\alpha_{[i+1]}+Q_{[i+1]}+R]}$) with $t_{[x]}$ ending at E, i.e. $t\alpha_{[i]}^{-\ell} = t(1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))(\alpha_{[i+1]}+Q_{[i+1]})^{-(\ell+f)}\left(\alpha_{[i+1]}+Q_{[i+1]}+R\right)^{f}$, resulting Eq. (9).

$$\alpha_{[i]} = (1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))(\alpha_{[i+1]}+Q_{[i+1]})^{-(\ell+f)/\ell}\left(\alpha_{[i+1]}+Q_{[i+1]}+R\right)^{-f/\ell} \tag{9}$$

Substituting Equation (8) to Equation (9) and after some algebraic operations we obtain Eq. (10).

$$\alpha_{[i]} = (1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))\left[\alpha_{[i+1]}+Q_{[i+1]}\right]^{\frac{\ell+f}{\ell}}\left[\frac{I(1-\ell)}{t} + \left(\alpha_{[i+1]}+Q_{[i+1]}\right)^{1-\ell}\right]^{\frac{-f}{\ell(1-\ell)}} \tag{10}$$

According to Figure 3, the value of I is given by $I = B_{m,[i]} - B_{m,[i+1]} - T_{m,[i+1]}$.

We need to remind that Eq. (10) applies when the operator experiences a partial forgetting after batch i+1, or when $T_E < t$. If the operator experiences a total forgetting, then $T_E = t$ and $\alpha_{[i]} = 0$. Therefore, Eq. (10) needs to be rewritten as in Eq. (11).

$$\alpha_{[i]} = \max\left(0, (1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))\left[\alpha_{[i+1]}+Q_{[i+1]}\right]^{\frac{\ell+f}{\ell}}\left[\frac{I(1-\ell)}{t} + \left(\alpha_{[i+1]}+Q_{[i+1]}\right)^{1-\ell}\right]^{\frac{-f}{\ell(1-\ell)}}\right) \tag{11}$$

The forgetting gradient f is computed at the beginning of total forgetting, i.e., when the value of $\hat{t}_{[x]} = t$. This situation is given by $t(1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))(\alpha_{[i+1]}+Q_{[i+1]})^{-(\ell+f)}\left(\alpha_{[i+1]}+Q_{[i+1]}+R\right)^{f} = t$, which can be used to find f:

$$f = \frac{\ell\ln(\alpha_{[i+1]}+Q_{[i+1]}) - \ln(1+\delta(\alpha_{[i+1]}+Q_{[i+1]}))}{\frac{1}{1-\ell}\ln\left[\frac{I}{t}(1-\ell) + \left(\alpha_{[i+1]}+Q_{[i+1]}\right)^{1-\ell}\right] - \ln(\alpha_{[i+1]}+Q_{[i+1]})}. \tag{12}$$

Similar to learning, we suppose that the process of batch i at machine m is started when parts have deteriorated for $\beta_{m,[i]}$ unit-equivalent ($\beta_{m,[i+1]} = 0$ at A) and completed at $\beta_{m,[i]} + Q_{[i]}$ unit-equivalent of deterioration. The length of process interruption for batch i between machine m–1 and machine m, $B_{m+1,[i]} - B_{m,[i]} - T_{m,[i]}$, can be computed by integrating function $D(x)$ in Eq. (2) from $\beta_{m,[i]}$ to $\beta_{m,[i]} + Q_{[i]}$, i.e.:

$$B_{m+1,[i]} - B_{m,[i]} - T_{m,[i]} = \int_{\beta_{m,[i]}+Q_{[i]}}^{\beta_{m,[i]}+Q_{[i]}+U_{m,[i]}} t_m(1+\delta_m x)dx = t_m\left(U_{m,[i]} + \delta\left(\frac{(\beta_{m,[i]}+Q_{[i]}+U_{m,[i]})^2}{2} - \frac{(\beta_{m,[i]}+Q_{[i]})^2}{2}\right)\right),$$

which can be used to obtain $\beta_{m,[i]} + Q_{[i]} + U_{m,[i]}$, the accumulated deterioration at the end of interruption as shown in Eq. (13).

