A Model of Batch Scheduling for a Single Batch Processor with Additional Setups to Minimize Total Inventory Holding Cost of Parts of a Single Item Requested at Multi-due-date

Abdul Hakim Halim¹, Ernawati², Nita P A Hidayat³

¹Manufacturing Systems Research Group, Department of Industrial Engineering, Bandung Institute of Technology, Bandung, INDONESIA
²Department of Industrial Engineering, Sekolah Tinggi Teknik Cipasung, Tasikmalaya, INDONESIA
³Department of Industrial Engineering, Jenderal Achmad Yani University, Cimahi, INDONESIA

Email: ahakimhalim@ti.itb.ac.id

Abstract. This paper deals with a model of batch scheduling for a single batch processor on which a number of parts of a single item are to be processed. The process needs two kinds of setups, i.e., main setups required before processing any batches, and additional setups required repeatedly after the batch processor completes a certain number of batches. The parts to be processed arrive at the shop floor at the times coinciding with their respective starting times of processing, and the completed parts are to be delivered at multiple due dates. The objective adopted for the model is that of minimizing total inventory holding cost consisting of holding cost per unit time for a part in completed batches, and that in in-process batches. The formulation of total inventory holding cost is derived from the so-called actual flow time defined as the interval between arrival times of parts at the production line and delivery times of the completed parts. The actual flow time satisfies not only minimum inventory but also arrival and delivery just in times. An algorithm to solve the model is proposed and a numerical example is shown.

Keywords: batch scheduling, batch processor, actual flow time, inventory holding cost, additional setups,

1. Introduction

Batch scheduling problems to minimize total actual flow time have extensively been discussed in the literature and among the many papers are those written by [1–7]. In the literature, the actual flow time of a part is defined as an interval between arrival time of the part at the production line and delivery time of the completed part. [1] explains the definition of a batch as follows. Let there be a number of parts to be processed on a single machine and to be delivered at a common due date. If all parts arrive at once at a particular point of time, the total actual flow time of all the parts in the shop floor is measured from the arrival time to the common due date. This can be considered that the parts are processed in a single large batch. As it is assumed that parts can be controlled to be available at their respective starting times of processing, the parts can be split into several smaller batches. If this is the case, questions arising are how many batches that the parts should be divided, how many parts that each batch should contain, and how to schedule the resulting batches. The answers to the questions should be directed to achieve the objective of minimizing the total actual flow time. Accordingly, a batch is understood as a number of
parts processed in one setup, and each part in the batch waits (on the production facility) until the processing of all the parts in the batch is completed.

[5] show that batch scheduling problems deal not only with the case of job processors where all parts in a batch are processed sequentially part by part (meaning that batch processing time is the multiplication of the part processing time by the number of parts in the batch), but also with the case of batch processors where all parts in a batch are processed at once (meaning that batch processing time equals the part processing time, no matter how many parts in the batch are). It can also be observed from all the papers mentioned above that the setup required before processing each batch is anticipatory (meaning that the corresponding batch is not necessarily available on the machine when a setup is started) and given. It can be comprehended that the existence of this setup is because parts are processed in batches. In practical production lines, however, there are cases that another kind of setups is also required, that is, when scheduled events happen. For example, in an etching process, a setup is required not only before processing each batch but also every time when the acid used for the etching process should be replaced or added. The replacement or addition of the acid may be done after the batch processor completes a certain number of batches.

The problems that will be encountered in this paper is a model of batch scheduling for a single batch processor processing a number of parts of a single items with additional setups, where the parts arrive at the shop at the times coinciding with their starting times of processing, and the completed parts are to be delivered at multiple due dates. The objective adopted for the model is that of minimizing total inventory holding cost derived from the so-called actual flow time.

