International Journal of Technology 13(4) 816-826 (2022)
Received May 2021 / Revised June 2021 / Accepted September 2021

International Journal of Technology

http://ijtech.eng.ui.ac.id

# Single Item Batch-scheduling Model for a Flow Shop with *m* Batch-processing Machines to Minimize Total Actual Flow Time

Abdul Hakim Halim[1], Nita Puspita Anugrawati Hidayat[2*], Wisnu Aribowo[1]

[1]*Department of Industrial Engineering, Bandung Institute of Technology, Jl. Ganesa 10 Bandung 40132, Indonesia*
[2]*Department of Industrial Engineering, Bandung Islamic University, Jl. Tamansari no.1 Bandung 40132, Indonesia*

**Abstract.** We propose a batch-scheduling model to minimize the total actual flow time (TAF) of parts to be processed in a flow shop consisting of *m* batch-processing machines. A batch-processing machine (BPM) is a machine that can process several parts at once, and the TAF of parts is the total interval time from the arrival times to the corresponding due date. In the real world, shop floors often have production lines with BPMs and multistage processes. We were motivated by a real problem in the aircraft industry and aimed to simultaneously satisfy the due dates and minimize the total time that parts spend in the shop. The problem was formulated as a mathematical model and solved using a proposed algorithm. The batch-scheduling problem is divided into batching and scheduling subproblems. The solution has been obtained by adopting backward scheduling. This paper develops a new model of flow shop scheduling problem for the shop with batch processing machines and the heuristic solution method. It provides numerical examples and their results to demonstrate the effectiveness of the proposed algorithm for solving the problem.

*Keywords:* Batch processor; Batch scheduling; Flow shop; Total actual flow time

## 1. Introduction

A batch can be defined as several parts sharing the same setup, and the parts can be processed either on a job-processing machine (abbreviated as JPM) or on a BPM. The difference between a JPM and a BPM lies in how they process parts in a batch. A JPM individually processes all parts in a batch sequentially, whereas a BPM processes them simultaneously. This research focuses on the shop floor with BPMs found in many industries, such as that used for burn-in operations in the semiconductor industry, heating and pressure operations in aeronautical manufacturing, and hardening and soaking processes in automobiles gear manufacturing, and drying operations in the lumber industry.

Several researchers have discussed flow shop scheduling problems. Gong et al. (2010) addressed a flow shop scheduling in steel manufacturing consisting of a soaking pit as the first stage with BPM and a rolling mill as a second stage with a JPM. The ingots (parts) in a batch will remain in the soaking pit if the rolling mill is processing another ingot, during

* Corresponding author's email: nita.anugrawati@unisba.ac.id, Tel.: (+62) 22 4264065
doi: 10.14716/ijtech.v13i4.4869

which the soaking pit will be blocked. Fu et al. (2012) also considered the blocking constraint in a flow shop scheduling problem with two stages where the first stage is a BPM, and the second stage with a JPM follows it. The buffer between consecutive machines is limited; thus, the completed batch will keep the BPM blocked if the buffer between straight machines is full. Chen et al. (2014) also considered the blocking constraint, particularly for the flow shop scheduling problem with two BPMs with dynamic job arrivals at the first BPM.

Liao and Huang (2011) have developed a batch scheduling model for a flow shop consisting of two BPMs with unlimited buffer capacity, and the objective is makespan minimization. To solve the problem, they created a heuristic procedure based on a Tabu search. It is shown that the heuristic is effective for solving scheduling problems with relatively many jobs. Matin et al. (2017) dealt with issues of BPMs flow shop where the parts in a batch and its batch size may change when the batch is processed on different machines. Rosi et al. (2013) developed a hybrid model of a flow shop for a sterilization plant to minimize makespan and the number of tardy jobs. They aim to reduce the number of tardy jobs (surgical kits) because late jobs will cause the surgery to be rescheduled, which brings heavy medical and economic consequences. The makespan is considered an objective because a lower makespan results in lower idle time and higher machine utilization and efficiency. On the other hand, Gokhale and Mathirajan (2011), Chou and Wang (2012), Peres and Monch (2013) have used due-date as a performance measurement. Utama et al. (2019) proposed a flow shop scheduling model to minimize energy consumption in scheduling jobs on each machine and adopted a new hybrid meta-heuristic for solving the problems.

