An integer batch scheduling model considering learning, forgetting, and deterioration effects for a single machine to minimize total inventory holding cost

R Yusriski¹, Sukoyo², T M A A Samadhi², A H Halim²
¹Department of Industrial Engineering, Universitas Jenderal Achmad Yani (UNJANI), PO. BOX 807 Bandung Indonesia
²Department of Industrial Engineering and Management, Institut Teknologi Bandung (ITB), Bandung 40132, Indonesia

E-mail: yusarisaki@yahoo.co.id

Abstract. This research deals with a single machine batch scheduling model considering the influenced of learning, forgetting, and machine deterioration effects. The objective of the model is to minimize total inventory holding cost, and the decision variables are the number of batches (N), batch sizes (Qi, i = 1, 2, ..., N) and the sequence of processing the resulting batches. The parts to be processed are received at the right time and the right quantities, and all completed parts must be delivered at a common due date. We propose a heuristic procedure based on the Lagrange method to solve the problem. The effectiveness of the procedure is evaluated by comparing the resulting solution to the optimal solution obtained from the enumeration procedure using the integer composition technique and shows that the average effectiveness is 94%.

1. Introduction
In the manufacturing industry, there are many cases where the processing time of parts could change due to operator learning-forgetting effects, and machine deterioration. The learning effect occurs when an operator work more efficient as the operator has produced identical parts repeatedly (Wright, [1]). On the other hand, if there is an interruption of two repetitive operations, the operator must re-learn the same operation on the next process after an interruption due to forgetting effect (Jaber and Bonney, [2]). Also, it can be observed that the capability of the machine will deteriorate after producing some parts (Kaminskiy and Krivtsov [3]).

Research in Mor and Mosheiov [4], Teyarachakul, et al. [5], Yang [6] shows that on batch scheduling problems the learning, forgetting, and deterioration effects also occur. All the research assume that the processing time of parts in a batch is constant and could change when the parts are scheduled in the next batches. Additionally, the research also assumes that all raw materials have been available since predetermined arrival time (say time zero) and the completed parts will exactly be delivered at the completion time of the batches. Therefore, the planner should consider the learning, forgetting, and deterioration effects when determining the number of batch and batch size to minimize the objective (among other total flow time, total holding cost, and total completion time).

In just-in-time (JIT) production systems, there are the systems (Halim et al. [7]; Halim and Ohta, [8]; Yusriski et al.[9]) where the company is able of managing the parts to be received at the shop at the right time and the right quantities, and the parts must be delivered at a common due date.
For this situation, if the planner does not consider the learning effect, the completion time of the parts could be earlier than the expected time. It leads to a condition where the total inventory holding cost increase as all completed parts must be delivered at a common due date. On the contrary, if the forgetting or deterioration effects are not considered, the completion time of parts will be later than the expected time so that some parts could be finished after the due date, and this cause the company should bear the penalty cost.

2. Problem Formulation
This research discusses an integer batch scheduling model considering learning, forgetting, and deterioration effects in JIT production system to minimize the total inventory holding cost. Suppose that there are \( n \) integer number parts of a single item to be processed on a single machine and to be divided into \( N \) batches with batch sizes, \( Q_i \) \((i = 1, 2, ..., N)\), the completed parts must be delivered at the common due date \((d)\) simultaneously. The batches are scheduled backwardly where the processing time of batch \((T_i)\) is influenced by operator learning-forgetting and machine deterioration effects. A setup time \((s)\) is needed before starting to process a new batch. However, any input material is not needed to accomplish the setup activity so that the parts to be processed can arrive at the shop floor at the same time when the machine is ready to begin processing their parts in a batch \((B_i)\). The objective is to minimize total inventory holding cost \((TFC)\). Two input costs associated with the objective, i.e., the inventory holding cost per unit time for a part in the completed batches \((c_i)\) and in-process batches \((c_s)\). The solve this problem, there are three decisions, i.e., to determine the number of batches, batch sizes, and the sequence of the resulting batches.

