

An Integer Batch Scheduling Model for a Single Machine with Simultaneous Learning and Deterioration Effects to Minimize Total Actual Flow Time

R Yusriski, Sukoyo, T M A A Samadhi, A H Halim

Department of Industrial Engineering and Management,
Institut Teknologi Bandung, 40132, Indonesia

E-mail: yusarisaki@yahoo.co.id

Abstract. In the manufacturing industry, several identical parts can be processed in batches, and setup time is needed between two consecutive batches. Since the processing times of batches are not always fixed during a scheduling period due to learning and deterioration effects, this research deals with batch scheduling problems with simultaneous learning and deterioration effects. The objective is to minimize total actual flow time, defined as a time interval between the arrival of all parts at the shop and their common due date. The decision variables are the number of batches, integer batch sizes, and the sequence of the resulting batches. This research proposes a heuristic algorithm based on the Lagrange Relaxation. The effectiveness of the proposed algorithm is determined by comparing the resulting solutions of the algorithm to the respective optimal solution obtained from the enumeration method. Numerical experience results show that the average of difference among the solutions is 0.05%.

1. Introduction

Research on batch scheduling, for example Santos and Magazine [1], Dobson et al. [2], Halim et al. [3], Halim and Ohta [4], Potts and Kovalyov [5], Mosheiov et al. [6], Mosheiov and Oron [7], assumes that processing times of respective jobs are always the same. In reality, there are conditions where the processing time of jobs in a batch will depend on its position in the schedule due to the effect of machine deterioration. It is due to the capability of a machine will decline with the increase of the number of jobs that have already been produced. This situation can be observed when the sharpness of a lathe chisel is getting blunt after producing a number of jobs. Since the blunt chisel requires more power to cut, the operator should re-setup the machine periodically by slowing down the speed of production to keep the quality of product high. It leads a condition that the processing time of jobs in the batches produced earlier would be shorter than those in the batches produced later.

Currently, the research on batch scheduling problems with machine deterioration effect has been widely studied, such as Barketau et al. [8], Ji and Cheng [9], and Mor and Mosheiov [10]. Barketau et al. [8] study the problem to minimize makespan. The researchers show that the longer the waiting time of the jobs, the longer time is needed to produce them. Meanwhile, Ji and Cheng [9] deal with a similar problem by assuming that deterioration is a linear function of the length of waiting time of the jobs before being processed. Moreover, Mor and Mosheiov [10] also deal with a similar problem to



minimize flow time, and the researchers propose an algorithm that could be applied to any manufacturing systems with either both non-integer or integer batch sizes.

The discussions of batch scheduling with dependent processing times also deal with the cases of learning effect. According to Kuo and Yang [11], those operators producing the same jobs repeatedly will work more efficient as a learning effect. The researchers have discussed the effect on batch scheduling for two objectives, namely, to minimize makespan and to minimize total completion time. Meanwhile, Yang [12] shows that learning effect occurs not only in a job processing time but also in batch setup time, and proposes a model with learning effect in both processing and setup times simultaneously to minimize makespan. Furthermore, Yusriski et al. [13] study both learning and forgetting effects simultaneously. Learning effect is influenced by the number of jobs that had been produced earlier, forgetting effect occurs as the operator conducts a batch setup.

This research develops a single machine batch scheduling model under a Just-In-Time (JIT) production systems by assuming that processing times of jobs processed as batches are influenced by learning and deterioration effects simultaneously. The objective is to minimize total actual flow time, defined as a time interval between the arrival of all jobs and their common due date. The definition of actual flow time has been discussed in Halim et al. [3]. The decision variables of current research are the number of batches, integer batch sizes, and the sequence of the resulting batches.

The research discussion is organized as follows. Section two discusses batch processing time with learning, and deterioration effects. Section three discusses the problem formulation. Section four discusses a proposed heuristic algorithm based on the Lagrange Relaxation. Section five discusses some numerical experience. Finally, the last section is concluding remarks.