2. Total inventory holding cost

Suppose there are a number of parts, \( n \), of a single item to be processed in batches on a single batch processor. The parts can be controlled to arrive at the production line at their respective starting times of processing, and the completed parts are to be delivered at a common due date. The processing time of each part is \( t \), and the setup time required before processing any batch is \( s \). If all parts to be processed have been partitioned into \( N \) batches and the batches are backwardly sequenced started from the common due date, moving to time zero on a time scale, the actual flow time for the batches, \( F_{[i]}^b \), can be formulated as follows:

\[
F_{[i]}^b = d - B_{[i]}, \quad i = 1, 2, 3, \ldots N, \tag{1}
\]

where \( d \) represents a common due date, and \( B_{[i]} \) is starting time for processing the batch scheduled at the \( i \)th position counted from \( d \). The actual flow time of parts in the batch, \( F_{[i]} \), can be formulated as follows:

\[
F_{[i]} = (d - B_{[i]})Q_{[i]}, \quad i = 1, 2, 3, \ldots N, \tag{2}
\]

or

\[
F_{[i]} = \{\sum_{j=1}^{i} (t + s) - s\} Q_{[i]}, \quad i = 1, 2, 3, \ldots N, \tag{3}
\]

where \( Q_{[i]} \) is the number of parts in the batch scheduled at the \( i \)th position counted from the common due date. [5] have formulated the total actual flow time, \( F \), of all parts through the shop with a batch processor as follows:

\[
F = \sum_{i=1}^{N} \{\sum_{j=1}^{i} (t + s) - s\} Q_{[i]} \tag{4}
\]

where \( N \) is the number of batches processed in the shop floor. It should be emphasized here that the actual flow time adopts the so-called backward scheduling, instead of the forward scheduling. The first mentioned approach sequences batches backwardly from the end position on a time scale, i.e., from the due date, moving to time zero, while the later approach sequences batches forwardly from time zero, appearing a mirror image of the backward schedule.
Now, if the process requires additional setups, $a$ (assumed to be fixed and given), performed when it completes a certain number of batches, in addition to the main setups, $s$, Equation (4) can be rewritten to be the followings:

$$F = \sum_{i=1}^{N} \left( \sum_{j=1}^{l_i} (t + s + a_{ij}) - (s + a_{i[1]}) \right) Q_{i[i]}$$

with

$$a_{i[j]} = \begin{cases} a & \forall i = N, (N - x), (N - 2x), \ldots, (N - (m - 1)x) \\ 0 & \text{otherwise} \end{cases}$$

where $a_{i[j]}$, $x$ and $m$ respectively denote an additional setup required before processing the batch at the $i$th position on a schedule, the number of batches between two consecutive additional setups, and the ratio of $N$ to $x$ ($m = N/x$). If $m$ is not an integer, $m$ is set to be the minimum integer greater than or equal to $N/x$ (a rounding up of $N/x$). Note that as both main setups and additional setups are assumed anticipatory, any setup for a batch is not included in the actual flow time of the batch but is included in the actual flow time of the batch processed previously.

If some batches completed before the common due date, the parts in such batches will incur inventory cost. If we follow the condition discussed in [1], distinguishing handling cost per unit time for a part in completed batches, denoted as $c_1$, from that in process batches, denotes as $c_2$, the total inventory handling cost, $T$, can be formulated as follows:

$$T = c_1 \left( \sum_{i=1}^{N} \left( \sum_{j=1}^{l_i} (t + s + a_{ij}) - (s + a_{i[1]}) \right) Q_{i[i]} \right) - (c_1 - c_2) \left( \sum_{i=1}^{N} t Q_{i[i]} \right)$$

Now, if there are $r$ numbers, $n_h$ of parts demanded at respective due dates, $d_h, h = 1, 2, 3, \ldots, r$ (note that $d_r$ is the closest due date to time zero), let divide the whole scheduling period into $r$ partitions of periods each of which constitutes an interval, $H_h, h = 1, 2, 3, \ldots, r$, between two consecutive due dates. It can be seen that the batch scheduling problem becomes, firstly, how to determine the numbers, $D_h$, of parts processed in period $H_h, h = 1, 2, 3, \ldots, r$, secondly, how to batch $D_h$ within $H_h$, and thirdly, how to schedule the resulting batches in periods $H_h$. The total inventory handling cost, $T$, can be formulated as follows:

$$T = c_1 \left[ \sum_{h=1}^{N_h} \left( \sum_{i=1}^{l_i} (t + s + a_{h[i]}) - (s + a_{h[i]}) \right) Q_{h[i]} \right]$$

$$+ c_1 \left( \sum_{g=1}^{r-1} \sum_{h=g+1}^{N_h} (D_h - n_h) (d_g - d_{(g+1)}) \right)$$

$$- (c_1 - c_2) \sum_{h=1}^{N_h} \sum_{i=1}^{l_i} t Q_{h[i]}$$

where $N_h, a_{h[i]}$, and $Q_{h[i]}$ respectively represent the number of batches processed in period $H_h$, an additional setup performed before the batch scheduled in the $i$th position in period $H_h$, and the number of parts in the batches scheduled in period $H_h, h = 1, 2, 3, \ldots, r$.