3. Inventory Holding Cost
According to Halim and Ohta [8], in batch scheduling problem, the inventory holding cost \((FC)\) of the batch \((L_i)\) can be evaluated during the parts in a batch flows in the shop, from its arrival to the common due date. It could be defined as the accumulation of the inventory holding cost for completed parts in batch and the inventory holding cost for in-process parts in batch. This research develops the formula for calculating the total inventory holding cost \((TFC)\) in Halim and Ohta [8] by replacing the static batch processing time, \((i)\) by the dynamic processing time of batch \((T_i)\) which is that processing time influenced by learning, forgetting, and deteriorating. The inventory holding cost function as shown as follows.

\[
TFC = \sum_{i=1}^{N} (F_i + L_i) 
\]

where \[
F_i = \left( \sum_{j=1}^{i} (s + T_j Q_i) - s - T_i Q_i \right) c_i Q_i; \\
L_i = c_s T_i Q_i^2
\]

Eq (1) shows that the inventory holding cost in a batch could be calculated by adding the inventory holding cost of completed parts in a batch \((F_i)\) with the inventory holding cost of in-process parts in a batch \((L_i)\). Therefore, the total inventory holding cost \((TFC)\) can be calculated by summing the inventory holding cost in all batches.

4. Learning-Forgetting and Deterioration Functions
There are some functions of learning, forgetting, and deterioration. The reader can find out the functions in (Wright [1]; Jaber and Bonney [2]; Teyarachakul et al. [5]; Janiak et al. [10]; Yusriski et al. [11; Yusriski et al. [12]). We develop the learning function in Wright [1], the forgetting and the learning-forgetting functions in Jaber and Bonney [2], and the deterioration function by RCOF function in Kaminskiy & Krivtsov [3] and AFT function in Iskandar and Jack [13]. This research proposes the learning, forgetting, and deterioration functions as shown as follows.

\[
IOP Conf. Series: Materials Science and Engineering 319 (2018) 012038 doi:10.1088/1757-899X/319/1/012038
\]
\[ T_i = \max \{ p(E_i)^{-m} \}, \delta = 1 \] (2)

where

\[ E_i = 1 + \sum_{i=1}^{N} Y_i; \quad Y_i = \left[ \left( Q_{i+1} - Q_{i} \right)^{(m)} f + \left( X_i \right)^{(m)} f \right], \quad 0 < \delta < 1; \]

\[ X_i = \left[ Q_{i+1} \left( 1 - \Lambda(B_i) \right) + \left( s(1-m) / T_{i+1} \right)^{(1-m)} \right]; \quad m = -\log \delta / \log 2; \]

\[ f_i = m(1-m) \log (Q_{i+1} + 1) / \log (1 + C_i); \quad C_i = t_b / H_i; \]

\[ H_i = \left( T_{i+1} / (1-m) \right)^{(1-m)}; \quad \Lambda(B_i) = \left( U_i / \alpha \right) ^{\beta}; \]

\[ U_i = \sum_{i=1}^{N} T_{i+1} Q_{i+1}; \quad \alpha_i = \alpha_1 \left[ d / \left( d + U_i \right) \right]; \quad \gamma = p(q)^{-m}. \]

This research adopts the learn-forget curve model (LFCM) in Jaber dan Boney [2] to develop the learning-forgetting and machine deterioration function. The LFCM function assumes that the processing time of parts influenced by learning and forgetting effect. We assume there is an initial processing time \((p)\) to process parts in batch \(N\)-th position (scheduled backwardly). The processing time of parts starting from batch on a position \((N-1)\) to the first position could decreases to be in accordance with the experience of operator \((E_i)\) to process the number of parts (as the consequences of the learning effect), and then constant when that achieves a learning threshold \((v)\). The effect of learning could depend on the value of learning rate \((\delta)\) where it is between 50% and 100% where \(\delta = 100\%\) implies no learning, and \(\delta = 50\%\) implies maximum learning. It can also be observed that decreasing the value of \(\delta\), increases the learning index \((m)\).