2. Batch Processing Time with Learning and Deterioration Effects

2.1. Batch Processing Time with Learning Effect

The effect of learning phenomenon is an operator would take a shorter jobs processing time in line with the increase of operator experience in processing the job. Different researchers have offered different function models to describe the learning effect (see Yelle [14], Teyarachakul et al. [15]). One of which is Wright's Cumulative Average Power (CAP). According to Wright [16], an operator's experience would increase in line with the number of jobs the operator has produced. The learning function of CAP model is expressed by the following Equation:

$$T_{[x]} = T_{[1]}x^{-m}, \text{ where } m = -\log(\delta) / \log(2) \quad (1)$$

$T_{[x]}$ is the processing time of producing x-units, $T_{[1]}$ is the initial processing time or a processing time of the job firstly processed, δ is learning rate, and m is learning slope. The value of δ is $0 < \delta < 1$. However, in a manufacturing system, the value of δ is usually between 0.7 and 0.9 (see Jaber and Bonney, [17]). The smaller the value of δ is, the greater the learning effect will be Yusriski et al. [13] suggested a batch processing time based on CAP model in a backward scheduling approach as follows:

$$T_{[i]} = \max \left\{ p \left(1 + \sum_{k=i}^N Q_{[k+1]} \right)^{-m}, v \right\}; \text{ where } m = -\log \delta / \log 2; i=1,2,\dots,N \quad (2)$$

The i^{th} batch processing time ($T_{[i]}$) would be on a basis of the maximum value of learning function and learning threshold (v). Though adopting a backward scheduling technique, the learning effect is steadily determined by the number of earlier produced jobs. The jobs in N^{th} batch would each

processed for a time period of p (initial processing time). Meanwhile, jobs processing time from $(N-1)^{\text{th}}$ batch position to first batch position would be shorter in line with the number of produced jobs. However, the job processing time would be constant when the processing time achieves a learning threshold.

An example of computation by using Eq. (2) is shown as follows. Suppose there is a demand of 70 units divided into five batches, each being containing $Q_{[1]} = 30$, $Q_{[2]} = 20$, $Q_{[3]} = 10$, $Q_{[4]} = 7$, and $Q_{[5]} = 3$. The batches are sequenced by using a backward approach. Assume $T_{[3]} = p = 1$ minutes, $\delta = 0.09$, or $m = 0.153$, and $v = 0.65$ minutes. The calculation results of batch processing times by Eq. (2) are as follows $T_{[5]} = 1$, $T_{[4]} = 0.81$, $T_{[3]} = 0.70$, $T_{[2]} = 0.65$, and $T_{[1]} = 0.65$.

2.2. Batch Processing Time with Deterioration Effects

A deterioration phenomenon occurs on any tool in line with a time or its use, even though it is created by the best design, production process, and construction (Jiang and Murthy, [18]). According to Kaminskiy and Krivtsov [19], a repairable deterioration on a tool is defined as an increasing rate of occurrence of failures (ROCOF). ROCOF is used to estimate the number of failures of a tool in a time interval of $[0, t]$. If the maintenance measure taken is a minimal repair with a two-parameter (α, β) of Weibull distribution, the expected number of failures in a time interval of $[0, t]$ may be as follow:

$$\Lambda(t) = \left(\frac{t}{\alpha} \right)^{\beta} \quad (3)$$

$\Lambda(t)$ is the expected number of failures during the closed time interval $[0, t]$, α is a scale parameter, and β is the shape parameter. As a note, deterioration usually occurs in wear-out phase by a value of $\beta > 1$.