However, those research adopted the forward scheduling approach that may violate the due dates of jobs, and precisely represents what customers need. Many manufacturing companies intend to satisfy the due dates and minimize inventory. For fulfilling the due date, we should adopt the backward-scheduling approach, starting from the due dates of jobs and moving backward until the job is released. Meanwhile, the so-called actual flow time, defined as an interval from the arrival time of a part to the due date, was used as an objective for batch-scheduling (abbreviated as BS) problems in Halim and Ohta (1993). They showed that minimizing TAF of parts in the shop minimizes the total time that the parts spend in the shop and guarantees that the delivery of the completed parts always meets the due date. Note that TAF is based on the backward-scheduling approach. This objective was applied for various cases of BS problems (Surjandari et al., 2015; Yusriski et al., 2016; Maulidya et al., 2020), but the research pieces were for shop floors with JPMs. In reality, in many cases, the shop floor constitutes production lines with BPMs and multistage processes.

Hidayat et al. (2013) adopted the TAF and the backward scheduling approach for a BS problem with a single BPM processing part of a single item where the completed parts should be sent simultaneously on $d$ as a due date. The model proposed by Hidayat et al. (2013) was developed further from different viewpoints: to tackle the condition of $m$ heterogeneous BPMs by distributing parts to each of the $m$ heterogeneous BPMs (Hidayat et al., 2014); to handle multiple due dates by distributing parts into all periods between two consecutive due dates (Hidayat et al., 2016); and to handle the condition of multiple items (Hidayat et al., 2018). These researches focus on single-stage scheduling issues but demonstrate how BPM challenges differ from JPM challenges.

However there is a need to develop a model of BS problems for a flow shop with BPMs to minimize the TAF. Such a model will extend the single-stage BS model with a BPM discussed in Hidayat et al. (2013 and 2016) to handle flow-shop processing parts of a single item delivered on $d$ as a due date. Under the premise that the due date is far enough off to produce workable schedules, it is possible to split the problem of BS for a flow shop with BPM into two subproblems that are solved concurrently. The first subproblem is determining the number of batches and the size of each batch. The second sub-problem is the scheduling of the resulting batches.

## 2. Experimental

We were motivated by a real problem in the aircraft industry to propose a BS model for a flow shop with $m$ BPMs. The problem is as follows: Let there be $n$ ordered parts of a single item where the completed parts should be sent on a due date, $d$. Each part should be processed through $m$ operations performed on corresponding BPMs. Thus, the parts are to be processed in a flow shop with $m$ BPMs, $BPM_1 - BPM_2 - \cdots - BPM_m$ in this machine order. Each BPM simultaneously processes $c$ parts in a batch, where the part-processing times on a machine are $t_1$, $t_2$,..., $t_m$ respectively where $t_1 \neq t_2 \neq \cdots \neq t_m$, and the setup time required before processing equals $s$.

Assumptions for the proposed model are: (1) all of the parts can be arrived at the starting time of batch processing, (2) during the scheduling period, all of BPMs are available, and (3) no rejection of the processed parts and the completed parts are always conforming. Figure 1 shows the problem for a flow shop with $m$ BPMs.

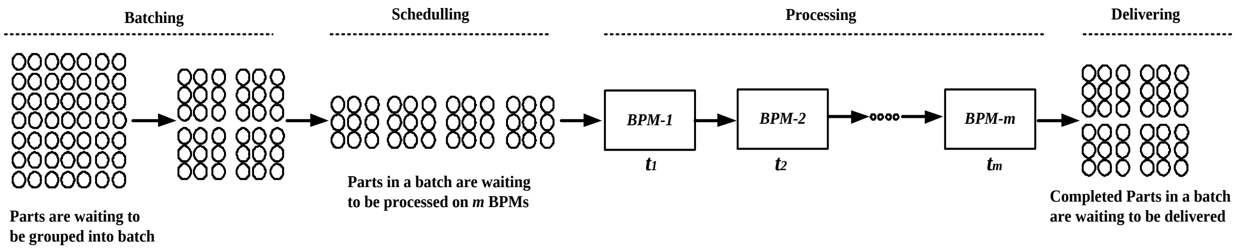


**Figure 1** Problem for a flow shop with $m$ BPMs

The notations used in the proposed models are as follows.