$$\beta_{m,[i]} + Q_{[i]} + U_{m,[i]} = \frac{-t_m + \sqrt{t_m^{\ 2} + 2t_m\delta\left(t_m(\beta_{m,[i]} + Q_{[i]}) + \frac{t_m\delta(\beta_{m,[i]}+Q_{[i]})^2}{2} + B_{m+1,[i]} - B_{m,[i]} - T_{m,[i]}\right)}}{t_m\delta} \tag{13}$$

The accumulated part deterioration in Eq. (13) is expressed in unit-equivalent of machine *m*. When parts at that time point are transferred to machine *m*+1, the accumulated part depreciation becomes $\beta_{m+1,[i]} = t_m/t_{m+1}\left(\beta_{m,[i]} + Q_{[i]} + U_{m,[i]}\right)$, or:

$$\beta_{m+1,[i]} = \frac{-t_m + \sqrt{t_m^{\ 2} + 2t_m\delta\left(t_m(\beta_{m,[i]} + Q_{[i]}) + \frac{t_m\delta(\beta_{m,[i]}+Q_{[i]})^2}{2} + B_{m+1,[i]} - B_{m,[i]} - T_{m,[i]}\right)}}{t_{m+1}\delta}. \tag{14}$$

Since part deterioration and operator learning occur differently, expression $t_{[x]} = t(1+\delta x)x^{-\ell}$ in Eq. (3) needs to be rewritten into $t_{[x,y]} = t(1+\delta x)y^{-\ell}$. Thus, the batch processing time $T_{m,[i]}$ is computed by integrating $t_{[x,y]}$ along the intervals of deterioration and learning, i.e., $T_{m,[i]} = \int_{\alpha_{[i]}}^{\alpha_{[i]}+Q_{[i]}} \int_{\beta_{m,[i]}}^{\beta_{m,[i]}+Q_{[i]}} t(1+\delta x)y^{-\ell_m} dxdy$, which results Eq. (15).

$$T_{m,[i]} = t\frac{\left(\alpha_{[i]} + Q_{[i]}\right)^{1-\ell} - \alpha_{[i]}^{1-\ell}}{1-\ell}\left(Q_{[i]} + \frac{\delta\left(Q_{[i]}^{\ 2} + 2Q_{[i]}\beta_{m,[i]}\right)}{2}\right) \tag{15}$$

4. Model and Algorithm

The problem investigated in this paper can be explained as follows. There are *n* parts that will be processed in *N* batches, and each batch *i* will be processed in *k* machines with a uniform routing. Each operation requires setup times s_m and initial processing times t_m. Operator at machine *m* learns at a learning gradient ℓ_m, and parts deteriorate at a deterioration rate δ. All operations must be finished no later than a due date *d*. The objective function in the model is the total actual flow time *F*, and the decision variables are the number of batches *N*, batch sizes $Q_{[i]}$, and the schedule of batch *i* at machine *m* ($B_{m,[i]}$). Assumptions used in this study are all parts and machines can be scheduled at $t = 0$, machines are always available during the scheduling horizon, parts arrive at the shop floor in undeteriorated condition, and operators have no prior learning experience.

Based on the analysis explained in Section 2, the flow shop batch scheduling model with part deterioration and operator learning-forgetting to minimize actual flow time is formulated in Model 1 as follows.

Model 1.

Minimize
$$F = \sum_{i=1}^{N} \left(d - B_{1,[i]}\right) Q_{[i]} \tag{16}$$

subject to

$$B_{k,[i]} = d - (i-1)s_k + \sum_{j=1}^{i} T_{k,[j]}, \quad \forall i \tag{17}$$

$$B_{m,[1]} = B_{m+1,[1]} - T_{m,[1]}, \quad m < k \tag{18}$$

$$B_{m,[i]} \le B_{m,[i-1]} - s_m - T_{m,[i]}, \quad m < k,\ i > 1 \tag{19}$$

$$B_{m,[i]} \le B_{m+1,[i]} - T_{m,[i]}, \quad m < k,\ i > 1 \tag{20}$$

$$B_{1,[N]} \ge 0 \tag{21}$$

$$T_{m,[i]} = t_m \frac{\left(\alpha_{m,[i]} + Q_{[i]}\right)^{1-\ell_m} - \alpha_{m,[i]}^{1-\ell_m}}{1-\ell_m} \left(Q_{[i]} + \frac{\delta\left(Q_{[i]}^2 + 2Q_{[i]}\beta_{m,[i]}\right)}{2} \right), \tag{22}$$