The present research assumed that production scheduling would be accomplished in wear-out phase, and thus a deterioration effect would appear. We assume that the deterioration would occur during an interval between time zero to due date $[0, d]$. Thus, a probability for the number of failures during the scheduling period would be less than one ($\Lambda(d) < 1$). If the deterioration will be evaluated at the time when machine began to process a batch ($B_{[i]}$), the job processing time in the batch by using a backward approach is formulated as follows:

$$t_{[i]} = p + \mu \left\{ \Lambda(B_{[i]}) \right\} \quad (4)$$

p is initial processing time when machine began to process the last position of batch ($Q_{[N]}$), μ is time addition per one failure, and $\Lambda(B_{[i]})$ is estimated number of failures during time interval of $[B_{[N]}, B_{[i]}]$. Solving Eq. (3) and Eq. (4) simultaneously yields:

$$T_{[i]} = p + \mu \left(\sum_{k=i}^N T_{[k+1]} Q_{[k+1]} / \alpha \right)^{\beta}; \text{ where } \left(\sum_{k=1}^i T_{[k+1]} Q_{[k+1]} / \alpha \right)^{\beta} < 1, T_{[N+1]} = 0, T_{[N]} = p \text{ and } Q_{[N+1]} = 0 \quad (5)$$

An example of the computation of Eq. (5) followed. If there were three batches ($N=3$) with the batch sizes were $Q_{[3]}=1$, $Q_{[2]}=2$, $Q_{[1]}=3$, respectively, $p=2$ minute, $\mu=0.2$, $\alpha=4$, and $\beta=2$, then, by using Eq.(5), it is found that $T_{[3]}=0.5$, $T_{[2]}=0.50$, and $T_{[1]}=0.53$.

2.3. Processing time with both learning and deteriorating effects simultaneously

Based on Equations (2) and (5), a model of learning-deterioration function simultaneously on a batch scheduling problem is shown as follows:

$$T_{[i]} = \max \left\{ p \left(1 + \sum_{k=i}^N Q_{[k+1]} \right)^{-m}, v \right\} + \mu \left(\sum_{k=i}^N T_{[k+1]} Q_{[k+1]} / \alpha \right)^{\beta}; \text{ where } m = -\log \delta / \log 2 \quad (6)$$

A numeric example of Eq. (6) as shown as follows. Suppose there is a demand of 70 units divided into five batches, each being containing $Q_{[1]} = 30$, $Q_{[2]} = 20$, $Q_{[3]} = 10$, $Q_{[4]} = 7$, and $Q_{[5]} = 3$, are scheduled by using a backward approach. Assume $T_{[5]} = p = 1$ minutes, $\delta = 0.9$, or $m = 0.153$, and $v = 0.1$ minutes, $\alpha = 30$ and $\beta = 10$, and batch setup time (s) = 1 minute, then, according to Eq. (9), the batches processing times are as follows: $T_{[5]} = 1$, $T_{[4]} = 0.81$, $T_{[3]} = 0.7$, $T_{[2]} = 0.64$, and $T_{[1]} = 0.69$. The results of the computation showed that the batch processing times were decreasing start from $Q_{[5]}$ until, $Q_{[2]}$ and then increase in $Q_{[1]}$. The increasing or decreasing trends of a batch processing time is largely determined by either learning or deterioration effects.

3. Problem Formulation

There are n -jobs to be grouped into N integer batches where these batches are processed on a single machine scheduled by a backward scheduling approach. This research assumes that the arrival time of the jobs in a batch exactly coincide at the time when the machine start to process the batch, and all the jobs must be completed by the machine just or before the due date. The processing time of batch is different from that at another batch due to the simultaneous learning, and deterioration effects. The setup times of batch are identical and needed whenever starting a new batch. In order to minimize total actual flow time, there is a trade-off between keeping small the number of setups, by having large batches, and keeping small the time each job waits, by having small batches. This research want to determine the number of batches (N), batch sizes ($Q_{[i]}$ where $i=1,2,\dots,N$), and to schedule the resulting batches under the condition that the processing time of batch influenced by simultaneous learning, and deteriorating effects. Let us define the following constants and variables:

Parameters

d	: due date
n	: the number of demands
s	: batch setup time
p	: batch processing time at N^{th} position (initial processing time)
δ	: learning rate
m	: learning parameter or learning slope
v	: minimum processing time
α	: scale parameter
β	: shape parameter

Dependent variables

$T_{[i]}$: batch processing time at the i^{th} position, $i=1, 2, \dots, N$
$B_{[i]}$: time to start to process batch at the i^{th} position, $i=1, 2, \dots, N$