3. Model formulation

The model of batch scheduling for a single batch processor with additional setups (BPAS) can be formulated as follows.

Minimize

$$T = c_1 \left[ \sum_{h=1}^{N_h} \left( \sum_{i=1}^{l_i} (t + s + a_{h[i]}) - (s + a_{h[i]}) \right) Q_{h[i]} \right]$$

$$+ c_1 \left( \sum_{g=1}^{r-1} \sum_{h=g+1}^{N_h} (D_h - n_h) (d_g - d_{(g+1)}) \right)$$

$$- (c_1 - c_2) \sum_{h=1}^{N_h} \sum_{i=1}^{l_i} t Q_{h[i]}$$
Subject to

\[ a_{h[i]} = \begin{cases} a & \forall i = N_h, (N_h - x), (N_h - 2x), \ldots, (N_h - (m_h - 1)x) \\ 0 & \text{otherwise} \end{cases} \quad \forall h = 1, 2, \ldots, r \quad (8) \]

\[ N_h t + (N_h - 1)s + \sum_{i=1}^{N_h-1} a_{h[i]} \leq H_h \quad \forall h = 1, 2, \ldots, r \quad (9) \]

\[ \sum_{h=g+1}^{r}(D_h - n_h) \geq 0 \quad \forall g = 1, 2, \ldots, (r - 1) \quad (10) \]

\[ \sum_{i=1}^{N_h} Q_{h[i]} = D_h \quad \forall h = 1, 2, \ldots, r \quad (11) \]

\[ B_{h[1]} + t = d_h \quad \forall h = 1, 2, \ldots, r \quad (12) \]

\[ 0 \leq Q_{h[i]} \leq k \quad D_h > 0 \quad N_h \geq 1 \quad \forall h = 1, 2, \ldots, r; \forall i = 1, \ldots, N_h \quad (13) \]

Constraint (8) shows that additional setups are conducted repeatedly after the batch processor completes \( x \) batches, where \( m_h \) is a rounding up of \( N_h/x \). Constraint (9) shows that the total processing time of parts distributed into period \( H_h \) must be no longer than the length of the period. Constraint (10) describes that the total number of parts processed within a time interval from periods \( H_r \) to \( H_h \) must be greater than or equal to the demanded parts, \( n_h \), at \( d_h \). However, Constraint (11) describing that the total number of parts processed within the scheduling period must be the same as the total number of parts distributed to all periods, \( H_h, h = 1, 2, \ldots, r \), in the scheduling period must be satisfied. Constrain (12) accomplishes material balances in period \( H_h \) respectively. Constrain (13) shows the batches processed last in respective periods \( H_h, h = 1, 2, \ldots, r \), must be completed at \( d_h \). Constrain (14) indicates that all batch sizes (\( \forall i, h \)) must be greater than or equal to zero and less than or equal to the capacity, \( k \), of the batch processor, and that the number of parts distributed to period \( H_h \) must be positive and the number of batches in respective periods \( H_h \) must be greater than or equal to 1.

4. Algorithm

The following is a proposed algorithm for solving Model [BPAS], developed from the algorithm discussed in [5].

Step 0. Set values of parameters \( t, s, a, x, k, c_1, c_2, n_h \) and \( d_h \) for \( h = 1, 2, \ldots, r \).

Step 1. Set \( h = 1, K_0 = 0 \) and \( d_{r+1} = 0 \), and go to Step 2.

Step 2. Calculate \( Y_h = n_h + K_{h-1} \) and \( N_h = Y_h/k \), and go to Step 3.

Step 3. If \( N_h \) is an integer, then set \( N_h = N_h' \) and \( Q_{h[i]} = k \) for \( i = 1, 2, \ldots, N_h' \). Go to Step 4. Otherwise, set \( N_h = \text{rounding up of } N_h' \); then set \( Q_{h[i]} = k \) for \( i = 1, 2, \ldots, (N_h - 1) \) and \( Q_{h[N_h]} = Y_h - (N_h - 1)k \). Go to Step 4.