The forgetting effect will occur during a setup time between two consecutive batches so that the processing time of parts from batch position \((N-1)\) to the first position could increase in line with the number of batch setup, and that could retard the effect of learning. The effect of forgetting depends on the value of forgetting index \((f)\), which associated with the learning index, and the interval time of break for an operator that will cause the operator full forget the operation \((t_b)\). The processing of batch also affected by the machine deterioration \((\Lambda(B_i))\), calculated by ROCOF function. The two parameters, adopted by Weibull distribution, are proposed, that is the scale parameter \((\alpha)\) and the shape parameter \((\beta)\). The value of \(\beta\) should be more than one which to indicate that failure rate increases with time. We assume that the deterioration could occur during an interval \([0, \alpha]\) with a due date \((d)\) is earlier than \(\alpha\). Therefore, the probability of failures during the scheduling period would be less than one \((\Lambda(d) \leq 1)\). This indicates that the machine lifetime will end beyond the due date. We also adopt the AFT model (the accelerated failure time) to consider the value of \(\alpha\) could decrease as the tool usage.

5. Mathematical Model
This research develops the model in Halim and Ohta [8] by including an additional assumption that the batch processing time is influenced by learning, forgetting, and deterioration effects simultaneously. The function of the batch processing time \((T_{ij})\) is formulated in Eq. (2). A mathematical model of the problem can be written as follows:

\[ \text{Minimize } TFC_{i} = \sum_{j=1}^{N} \left[ \left( \sum_{i=1}^{j} \left( s + T_{ij} Q_{ij} \right) - s - T_{ij} Q_{ij} \right) \right] c_1 Q_{ij} + c_2 T_{ij} Q_{ij}^2 \] (3)
Subject to constraints

\[ \sum_{i=1}^{N} Q_{[i]} = n \]  
\[ \sum_{i=1}^{N} T_{[i]}Q_{[i]} + (N-1)s \leq d \]  
\[ B_{[i]} + T_{[i]}Q_{[i]} = d \]  
\[ Q_{[i]} \geq 1 \text{ and integer}, \]  
\[ 1 \leq N \leq n \text{ and integer}, \]

Eq. (3) expresses the objective of minimizing the total inventory holding cost where the batch processing time is influenced by learning, forgetting, and deterioration effects simultaneously. Constraint (4) is a material balancing that expresses the number of parts in all batches produced equal to demand. Constraint (5) expresses that all batches should be scheduled at the available time, that is, the time interval from the start of scheduling period (time zero) to their common due date. Constraint (6) expresses that the batch scheduled in the first position backwardly should be completed exactly at the common due date. Constraint (7) expresses that batch sizes should be one or more and integer. Constraint (8) expresses that the number of batches should be a positive integer between 1 and the number of demands.

6. Problem Solution

We propose the heuristic solution based on Lagrange method. The step for searching the solution can be found in Halim et al. [7] and yusriski et al. [11]. The formula of the solution can be shown as follows:

\[ Q_{[i]} = \max \left\{ \left( n - \sum_{k=i+1}^{N} Q_k \right) / i + \left\{ c_i \left( 2c_2 - c_1 \right) / \left( 1/2(i+1) \right) \left( s / T_{[i]} \right) \right\} - \left\{ c_i / \left( 2c_2 - c_1 \right) \left( s / T_{[i]} \right) i \right\}, \text{ where } i = N, \ldots, 1 \right\} \]

\[ N_{\max} = \min \left\{ 1 + \left( d - T_{\min}n \right) / s , n \right\} \]

**Proposition 1.** Suppose that there are N batches with batch size, \( Q_{[i]} \), \( i = 1, 2, \ldots, N \), respectively, and the processing time of a batch is influenced by the simultaneous effects of learning, forgetting, and deterioration. The optimal backward sequence that minimizes the total inventory holding cost is obtained by sequencing the resulting batches in non-increasing order of batch sizes. \( Q_{[1]} \geq Q_{[2]} \geq \ldots \geq Q_{[N-1]} \geq Q_{[N]} \)