Objective:

F^a	: actual flow time
-------	--------------------

The mathematical model of the problem could be written as follows:

Objective:

$$\text{minimize } F^a = \sum_{i=1}^N \left(\sum_{j=1}^i (s + T_{[j]} Q_{[j]}) - s \right) Q_{[i]} \quad (7)$$

Subject to constraints

$$\sum_{i=1}^N Q_{[i]} = n \quad (8)$$

$$\sum_{i=1}^N T_{[i]} Q_{[i]} + (N-1)s \leq d \quad (9)$$

$$B_{[1]} + T_{[1]} Q_{[1]} = d \quad (10)$$

$$Q_{[i]} \geq 1 \text{ and integer,} \quad (11)$$

$$1 \leq N \leq n \text{ and integer,} \quad (12)$$

Eq. (7) expressed the minimized objective, that is, to minimize the total actual flow time with a processing time influenced by learning and deterioration effects simultaneously. Constraint (8) is a material balancing expressing the number of parts produced is equal to demands. Constraint (9) expressed that the total processing time needed to process all batches should not exceed the available time, that is, in a time interval from the start of scheduling to the due date. Constraint (10) expressed that the batch scheduled in the first schedule should be finished precisely on the due date. Constraint (11) expressed that batch size should be one or more and integer. Constraint (12) expressed that the number of batches should be a positive integer between 1 and the number of demands.

4. The Proposed Heuristic Solution

This research proposes the heuristic solution based on Lagrange Relaxation method. It begins with doing a relaxation on the dependent variable of batch processing time ($T_{[i]}$) as a constant value (T). Thus, Eq. (7) could be rewritten into:

$$\text{minimize } F^a = s \sum_{i=1}^N i Q_{[i]} + (1/2) T \sum_{i=1}^N Q_{[i]}^2 + (1/2) T \left(\sum_{i=1}^N Q_{[i]} \right)^2 - s \sum_{i=1}^N Q_{[i]} \quad (13)$$

Solving the problem by applying Karush-Kuhn-Tucker (KKT) yielded a condition necessary for a stationary point as follows:

$$si + T Q_{[i]} + T \left(\sum_{i=1}^N Q_{[i]} \right) - s - \lambda_1 - T \lambda_2 - T \lambda_3 = 0, \quad i = 1, 2, \dots, N \quad (14)$$

$$\sum_{i=1}^N Q_{[i]} - n = 0 \quad (15)$$

$$T \sum_{i=1}^N Q_{[i]} + Ns - s - d + \mathcal{G} = 0 \quad (16)$$

$$B_{[1]} + T_{[1]} Q_{[1]} - d = 0 \quad (17)$$

Based on Eq. (13), the problem discussed is to minimize a convex function (square in batch size) with linear boundaries (shown in Eqs. (14)-(17)).

4.1. Determining Integer Batch Size

Solving Eq. (14) and Eq. (15) simultaneously yields:

$$Q_{[i]} = (\lambda_1 / T) + \lambda_2 + \lambda_3 + (s / T) - n - (s / T) i \quad (18)$$

Substituting Eq. (18) to (15) yields:

$$\lambda_1 = (nT) - (T \lambda_2) - (T \lambda_3) + (s / 2)(N - 1) + (nT / N) \quad (19)$$

Substituting Eq. (19) to Eq. (18) yields :

$$Q_{[i]} = (n / T) + (1 / 2)(N + 1) - (s / T) i, \quad i = 1, 2, \dots, N \quad (20)$$

Substituting T in Eq. (20) for $T_{[i]}$ in Eq. (6) yields:

$$Q_{[i]} = \left(n / T_{[i]} \right) + (1/2)(N+1) - (s / T_{[i]})i, \quad i = 1, 2, \dots, N \quad (21)$$

$$\text{where } T_{[i]} = \max \left\{ p \left(1 + \sum_{k=i}^N Q_{[k+1]} \right)^{-m}, v \right\} + \mu \left(\sum_{k=i}^N T_{[k+1]} Q_{[k+1]} / \alpha \right)^\beta$$