$$\alpha_{m,[i]} = \begin{cases} 0, & \forall m, i = N, \ell_m = 0 \\ \max\left(0, (1+\delta(\alpha_{m,[i+1]} + Q_{[i+1]}))\left[\alpha_{m,[i+1]} + Q_{[i+1]}\right]^{\frac{\ell_m + f_{m,[i]}}{\ell_m}} \left[\frac{I_{m,[i]}}{t_m}(1-\ell_m) + \left(\alpha_{m,[i+1]} + Q_{[i+1]}\right)^{1-\ell_m} \right]^{\frac{-f_{m,[i]}}{\ell_m(1-\ell_m)}} \right), & \forall m, i < N, \ell_m > 0 \end{cases} \tag{23}$$

$$f_{m,[i]} = \frac{\ell_m \ln(\alpha_{m,[i+1]} + Q_{[i+1]}) - \ln(1 + \delta(\alpha_{m,[i+1]} + Q_{[i+1]}))}{\frac{1}{1-\ell_m} \ln\left[\frac{I_{m,[i]}}{t_m}\left(1-\ell_m\right) + \left(\alpha_{m,[i+1]} + Q_{[i+1]}\right)^{1-\ell_m} \right] - \ln(\alpha_{m,[i+1]} + Q_{[i+1]})}, \quad \forall m, i < N, \ell_m > 0 \tag{24}$$

$$\beta_{m,[i]} = \begin{cases} 0, & m = 1, \forall i, \delta = 0 \\ \dfrac{-t_{m-1} + \sqrt{t_{m-1}{}^2 + 2t_{m-1}\delta\left(t_{m-1}(\beta_{m-1,[i]} + Q_{[i]}) + \dfrac{t_{m-1}\delta(\beta_{m-1,[i]} + Q_{[i]})^2}{2} + J_{m,[i]} \right)}}{t_m \delta}, & m > 1, \forall i, \delta > 0 \end{cases} \tag{25}$$

$$I_{m,[i]} = B_{m,[i]} - B_{m,[i+1]} - T_{m,[i+1]}, \quad \forall m,\ i < N \tag{26}$$

$$J_{m,[i]} = B_{m,[i]} - B_{m-1,[i]} - T_{m-1,[i]}, \quad m > 1,\ \forall i \tag{27}$$

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

$$Q_{[i]} > 0,\ 1 \le N \le n, \quad i = 1, \ldots, N. \tag{29}$$

Objective function (16) is to minimize the total actual flow time of all parts, i.e., the total time spent by all parts from the batch arrival at machine 1 to the due date. The schedule of each batch at each machine is computed in constraints (17) to (21). Constraint (17) computes the schedule of batches at machine k, constraint (18) computes the schedule of batch 1 at machine $m < k$, and constraint (19) and (20) state that the schedule of batch $i > 1$ at machine $m < k$. Constraint (21) states that the schedule of batch N at machine 1 must not be earlier than time zero. Constraint (22) defines batch processing times used in constraint (17) to (20). Constraint (23) computes the equivalent number of parts of the retained learning experience at the beginning of batch i at machine m, and constraint (24) calculates the forgetting gradient. Constraint (25) defines the equivalent number of parts of accumulated part deterioration at the beginning of batch i at machine m. The length of process interruption for learning and deterioration are computed in constraint (26) and (27) respectively. Constraint (22) to (25) are based on Eq. (11) to (15) explained in Section 2. Constraint (28) ensures that the number of parts in all batches must be equal to the total number of parts. Finally, constraint (29) states that batch sizes must be positive, and that the possible number of batches is from one to the number of parts.

The problem formulated in Model 1 is unsolvable since N, the upper limit of a sum in the objective function is unknown. We need to relax N from a decision variable to a parameter by setting a value for N. The optimal solution for the problem can be found by trying all possible N values $(1 \leq N \leq n)$ and finding the best objective value. However, this procedure is not efficient as the computation time grows quickly when N increases. Therefore, we follow a procedure used in Bukchin et al. (2002), where several N values are tried, starting from one, and increasing N one-by-one until the value of F stops improving. The best F value at all tried N values is set as the optimal solution. Algorithm 1 explains the solution method to solve Model 1.