Proof. Looking at Eq. (3). The formula can be written as follows:

\[ \text{Minimize } TFC = c_1 s \sum_{i=1}^{N} i Q_{[i]} - \left( 1/2 \right) c_1 T \sum_{i=1}^{N} Q_{[i]}^2 + \left( 1/2 \right) c_1 T \left( \sum_{i=1}^{N} Q_{[i]} \right)^2 - s \sum_{i=1}^{N} Q_{[i]}^2 + c_2 T \sum_{i=1}^{N} Q_{[i]}^2 \]

The order of batches in a sequence just influenced by the value of the first term on the right-hand side. Relaxing the constant \( c_1 \) and \( s \), we can see that since the value of index \( i \) (\( i=1, 2, \ldots, N \)) are already in increasing order. Therefore, minimizing the TFC can be obtained by sequencing the batch size, \( Q_{[i]} \), in a non-increasing order using the backward scheduling approach (\( Q_{[1]} \geq Q_{[2]} \geq \ldots \geq Q_{[N-1]} \geq Q_{[N]} \)). □
7. The Proposed Procedure

Based on Lagrange Relaxation method, we propose the heuristic procedure as shown below:

Step 1: Set input parameters of \( n, d, p, s, \delta, v, w, t_b, \mu, \alpha, \beta, c_1, c_2 \). Continue to Step 2.

Step 2: Compute \( N_{\text{max}} \) by Eq. (10). Set \( j \) as the index of the number of batches \(( j=1,2,..., N_{\text{max}} )\). Continue to Step 3.

Step 3: Begin with \( j=1 \). Set \( Q_{[1]} = n, T_{[1]} = p \). Continue to Step 4.

Step 4: Use Constraint (5) to evaluate the makespan of the job, if it does not exceed the available time, continue to Step 5; otherwise, there is no solution because the demand cannot be scheduled; then STOP.

Step 5: Compute a total inventory holding cost \( (TFC_{[1]}) \) by Eq. (3) and set the value of \( TFC_{[1]} \) as a temporary optimal solution \( (TFC^*) \). Continue to Step 6.

Step 6: Set an index of the number of batches \( j = j+1 \); if \( 1 < j \leq N_{\text{max}} \) then continue to Step 7 otherwise the \( TFC^* \) is an optimal solution and the STOP.

Step 7: Set \( N = j \). Use Eq. (9) to compute batch size, \( Q_{[i]} \), and Eq. (2) to compute batch processing time \( (T_{[j]} ) \) using backward index computation \( (i=N, ..., 1) \). Continue to Step 8.

Step 8: Use Constraint (4) to evaluate the sum of batch sizes, if that is equal to than the demand, continue to Step 9, otherwise set \( TFC^* \) as the optimal solution and then STOP.

Step 9: Use Constraint (5) to evaluate the makespan of the job, if it does not exceed the available time then continue to Step 10; otherwise, the \( TFC^* \) is the optimal solution and then STOP.

Step 10: According to Proposition 1, sequence the batches in non-increasing \( Q_{[i]} \). The largest \( Q_{[i]} \) is closest to the due date. Compute the total inventory holding cost of the batch \( (TFC_{[i]}) \) by Eq. (3). Continue to Step 11.

Step 11: Compare \( TFC_{[i]} \) and \( TFC^* \). If \( TFC_{[i]} < TFC^* \) then \( TFC_{[i]} \) is seted as a \( TFC^* \) otherwise \( TFC^* \) is still as a \( TFC^* \). Return to Step 6.

8. Numerical Experience

We demonstrate the ability of the heuristic procedure to solve the following numerical examples.