Though scheduled by a backward scheduling approach, both learning and deterioration effects occurred initially with the batch firstly processed by the machine. Thus, by applying a backward index computation, Eq. (21) can be rewritten into:

$$Q_{[i]} = \left(n - \sum_{k=i+1}^N Q_k \right) / i + (1/2)(i+1) \left(s / T_{[i]} \right) - (s / T_{[i]})i, \quad i = N, \dots, 1 \quad (22)$$

Solving Eq. (22) and Eq. (11) simultaneously yields:

$$Q_{[i]} = \max \left\{ \left[\left(n - \sum_{k=i+1}^N Q_k \right) / i + (1/2)(i+1) \left(s / T_{[i]} \right) - (s / T_{[i]})i \right], 1 \right\}, \quad i = N, \dots, 1 \quad (23)$$

4.2. Determining the maximum number of batches

A maximum number of batches can be computed by assuming that the whole jobs in all batches are processed by using a minimum processing time. A minimum processing time is obtained under a condition of the existence of a learning effect. Thus, a minimum processing time formula can be expressed by the following equation:

$$T_{\min} = \max \left\{ p(1+n)^{-m}, v \right\} \quad (24)$$

Solving Eq. (15), Eq. (16), and Eq. (24) simultaneously yields:

$$N = 1 - (\vartheta - d + T_{\min} n) / s \quad (25)$$

Based on Eq. (17), it is found that the batch scheduled firstly should be finished coincide at the due date, and thus the value of $\vartheta=0$. Therefore, Eq. (25) can be rewritten into:

$$N = 1 + \lfloor (d - T_{\min} n) / s \rfloor \quad (26)$$

Solving Eq. (25) and Eq. (12) simultaneously yields:

$$N_{\max} = \min \left\{ 1 + \lfloor (d - T_{\min} n) / s \rfloor, n \right\} \quad (27)$$

4.3. Determining the Sequence of the Resulting Batches

Proposition 1 If there are N batches to process on a single machine with a constant setup time and the processing time of a batch is influenced by its position in the schedule, the minimum total actual flow time is found by scheduling the resulting batches by using the LPT rule in a backward scheduling approach.

Proof Suppose there are two feasible schedules for N batches. The first schedule places $(i)^{\text{th}}$ batch at a position of (i) and $(i+1)^{\text{th}}$ at a position of $(i+1)$. The second schedule differs from the first one only in the $(i)^{\text{th}}$ batch at a position of $(i+1)$ and $(i+1)^{\text{th}}$ at a position of (i) . The two schedules are then sequenced according to backward scheduling. If F^{a1} and F^{a2} are each total actual flow time of the first and the second schedule consecutively, then the following is found:

$$\begin{aligned} F^{a1} - F^{a2} &= \left(T_{[i]} Q_{[i]} Q_{[i+1]} + s Q_{[i+1]} \right) - \left(T_{[i+1]} Q_{[i+1]} Q_{[i]} + s Q_{[i]} \right) \\ F^{a1} \leq F^{a2} &\Leftrightarrow T_{[i]} + s / Q_{[i]} \leq T_{[i+1]} + s / Q_{[i+1]} \end{aligned}$$

The value of total actual flow time on the left will be minimum to that on the right if the batch size on the left ($Q_{[i]}$) is greater than that on the right ($Q_{[i+1]}$). Thus, to find an optimal value of a total actual flow time, the batches should be scheduled by the LPT rule in a backward scheduling approach.