Algorithm 1.
Step 1. Set parameters n, k, s_m, t_m, ℓ_m, δ and d. Go to Step 2.
Step 2. Set $N = 1$. Go to Step 3.
Step 3. Solve Model 1, find 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. Otherwise, go to Step 6.
Step 5. If $N = n$, go to Step 6. Otherwise, set $N = N + 1$, return to Step 3.
Step 6. Stop. The current F^* is the optimal solution.

Please remind that Step 3 in Algorithm 1 was performed using Lingo, as it is a comprehensive optimization tool, capable of solving linear or non-linear models efficiently (Goodarzi et al., 2014).

5. Results and Discussion

The proposed model and algorithm were applied to 16 data sets with 4 different δ values, from 0 to 0.015, and 4 different ℓ_m values, from 0 to 0.36. The other parameter values were set to equal for all data sets, i.e., $k = 3$, $n = 60$, $d = 10{,}000$, $\boldsymbol{s} = (52\ 50\ 54)$, and $\boldsymbol{t} = (3\ 5\ 4)$. Table 1 shows the optimal number of batches and the optimal objective function value for each data set. The results in Table 1 show that the faster operators learn, the lower the optimal number of batches and the lower the total actual flow time. This finding is consistent with Proposition 3 and 4 postulated in Kurniawan et al. (2020). However, Proposition 1 and 2 in Kurniawan et al. (2020) do not match with the finding in this research. The different finding between Kurniawan et al. (2020) and this research is understandable since in Kurniawan et al. (2020) did not consider part deterioration in their model. Second, the faster parts deteriorate, the higher the optimal number of batches and the total actual flow time. This is because the faster parts deteriorate, the faster the processing time increases, so the value of the objective function is higher.

Table 2 shows a calculation example using Algorithm 1 for Data Set 12. We tried some N values, starting from $N = 1$. Increasing N value one-by-one, the objective function value improved (decreased) until $N = 19$, and then it increased at $N = 20$. The calculation was then stopped and the objective function value at $N = 19$ was set as the optimal solution.

Table 1. The optimal solution of data sets

δ	$\boldsymbol{\ell} = (0\ 0\ 0)$			$\boldsymbol{\ell} = (0.11\ 0.13\ 0.15)$			$\boldsymbol{\ell} = (0.24\ 0.22\ 0.21)$			$\boldsymbol{\ell} = (0.33\ 0.31\ 0.36)$		
	Data Set	N^*	F^*	Data Set	N^*	F^*	Data Set	N^*	F^*	Data Set	N^*	F^*
0.000	1	23	59,895.2	2	21	56,859.0	3	20	54,792.2	4	19	52,559.9
0.005	5	23	60,366.3	6	21	57,034.9	7	20	54,979.5	8	19	52,726.8
0.010	9	23	60,832.2	10	21	57,144.5	11	20	55,129.1	12	19	52,860.6
0.015	13	23	61,294.9	14	21	57,156.9	15	21	55,233.9	16	20	52,957.1

Table 2. Solution searching for Data Set 12

N	1	2	3	4	5	6	7	8	9	10
F	1,508,331.0	438,890.8	233,007.9	155,167.8	116,711.0	94,828.9	80,413.3	72,484.4	66,539.9	62,348.6
N	11	12	13	14	15	16	17	18	19*	20
F	59,346.6	57,189.2	55,649.3	54,570.4	53,834.3	53,352.7	53,062.9	52,913.9	52,860.6*	52,863.7

* Optimal solution

Table 3 details the optimal solution of Data Set 12, showing the decision variables from batch 19 to batch 1 (note that batch 19 will be processed first), batch sizes $Q_{[i]}$, batch operation schedule, and processing time fluctuation ('first' and 'last' in the table mean the processing time of the first and last part in the batch). The detailed solution in