Suppose input parameters as follows: \( p=0.5, n=5, d=10, s=1, v=0.25, \delta=90\%, t_b=20, w=1, \mu=1, a=20, \beta=2, c_1=$3 and \( c_2=$2.5. The result of processing by the heuristic algorithm is presented in Table 1.

**Table 1. The heuristic solution of example 1**

| \( N_{\text{max}} \) | \( N \) | \( Q_{[i]} = 1, 2, 3, 5 \) | \( T_{[j]} , i = 1, 2, 3, 5 \) | \( B_{[i]} , i = 1, 2, 3, 5 \) | \( TFC \) ($) |
|----------------------|--------|--------------------------|--------------------------|--------------------------|----------|
| 5                    | 1      | [5]                      | [0.5]                    | [7.5]                    | 31.25    |
|                      | 2      | [4, 1]                   | [0.5, 0.5]               | [8, 6.5]                 | 24.25    |
|                      | 3      | [3, 1, 1]                | [0.42, 0.5, 0.5]         | [8.7, 7.2, 5.7]          | 21.63* optimal |
|                      | 4      | [2, 1, 1, 1]             | [0.42, 0.45, 0.5, 0.5]   | [9.2, 7.7, 6.2, 4.7]     | 25.86    |

Table 1 presents the solution of the procedure. The result of Eq. (10) yields \( N_{\text{max}} = 5 \). The search of algorithm solution begins with the number of batches, \( N \), equals 1. The result of \( TFC \) is 31.25. The search of the solution is continued for \( N = 2 \). The result of \( TFC \) is 24.25. The value of the \( TFC \) for \( N = 2 \) is compared with \( N = 1 \), and it is found that the value of \( TFC \) for \( N = 2 \) is cheaper than \( N = 1 \). Thus, the value of the total inventory holding cost for \( N = 2 \) is a temporarily optimal solution, \( TFC^* \). The search continues for \( N = 3 \). It is found that the value of \( TFC \) is 21.63, and it is found that the value of \( TFC \) for \( N = 3 \) is cheaper than \( N = 2 \). Thus, the value of the total inventory holding cost for \( N = 3 \) is a temporarily optimal solution, \( TFC^* \). The solution for \( N = 4 \) found that the value of \( TFC \) is 25.86, and it is found that the value of \( TFC \) for \( N = 3 \) is cheaper than \( N = 4 \). Thus, the value of the total inventory holding cost for \( N = 3 \) is an optimal solution, obtained by \( Q_{[1]} = 3, Q_{[2]} = 1 \) and \( Q_{[3]} = 1 \).
The comparison test is determined by comparing the solutions of the heuristic procedure to the optimal solution produced by the enumeration procedure using the integer composition technique (see Yusriski et al. [11]). We show the result of the tests for the number of demand between 1 unit and 20 units since the enumeration procedure cannot solve the number of demands more than 22 units due to out of memory.

Table 2. The effectiveness results

| n  | Proposed Procedure | Enumeration Procedure | Comparison Test Results |
|----|--------------------|-----------------------|-------------------------|
|    | TFC ($) | Computational Time (second) | TFC ($) | Computational Time (second) | Solution Effectiveness (%) | Time-Saving (second) |
|----|---------|--------------------------|---------|--------------------------|---------------------------|-------------------|
| 5  | 21.63   | 0.00183                  | 21.63   | 0.01498                  | 100                       | 0.01315           |
| 7  | 52.58   | 0.00275                  | 48.40   | 0.01145                  | 91                        | 0.0087            |
| 9  | 43.82   | 0.00296                  | 42.31   | 0.06437                  | 96                        | 0.06141           |
| 10 | 42.12   | 0.00363                  | 40.43   | 0.11895                  | 96                        | 0.11532           |
| 11 | 137.62  | 0.00288                  | 128.60  | 0.22987                  | 93                        | 0.22699           |
| 12 | 96.90   | 0.00163                  | 87.95   | 0.44892                  | 90                        | 0.44729           |
| 14 | 148.28  | 0.00139                  | 138.58  | 2.02380                  | 93                        | 202379.9986       |
| 16 | 98.11   | 0.00302                  | 90.19   | 10.70594                 | 91                        | 1070593.997       |
| 18 | 113.84  | 0.00645                  | 107.86  | 78.24584                 | 94                        | 7824583.994       |
| 20 | 137.85  | 0.00256                  | 133.49  | 941.85271                | 97                        | 94185271          |