Step 4. Backwardly sequence the resulting batches in order of non-increasing batch sizes: \( Q_{h[1]} \geq Q_{h[2]} \geq \cdots \geq Q_{h[N_h]} \). Go to Step 5.

Step 5. Compute \( Z_h' = \frac{d_h - d_{h-1}}{(t + s - a)} \). Go to Step 6.

Step 6. If \( Z_h' \) is an integer, then set \( Z_h = Z_h' \). Go to Step 7. Otherwise, set \( Z_h = \text{rounding up of } Z_h' \). Go to Step 7.

Step 7. If \( N_h > Z_h \) then set \( N_h = Z_h \) and sequence the bastes in order of non-increasing batch sizes: \( Q_{h[1]} \geq Q_{h[2]} \geq \cdots \geq Q_{h[N_h]} \), and go to Step 8. Otherwise go to Step 8.

Step 8. Compute \( D_h = \sum_{i=1}^{N_h} Q_{h[i]} \); \( Y_h = n_h - D_h \) and \( m_h = N_h/x \). Go to Step 9.

Step 9. If \( m_h \) is an integer, then set \( m_h = m_h' \), and go to Step 10. Otherwise, set \( m_h = \text{rounding up of } m_h' \), and go to Step 10.

Step 10. Set \( a_{h[i]} = a \) for \( \forall i = N_h, (N_h - x), (N_h - 2x), \ldots, (N_h - (m_h - 1)x) \), and otherwise set \( a_{h[i]} = 0 \). Go to Step 11.
Step 11. For \( i = 1, 2, \ldots, N_h \), compute \( FL_{h[i]}^B = \sum_{j=1}^{i} (t + s + a_{h[i]}) - (s + a_{h[i]}) \) and set the time interval between \( d_{h+1} \) and the starting setup time before processing \( b_{h[N_h]} \). \( IT_h \), as the following \( IT_h = B_{h[N_h]} - (t + s + a_{h[N_h]}) - d_{h+1} \). Go to Step 12.

Step 12. If \( IT_h \geq 0 \), then compute the objective value (T), and stop. Otherwise, set \( N_h = N_h - 1 \), and go to Step 11.

5. Numerical experience and analysis

Suppose there is a shop floor with a batch processor producing parts of a single item requested at 3 due dates where the demands at respective due dates are shown in Table 1. The processing time per part is 100, setup time required before processing each batch is 10, and additional setup times required after the processor completes 2 batches is 15, and the capacity of the batch processor is 20.

### Table 1. The numbers of parts demanded at respective due dates

| \( h \) | 1   | 2   | 3   |
|--------|-----|-----|-----|
| \( n_h \) | 95  | 100 | 95  |
| \( d_h \) | 2000| 1525| 920 |

The resulting solution obtained from applying the proposed algorithm is shown in Table 2, and the Gantt chart for the solution is shown in Figure 1. Table 2 shows that the scheduling period is divided into 3 periods, \( H_1 \), \( H_2 \), and \( H_3 \), and the numbers of batches in respective periods are 4, 5, and 6. Table 2 also shows all batch sizes and the positions where the additional setups take places. Other values of decision variables shown in Table 2 are those of actual flow times of respective batches, \( F_{h[i]} \), the starting times of processing batches, \( B_{h[i]} \) and total flow time of parts processed in each period, \( F_h \). The total actual flow time, \( F \), of parts processed in the shop is 95,600 with the total holding cost, \( T \), for all parts through the shop is 181200.

The algorithm applies that if the numbers, \( D_h \), of parts distributed into periods \( H_h \), \( h = 1, 2, 3, \ldots, r \), have been decided, the problems in respective periods are to batch \( D_h \) and to schedule the resulting batches. The problem in each period constitutes a problem of batch scheduling in the condition of a common due date, is solved separately.