Table 3 shows us that, first, the optimal batch sizes are small. This occurs because the model keeps learning more dominant than deterioration by forming small batches to avoid increasing processing time after the $x_{\min}$-th part as indicated in Figure 1. Thus, as seen in all batch operations at each machine, the processing times decrease from the first part to the last part, e.g. 3.00 to 2.55 in batch 19 at machine 1. Second, the batch sizes do not follow shortest processing time (SPT) nor longest processing time (LPT) rules, but increase from batches 19 to 14, and then decrease until batch 1. This is because SPT and LPT usually appear in single-machine systems, not in flow shops. Third, deterioration and learning coincide from the first to the last part of each batch at each machine, e.g. from 3.00 to 2.55 for batch 19 at machine 1. The operator then experiences forgetting during interruption to the next batch, increasing the processing time from 2.55 to 2.93 at the beginning of batch 18, followed by deterioration and learning again, and so on. Partial forgetting during batch-to-batch interruption indicates the model works to reduce interruption intervals $I_{m,[i]}$ as minimum as possible. Meanwhile, parts continue to deteriorate during an interruption to the next machine, e.g., causing the processing time starts at 5.05 (instead of $t_2 = 5$) and at 4.66 (instead of $t_3 = 4$) at machine 2 and 3 respectively.

Table 3. The detail of optimal solution for Data Set 12

Batch i			19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
$Q_{[i]}$			2.89	3.04	3.22	3.43	3.63	3.60	3.58	3.55	3.51	3.47	3.43	3.37	3.29	3.20	3.09	2.93	2.70	2.36	1.71
$m = 1$	$B_{m,[i]}$#		8316	8377	8446	8522	8604	8689	8778	8869	8961	9055	9151	9247	9345	9443	9542	9645	9742	9835	9922
	$t_{[x]}$	First	3.00	2.93	2.87	2.85	2.84	2.84	2.83	2.83	2.83	2.83	2.82	2.82	2.82	2.82	2.82	2.82	2.82	2.83	2.84
		Last	2.55	2.05	1.97	1.94	1.92	1.90	1.89	1.88	1.87	1.87	1.86	1.86	1.85	1.85	1.85	1.87	1.90	1.92	1.94
$m = 2$	$B_{m,[i]}$#		8322	8390	8463	8542	8625	8712	8802	8894	8987	9082	9178	9275	9373	9472	9571	9671	9766	9855	9941
	$t_{[x]}$	First	5.05	4.96	4.86	4.84	4.83	4.82	4.82	4.82	4.81	4.81	4.81	4.81	4.81	4.80	4.80	4.80	4.80	4.81	4.82
		Last	4.35	3.58	3.47	3.42	3.38	3.36	3.34	3.33	3.32	3.31	3.30	3.29	3.29	3.28	3.28	3.31	3.35	3.39	3.42
$m = 3$	$B_{m,[i]}$#		8397	8468	8542	8622	8705	8792	8880	8970	9061	9154	9246	9340	9434	9527	9622	9716	9805	9891	9974
	$t_{[x]}$	First	4.66	4.59	4.47	4.43	4.40	4.37	4.34	4.31	4.28	4.25	4.21	4.17	4.13	4.08	4.03	4.01	4.00	4.00	4.00
		Last	3.89	3.10	2.98	2.91	2.86	2.82	2.79	2.76	2.73	2.70	2.67	2.64	2.61	2.58	2.55	2.57	2.60	2.63	2.65

Rounded to the nearest integer

Figure 4 shows the resulting Gantt-chart for Data Set 12. Small batches result short processes in the chart so that most setups are longer than the processes. However, if we reduce setup times by shortening setups, the number of batches will be higher, and the Gantt-chart will be more difficult to visualize. So the Gantt-chart is the best schedule that can be made by the model for the given data set.