4.4. The Heuristic Procedure

- Step 1: Set input parameters of $n, d, p, s, \delta, \mu, \alpha$, and β . Continue to Step 2.
- Step 2: Compute N_{\max} by Eq. (26) Set j as an index of the number of batches ($j=1, 2, \dots, N_{\max}$). Continue to Step 3.
- Step 3: Begin with $j=1$ Set $Q_{[1]}=n, T_{[1]}=p$. Continue to Step 4.
- Step 4: Compute $M=T_{[1]}Q_{[1]}$ If $M \leq d$, continue to Step 5; otherwise, there is no solution because the demand cannot be scheduled; then STOP.
- Step 5: Compute a total actual flow time ($F_{[1]}$) by Eq. (7). Set the value of $F_{[1]}^a$ as a temporary optimal solution (F^{a*}). Continue to Step 6.
- Step 6: Set an index of the number of batches $j=j+1$. If $1 < j \leq N_{\max}$, continue to Step 7; otherwise, set F^{a*} as the optimal solution and the STOP.
- Step 7: Set $N=j$. Use Eq. (23) to compute batch size ($Q_{[i]}$) and batch processing time ($T_{[i]}$) by using a backward index computation ($i=N, \dots, 1$). Continue to Step 8.
- Step 8: Based on Proposition 1, schedule the resulting by a non-increasing batches. Continue to Step 9.
- Step 9: Compute $M = \sum_{i=1}^N T_{[i]}Q_{[i]} + (N-1)s$. If $M \leq d$, continues to Step 10; otherwise, set F^{a*} as the optimal solution, and then STOP.
- Step 10: Compute a total actual flow time ($F_{[j]}$) by Eq. (7). Continues to Step 11
- Step 11: Compare $F_{[j]}^a$ and F^{a*} If $F_{[j]}^a < F^{a*}$, set $F_{[j]}^a$ as F^{a*} and then return to Step 6.

5. Numerical Experiences

5.1. Verification and validation test

A verification of the procedure is determined by a numerical example as follows. Supposed the parameter of data inputs: $p=0.5, n=3, d=6, s=1, v=0.1, \delta=0.9, \mu=0.2, \alpha=50, \beta=2$. The result of the procedure is presented in Table 1.

Table 1. The solution of heuristic procedure

N	i	$Q_{[i]}$	$T_{[i]}$	$B_{[i]}$	F^a
1	1	3	05	4.5	4.5
2	1	2	0.45	5.58	4.2*
	2	1	0.5	3.58	The optimal solution
3	1	1	0.42	5.58	
	2	1	0.45	4.13	
	3	1	0.5	2.63	

Table 1 shows that there are 3 combinations of integer batch sizes. The minimum total actual flow time is obtained on $N=2$ with $Q_{[1]}=2$ and $Q_{[2]}=1$. Henceforth, the procedure has been verified.

A validation test is determined by comparing the solution of the heuristic algorithm with the enumeration method based on integer composition technique. The enumeration solution is shown in Table 2.

Table 2: The Testing Results of The Enumeration Method

N	$Q_{[i]}, i=1, 2, \dots, N$	$T_{[i]}, i=1, 2, \dots, N$	F^a
1	(3)	(0.5)	4.5
2	(2,1)	(0.45, 0.5)	4.2 (optimal solution)
	(1,2)	(0.423, 0.5)	5.27
3	(1,1,1)	(0.5, 0.45, 0.423)	5.67

Table 2 showed that there are 4 combinations of integer batch sizes ($N=4$) that is (3), (2, 1), (1, 2), (1, 1, 1). A solution that produced an optimal solution is for $N=2$ with $Q_{[1]}=2$ and $Q_{[2]}=1$. It shows that the solution of enumeration method is same as the proposed algorithm. Henceforth, the solution of the proposed algorithm has been validated.

5.2. The sensitivity test

The sensitivity tests are given by input data of 10 different cases where the learning parameter (δ) and deterioration parameters (α) will be changed in each case. Case no 1 until 5 will investigate the sensitivity analysis of δ parameter and Case no 6 until 10 will investigate the sensitivity analysis of α parameter. The result of the test is presented in Table 3 as follows.

Table 3. The results of the proposed procedure.