### Table 2. The solution for the generated problem

| \( H \) | \( N_h \) | 1 | 2 | 3 |
|--------|--------|---|---|---|
| \( i \) | 4      | 5 | 6 |
| \( Q_{h[i]} \) | 20     | 20 | 20 | 20 |
| \( s \) | 100    | 100 | 100 | 100 |
| \( a_{h[i]} \) | 0      | 15 | 0  | 15 |
| \( F_{h[i]}^B \) | 100    | 210 | 335 | 445 |
| \( B_{h[i]} \) | 1900   | 1790 | 1665 | 1555 |
| \( F_{h[i]} \) | 2000   | 4200 | 6700 | 8900 |
| \( IT_h \) | 5      | 10 | 215 |
| \( F_h \) | 21800  | 33800 | 40000 |
| \( F \) | 95600  |
| \( T \) | 181200 |
From Table 2 and Figure 1, it can be observed that additional setups are required before processing the batch scheduled at the first position in period $H_2$, $b_{21}$, and the batch scheduled at the forth position in period $H_3$, $b_{34}$, even though both batches are sequenced consecutively. When an additional setup is conducted, the chemicals (acid for an etching process) used in current processing will be replaced or added even though this action has not been necessary, as in this numerical example, the additional setup is required if the batch processor has completed 2 batches. Accordingly, the solution shown in Table 2 may be modified to accommodate this condition.

The modified solution is shown in Table 3 and Figure 2. Comparing the two solutions, the modified solution has made additional setups be conducted every time after the batch processor completes 2 batches. However, the modified solution leads to higher inventory holding cost. As the objective is to minimize holding cost, the solution for the problem is the original one, even though the chemicals currently used should be replaced or added.

| $H$ | 1 | 2 | 3 |
|-----|---|---|---|
| $N_H$ | 4 | 5 | 6 |
| $t^*$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| $Q_{b[i]}$ | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| $P_{b[i]}$ | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| $a_{b[i]}$ | 15 | 0 | 15 | 0 | 15 | 0 | 15 | 0 | 15 | 0 | 15 | 0 | 15 | 0 | 15 |
| $F_{b}$ | 22400 | 33800 | 40000 |
| $F$ | 96200 |
| $T$ | 181500 |

**Table 3. The modified solution**

**Figure 1.** Gantt chart for the resulting solution
The objective is to minimize total holding cost derived from the so-called actual flow time. To formulate the problem, the whole scheduling period is partitioned into periods each of which constitutes a time interval between two consecutive due dates. The proposed algorithm solves the problem period by period where in each period the problem is considered as a problem with a common due date, and the algorithm is effective to solve the problem.

6. Concluding remarks

This paper has dealt with a problem of batch scheduling for a batch processor on which a number of parts are processed. The processor requires not only main setups conducted before processing any batches but also additional setups conducted after the batch processor completes a number of batches. The objective is to minimize total holding cost derived from the so-called actual flow time. To formulate the problem, the whole scheduling period is partitioned into periods each of which constitutes a time interval between two consecutive due dates. The proposed algorithm solves the problem period by period where in each period the problem is considered as a problem with a common due date, and the algorithm is effective to solve the problem.

References
[1] Halim A H 1993 Batch scheduling for production systems under just in time environment Doctorate Dissertation (Japan: University of Osaka Prefecture)
[2] Halim A H and Ohta H 1993 Batch-scheduling problems through the flowshop with both receiving and delivery just in time Int. J. Prod. Res. 31 1943–55
[3] Halim A H, Miyazaki S and Ohta H 1994 Batch-scheduling problems to minimize actual flow times of parts through the shop under JIT environment Eur. J. Oper. Res. 72 529–44
[4] Halim A H, Miyazaki S and Ohta H 1994 Lot scheduling problems of multiple items in the shop with both receiving and delivery just in time Prod. Plan. Control 5 175–84
[5] Hidayat N P A, Cakravastia A, Samadhi T M A A and Halim A H 2016 A batch scheduling model for m heterogeneous batch processor Int. J. Prod. Res. 54 1170–85
[6] Yusriski R, Sukoyo S, Samadhi T and Halim A 2015 Integer batch scheduling problems for a single-machine with simultaneous effect of learning and forgetting to minimize total actual flow time Int. J. Ind. Eng. Comput. 6 365–378
[7] Zahedi Z, Samadhi Tmaa, Suprayogi S and Halim A 2016 Integrated batch production and maintenance scheduling for multiple items processed on a deteriorating machine to minimize total production and maintenance costs with due date constraint Int. J. Ind. Eng. Comput. 7 229–244