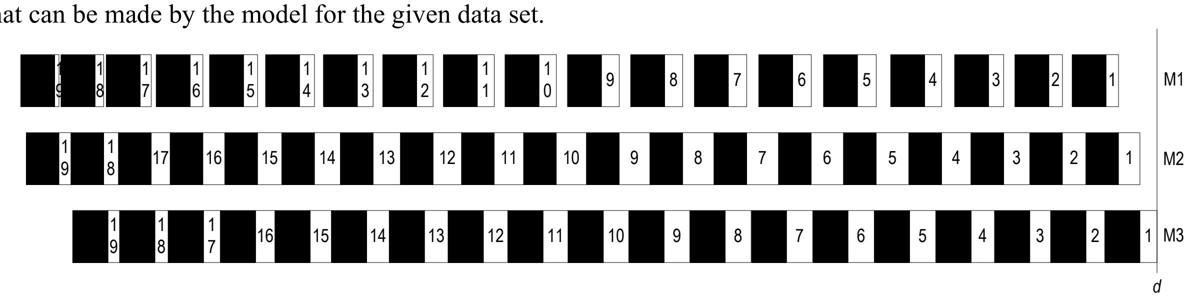

Figure 4. Gantt-chart for Data Set 12

6. Concluding Remarks

This research develops a flow shop batch scheduling model with learning-forgetting and part deterioration to minimize total actual flow time. An algorithm is developed to solve the model, and the algorithm works by trying different numbers of batches, starting from one, then increasing the number of batches one-by-one until the objective function value stops improving. We find that the faster operators learn, the lower the optimal number of batches and the lower the optimal total actual flow time. Meanwhile, the faster parts deteriorate, the higher the optimal number of batches and the higher the optimal total actual flow time. Future studies need to consider the requirement of pre-processing and its relationship with part deterioration.

Acknowledgements

This research was funded by Research, Community Services and Innovation (PPMI) of Bandung Institute of Technology (BIT) for fiscal year 2021.

References

Arizono, I., Yamamoto, A. & Ohta, H. (1992). Scheduling for minimizing total actual flow time by neural networks. *The International Journal of Production Research,* 30**,** 503-511.

Arroyo, J. E. C. & Leung, J. Y.-T. (2017). Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times. *Computers & Operations Research,* 78**,** 117-128.

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.

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.

Cheng, T., Kang, L. & Ng, C. T. (2004). Due-date assignment and single machine scheduling with deteriorating jobs. *Journal of the Operational Research Society,* 55**,** 198-203.

Fogarty, D. W., Blackstone, J. H. & Hoffmann, T. R. (1991). *Production & inventory management*, South-Western Publishing Company.

Goodarzi, E., Ziaei, M. & Hosseinipour, E. Z. (2014). Optimization analysis using LINGO and MATLAB. *Introduction to Optimization Analysis in Hydrosystem Engineering.* Springer.

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.

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.

Hidayat, N. P., Cakravastia, A., Samadhi, T. A. & Halim, A. H. (2016). A batch scheduling model for m heterogeneous batch processor. *International Journal of Production Research,* 54**,** 1170-1185.

Jaber, M. Y. & Bonney, M. (1996). Production breaks and the learning curve: the forgetting phenomenon. *Applied Mathematical Modelling,* 2**,** 162-169.

Jeng, A. A.-K. & Lin, B. M. (2004). Makespan minimization in single-machine scheduling with step-deterioration of processing times. *Journal of the Operational Research Society,* 55**,** 247-256.

Ji, M., Yang, Q., Yao, D. & Cheng, T. (2015). Single-machine batch scheduling of linear deteriorating jobs. *Theoretical Computer Science,* 580**,** 36-49.

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.

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**,** 546-564.

Li, S., Cheng, T. E., Ng, C. & Yuan, J. (2015). Single-machine batch scheduling with job processing time compatibility. *Theoretical Computer Science,* 583**,** 57-66.

Liang, Y. & Hui, C. W. (2016). Simultaneous subtour elimination model for single-stage multiproduct parallel batch scheduling with sequence dependent changeovers. *Computers & Chemical Engineering,* 87**,** 63-67.

Maulidya, R., Wangsaputra, R. & Halim, A. (2018). Batch Scheduling for Hybrid Assembly Differentiation Flow Shop to Minimize Total Actual Flow Time. *IOP conference series: materials science and engineering.*

Morton, T. & Pentico, D. W. (1993). *Heuristic scheduling systems: with applications to production systems and project management*, John Wiley & Sons.

Mosheiov, G. & Oron, D. (2008). Open-shop batch scheduling with identical jobs. *European Journal of Operational Research,* 187**,** 1282-1292.

Nembhard, D. A. & Uzumeri, M. V. (2000). Experiential learning and forgetting for manual and cognitive tasks. *International journal of industrial ergonomics,* 25**,** 315-326.