Indices

$i$ : The index denoting the positions of a batch on a production schedule.
$k$ : The index that identifies the machine number in the shop, $k = 1,2, \ldots, m$

Parameters

$d$ : The common due date
$s_k$ : The setup time before processing a batch on $BPM_k$
$s_{k(i)}$ : The setup time before processing a batch $b_{k(i)}$ on $BPM_k$
$t_k$ : The processing time of each part on $BPM_k$
$t_{k(i)}$ : The processing time of batch $b_{k(i)}$

Variables

$B_{k(i)}$ : The starting time of the processing batch $b_{k(i)}$
$C_{k(i)}$ : The completion time of batch $b_{k(i)}$
$FL^a_{k(i)}$ : The actual flow time of batch $b_{k(i)}$
$N$ : The number of batches in the shop
$Q_{k(i)}$ : The size of batch $b_{k(i)}$
$F^a$ : TAF of parts through the shop

The flow shop model with a single item and an expected due date (Model FsCdd) is shown in Figure 2.

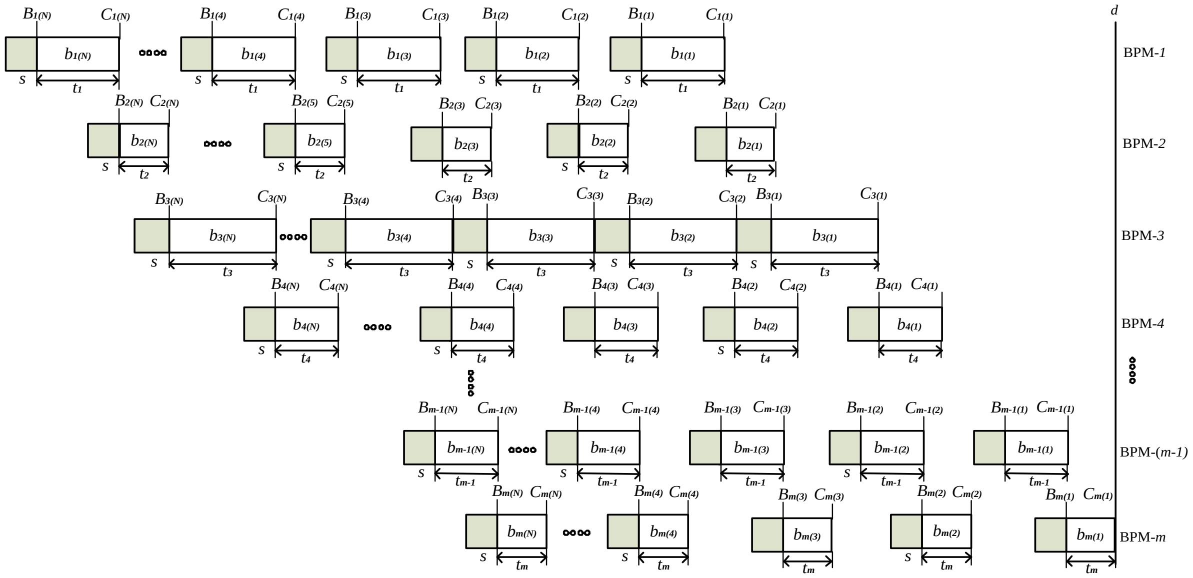


**Figure 2** Flow shop of $m$ BPMs with $N$ batches

Batches $b_{k(N)}$, for $k$ = 1,2,..., $m$, are the first batches processed on the corresponding $BPM_k$ (Figure 2). The first operations of each batch $b_{1(i)}$, for $i$=1, 2,..., $N$, are processed on $BPM_1$, and so on until the last operations of each batch $b_{m(i)}$ on $BPM_m$. Since an identical sequence of batches is assumed for each $BPM$, it is sufficient to determine the TAF of the batches only on $BPM_1$. For only $BPM_1$, the Gantt chart will be identical to that for the single-stage BS problem where the completed parts are sent on a due date, $d$ (Figure 3).