Cases	Input Parameters									Performance of Solutions		
	n	d	p	s	v	δ	μ	α	β	F^a	N	$Q_{[i]}$
1	10	8	0.5	2	0.1	0.9	0.2	50	2.0	45.51	2	7, 3
2	10	8	0.5	2	0.1	0.8	0.2	50	2.0	36.72	3	6, 3, 1
3	10	8	0.5	2	0.1	0.7	0.2	50	2.0	31.31	3	6, 3, 1
4	10	8	0.5	2	0.1	0.6	0.2	50	2.0	27.31	3	6, 3, 1
5	10	8	0.5	0.5	0.1	0.5	0.2	50	2.0	24.05	3	6, 3, 1
6	10	8	0.5	0.5	0.1	0.9	0.2	40	2.0	45.55	2	7, 3
7	10	8	0.5	0.5	0.1	0.9	0.2	30	2.0	45.59	2	7, 3
8	10	8	0.5	0.5	0.1	0.9	0.2	20	2.0	45.7	2	7, 3
9	10	8	0.5	0.5	0.1	0.9	0.2	10	2.0	46.29	2	7, 3
10	10	8	0.5	0.5	0.1	0.9	0.2	8	2.0	46.73	2	7, 3

It can be observed in Table 3 that the decrease of learning rate (δ) or the increase of learning effect impact to increase the number of demands (N) and to decrease the total flow time. Meanwhile, the deterioration parameter (α) has a small impact so that the number of batches is same for all cases. The result of tests also shows that learning parameter has more significance impact than the deterioration parameter to the decision of the number of batches.

5.3. The effectiveness test

The effectiveness test is determined by comparing the solutions of the heuristic procedure with the optimal solution produced by enumeration procedure with integer composition technique. The explanation of the integer composition technique has been discussed in Shen and Evan [20]. The two procedures are coded in a programming language of Microsoft Visual C# 2013. The results of testing the two algorithms are then reported by using a computer processor of Intel Core i3 with a 6-GB Random Access Memory (2 GB used on 64-bit computer platform) and 1 GB Graphic Processor Unit. Table 4 presents the results of testing 10 cases generated randomly.

Table 4. The results of comparison test.

Demand	Integer Composition Procedure			Heuristic Procedure			Difference of Comparison Results		
	F^a	Length of computation (minute)	Memory usage (KB)	F^a	Length of computation (minute)	Memory usage (KB)	F^a (%)	Length of computation (%)	Memory usage (%)
10	31.8	2	0.34×10^3	31.8	< 1	5	0	100	6.7×10^3
12	33.01	8	1.59×10^3	33.01	< 1	8	0	700	19.7×10^3
15	96.04	57	15.4×10^3	96.04	< 1	12	0	5,600	128×10^3
11	33.74	4	0.74×10^3	33.74	< 1	7	0	300	10.4×10^3
15	69.17	68	15.4×10^3	69.17	< 1	12	0	6,700	128×10^3
18	113	451	144×10^3	113	< 1	14	0	45,000	$1,031 \times 10^3$
9	31.28	< 1	0.16×10^3	31.28	< 1	4	0	0	3.8×10^3
19	91.01	1051	30.3×10^3	91.22	< 1	20	0.23	105,000	151.6×10^3
16	114.12	114	32.5×10^3	114.39	< 1	14	0.24	11,300	232.4×10^3
29	-	-	o.o.m		< 1	32	-	-	-
Average difference results (%)							0.05	19,411	190.3×10^3
Max difference results (%)							0.24	105,000	$1,031 \times 10^3$
Minimum difference results							0	0	3.8×10^3
Standard deviations (%)							0.001	35,113	325.1×10^3

Table 4 showed that though the heuristic procedure does not guarantee to produce an optimal solution. However, the average of difference solution is 0.05% with 0.001% of standard deviation. It also can be observed that the proposed procedure is more efficient than the enumeration procedure for the need of memory and time to compute. Thus, the heuristic procedure can be applied to solve the problems in the real case.

6. Concluding Remarks

The present research dealt with batch scheduling on a single machine with processing times influenced by both learning, and deterioration effects simultaneously to minimize total actual flow time. This research develops the proposed heuristic procedure based on the Lagrange relaxation. Numerical experience tests show that the increase of learning parameter (δ) have a significant pact to increase the number of batches. However, the increase of deterioration parameter (α) has a small impact on determining the number of batches. We conclude that learning effect has more significance impact than deterioration effect. The comparison test result between the proposed procedure and enumeration procedure show that the heuristic procedure does not guarantee to produce the optimal solution, but the average of difference solution is 0.05%. Further research should search for solutions of batch scheduling problems with simultaneous effects of learning, forgetting, and deterioration effects.