Parsa, N. R., Karimi, B. & Husseini, S. M. (2017). Exact and heuristic algorithms for the just-in-time scheduling problem in a batch processing system. *Computers & Operations Research,* 80**,** 173-183.

Sebrina, Diawati, L. & 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.

Sukoyo, S., Samadhi, T. A., Iskandar, B. P. & Halim, A. H. (2010). Model penjadwalan batch multi item dengan dependent processing time. *Jurnal Teknik Industri,* 12**,** pp. 69-80.

Wang, J., Qiao, F., Zhao, F. & Sutherland, J. W. (2016a). Batch scheduling for minimal energy consumption and tardiness under uncertainties: A heat treatment application. *CIRP Annals-Manufacturing Technology*.

Wang, J. & Xia, Z. (2006). Flow shop scheduling problems with deteriorating jobs under dominating machines. *Journal of the Operational Research Society,* 57**,** 220-226.

Wang, S., Liu, M., Chu, F. & Chu, C. (2016b). Bi-objective optimization of a single machine batch scheduling problem with energy cost consideration. *Journal of Cleaner Production,* 137**,** 1205-1215.

Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of the aeronautical sciences,* 3**,** 122-128.

Yin, Y., Ye, D. & Zhang, G. (2014). Single machine batch scheduling to minimize the sum of total flow time and batch delivery cost with an unavailability interval. *Information Sciences,* 274**,** 310-322.

Yusriski, R., Astuti, B. & Ilham, M. (2019). Integrated Batch Production and Multiple Preventive Maintenance Scheduling on A Single Machine to Minimize Total Actual Flow Time. *IOP Conference Series: Materials Science and Engineering.* IOP Publishing.

Yusriski, R., Sukoyo, S., Samadhi, T. & Halim, A. (2015a). Integer batch scheduling problems for a single-machine with simultaneous effect of learning and forgetting to minimize total actual flow time. *International Journal of Industrial Engineering Computations,* 6**,** 365-378.

Yusriski, R., Sukoyo, S., Samadhi, T. A. & Halim, A. H. (2015b). 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.

Biography

Dr. Dwi Kurniawan is a Senior Lecturer at Industrial Engineering Department, Institut Teknologi Nasional, Bandung, Indonesia. He received his BS, MSc and Doctor from Industrial Engineering Department, Bandung Institute of Technology (BIT), Indonesia. He teaches and conducts research in production planning and controlling, operations research and research methodology. He currently published his research in JIEM and in Asia Pacific Industrial Engineering and Management Society (APIEMS) conference proceedings.

Dr. Rinto Yusriski is a lecturer at Industrial Engineering Department, Universitas Jenderal Achmad Yani, Bandung, Indonesia. He received his BS from the university and his Master and Doctoral degree from the Graduate School of Industrial Engineering and Management, Bandung Institute of Technology (BIT), Indonesia. His research interests are in production scheduling, inventory control, JIT systems, maintenance and production systems.

Dr. Mohammad Mi'radj Isnaini is an Assistant Professor at the Industrial Engineering Department, Bandung Institute of Technology (BIT), Indonesia. He received his doctor degree from Kobe University Toyohashi, Japan. His research interests include production automation and manufacturing systems design.

Dr. Anas Ma'ruf is an Associate Professor at the Industrial Engineering Department, Bandung Institute of Technology (BIT), Indonesia. He received his doctorate from Toyohashi University of Technology, Japan. His research interests include production automation and manufacturing systems design. Dr. Anas is a board member of the Asia Pacific Industrial Engineering and Management Society.

Dr. Abdul Hakim Halim is a Professor at the Industrial Engineering Department, Bandung Institute of Technology (BIT), Indonesia. He received his degrees in Industrial Engineering: Doctor from University of Osaka Prefecture, Japan, and BS/MSc from BIT. His research interests include scheduling, inventory, and manufacturing systems design. He published his research outcomes in the scheduling/inventory areas in international journals such as IJPR, EJOR, PPC, IJPE, IJTech, and international conference proceedings. He also served as one of the Guest-Editors for CAIE. Dr. Halim is immediate past President of Asia Pacific Industrial Engineering and Management Society (APIEMS) and fellows of APIEMS and IEOM.