Table 2 shows that although the heuristic procedure does not guarantee an optimal solution. It may produce an outstanding solution with the average effectiveness is 94%. Additionally, the average of time-saving shows that the heuristic procedure is more efficient than the enumeration procedure.

Concluding Remarks

The current research deals with single machine batch scheduling problems to minimize inventory holding cost in a just-in-time production system considering that the batch processing time is influenced by learning, forgetting, and deterioration effects simultaneously. We propose a heuristic procedure based on Lagrange relaxation method to solve the problem. The effectiveness of the procedure is determined by comparing the resulting solution with the optimal procedure obtained from the enumeration procedure. The comparison test result shows that the average effectiveness is 94%. The comparison test also shows that the computational time of the heuristic procedure is more efficient than the enumeration procedure. In this paper, we have considered problems with only a single item. Hence, further research may focus on the multi-items problem.

Acknowledgments

The authors wish to thank the editors and the referees for their constructive comments and suggestions. This research was supported by Institute Teknologi Bandung (ITB) and Universitas Jenderal Achmad Yani (UNJANI) Bandung.

References

[1] Wright T P 1936 Factors affecting the cost of airplanes *Journal of Aeronautical Sciences* 3 122–128.

[2] Jaber M Y and Bonney M 1996 Production break and the learning curve: The forgetting phenomenon *Applied Mathematical Modelling* 20 162–169.
[3] Kaminskiy M P and Krivtsov V V 2008 An integral measure of aging/rejuvenation for repairable and non-repairable systems Reliability and Risk Analysis: Theory & Applications 1 69–76.

[4] Mor B and Mosheiov G. 2012 Batch scheduling with step-deteriorating processing times to minimize flow time Naval Research Logistics 59 587–600.

[5] Teyarachakul, Chand S and Ward J 2011 Effect of learning and forgetting on batch sizes. Production and Operations Management 20 116–128.

[6] Yang W-H 2009 A batching problem with learning effect considerations Asia-Pacific Journal Of Operational Research 26 307–317.

[7] Halim A H, Miyazaki S and Ohta H 1994 Batch scheduling problem to minimize actual flow times of component through the shop under JIT environment European Journal of Operational Research 72 529–544.

[8] Halim A H and Ohta H 1994 Batch Scheduling Problem to minimize inventory cost in the shop with both receiving Batch Scheduling Problem to minimize inventory cost in the shop with both receiving and delivery Just-In-Time International Journal of Production Economics 33 185–194.

[9] Yusriski R, Astuti B, Sukoyo, Samadhi T M A A and Halim A H 2015 Integer batch scheduling problems for a single-machine to minimize total actual flow time Procedia Manufacturing 2 118–123.

[10] Janiak, Lichtensteind M and Ruson A 2011 Scheduling jobs with a linear model of simultaneous ageing and learning effects Decision Making in Manufacturing and Services 5 37–48.

[11] Yusriski R, Sukoyo S, Samadhi T M A A and Halim A H 2015 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.

[12] Yusriski R, Sukoyo, Samadhi T M A. A and Halim A H 2016 An Integer Batch Scheduling Model for a Single Machine with Simultaneous Learning and Deterioration Effects to Minimize Total Actual Flow Time IOP Conference Series: Materials Science and Engineering 114 1-10.

[13] Iskandar B P and Jack N 2011 Warranty Servicing with Imperfect Repair for Products Sold with a Two-Dimensional Warranty Chapter Replacement Models with Minimal Repair, Part of the series Springer Series in Reliability Engineering 163-175.