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] C. Santos and M. Magazine, "Batching single operation manufacturing systems," *Operational Research Letter*, vol. 4, pp. 99–103, 1985.
- [2] G. Dobson, U. S. Karmarkar, and J. L. Rummel, "Batching to minimize flow times on one machine," *Management Science*, vol. 33, pp. 784–799, 1987.
- [3] A. H. Halim, S. Miyazaki, and H. Ohta, "Batch scheduling problem to minimize actual flow times of component through the shop under JIT environment.," *European Journal of Operational Research*, vol. 72, pp. 529–544, 1994.
- [4] A. H. Halim and H. Ohta, "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*, vol. 33, pp. 185–194, 1994.
- [5] C. N. Potts and M. Y. Kovalyov, "Scheduling with batching: A review," *European Journal of Operational Research*, vol. 120, pp. 228–249, 2000.
- [6] G. Mosheiov, D. Oron, and Y. Ritov, "Minimizing flow-time on a single machine with integer batch sizes," *Operational Research Letter*, vol. 33, pp. 497–501, 2005.
- [7] G. Mosheiov and D. Oron, "A single machine batch scheduling problem with bounded batch size," *European Journal of Operational Research*, vol. 187, pp. 1069–1079, 2008.
- [8] M.S Barketau, T.C.E. Cheng, C.T. Ng, V. Kotov, and M. Y. Kovalyov, "Batch scheduling of step deteriorating jobs," *Journal of Scheduling.*, vol. 11, pp. 17–28, 2008.
- [9] M. Ji and T. C. E. Cheng, "Batch scheduling of simple linear deteriorating jobs on a single machine to minimize makespan," *European Journal of Operational Research*, vol. 202, pp. 90–98, 2010.
- [10] B. Mor and G. Mosheiov, "Batch scheduling with step-deteriorating processing times to minimize flow time," *Naval Research Logistics*, vol. 59, pp. 587–600, 2012.
- [11] W.-H. Kuo and D.-L. Yang, "Single-machine group scheduling with a time-dependent learning effect," *Computers & Operations Research*, vol. 33, pp. 2099–2112, 2006.
- [12] W.-H. Yang, "A batching problem with learning effect considerations," *Asia-Pacific Journal Of Operational Research*, vol. 26, pp. 307–317, 2009.
- [13] R Yusriski, S. Sukoyo, T. M. A. A. Samadhi, and A. H. Halim, "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*, vol. 6, pp. 365–378, 2015.
- [14] L. . Yelle, "The learning curve: historical review and comprehensive survey," *Decision Science*, vol. 10, pp. 302–328, 1979.
- [15] S. Teyarachakul, S. Chand, and J. Ward, "Effect of learning and forgetting on batch sizes," *Production and Operations Management*, vol. 20, pp. 116–128, 2011.
- [16] T. P. Wright, "Factors affecting the cost of airplanes.," *Journal of Aeronautical Sciences.*, vol. 3, pp. 122–128, 1936.
- [17] M. Y. Jaber and M. Bonney, "Production break and the learning curve: The forgetting phenomenon," *Applied Mathematical Modelling*, vol. 20, pp. 162–169, 1996.
- [18] R Jiang and D. N. P. Murthy, *Maintenance, Decision Models for Management*. Science Press, 2008.
- [19] M. P. Kaminskiy and V. V. Krivtsov, "An integral measure of aging/rejuvenation for repairable and non-repairable systems.," *Reliability and Risk Analysis: Theory & Applications*, vol. 1, pp. 69–76, 2008.
- [20] H. Shen and D. J. Evans, "A new method for generating integer compositions in parallel.," *Parallel Algorithms Applications*, vol. 9, pp. 101–109, 1996.