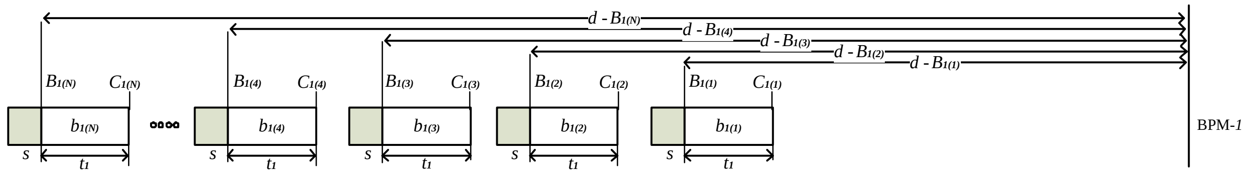


**Figure 3** Batch-scheduling problem with $N$ batches on $BPM_1$

From Figure 3, the TAF of each batch can be determined as follows.

$$FL^a_{1(i)} = d - B_{1(i)} \; for \; i = 1, 2, \dots, N$$

The sizes of respective batches are defined as $Q_i$, for $i$= 1, 2, ..., $N$; then, TAF of all parts through the shop is formulated as follows:

$$F^a = \sum_{i=1}^{N} \left(d - B_{1(i)}\right) Q_i \tag{1}$$

The following conditions must be satisfied to minimize the TAF of all parts through the flow shop with $m$ BPMs:

(1) The completion time of the batch $b_{m(1)}$ on the last machine $BPM_m$ should exactly coincide with the due date, $C_{m(1)} = d$ and $C_{m(1)} = B_{m(1)} + t_m$, thus, $B_{m(1)} + t_m = d$.

(2) Processing of batch $b_{m(1)}$ on $BPM_m$ should start immediately after the processing of batch $b_{m-1(1)}$ on $BPM_{m-1}$ is completed. Thus $B_{k(1)} = B_{k+1(1)} + t_k$ for $k$ = 2, 3,..., ($m$-1).

(3) Processing a batch on a machine can be started if the previous processing is completed and the machine is not currently processing any batch. Thus $B_{k(i)} = Max\left\{B_{k-1(i)} + t_{k-1}\, ; B_{k(i+1)} + t_k + s\right\}$ for $i$ = 1, 2,..., ($N$-1); $k$ = 2, 3,..., $m$ and $B_{k(N)} = B_{k-1(N)} + t_{k-1}$ for $k$ = 2, 3,..., $m$.

(4) $BPM_1$ is the machine that performs the first operation for all batches. If $BPM_1$ has completed an operation, then $BPM_1$ will be ready to set up and process the next batch. Thus, $B_{1(i)} \geq B_{1(i+1)} + t_1 + s$ for $i$ = 1, 2,..., ($N$-1) and $B_{1(N)} \geq 0$.

In this study, the parts are assumed to arrive at the scheduled starting time of processing; all machines are always available, and there is no defect in the processes. The formulation of Model FsCdd is as follows.

<u>Model FsCdd</u>

Minimize

$$F^a = \sum_{i=1}^{N}\left(d - B_{1(i)}\right)Q_i \quad (2)$$

subject to

$$B_{m(1)} + t_m = d \quad (3)$$

$$B_{k(1)} = B_{k+1(1)} + t_k \quad \forall k = 2, 3, \dots, (m-1) \quad (4)$$

$$B_{k(i)} = \text{Max}\left\{B_{k-1(i)} + t_{k-1}\, ; B_{k(i+1)} + t_k + s_k\right\} \quad \forall i = 1, \dots, (N-1), \quad \forall\, k = 2, \dots, m \quad (5)$$

$$B_{k(N)} = B_{k-1(N)} + t_{k-1} \quad \forall k = 2, 3, \dots, m \quad (6)$$

$$B_{1(i)} \geq B_{1(i+1)} + t_1 + s_1 \quad \forall i \ = 1, 2, \dots, (N-1) \quad (7)$$

$$B_{1(N)} \geq 0 \quad (8)$$

$$\sum_{i}^{N} Q_{k(i)} = n \quad \forall k = 1,2, \dots, m \quad (9)$$

$$0 < Q_{k(i)} \leq c \ \ \forall\, i = 1,2, \dots, N;\ \forall k = 1,2, \dots, m \quad (10)$$

$$N \geq 1 \quad (11)$$

Equation (3) shows that the completion time of the last batch processed on the last machine must coincide with the due date, $d$. Equation (5) determines the start time of all other batches. A batch can only be started on $BPM_k$ after it is completed on the previous machine and after the batch previously processed on $BPM_k$ is completed. As soon as a batch is completed, the processing on the next machine can be started, and so on for the other subsequent batches until the last machine $BPM_m$, as shown in Equation (6). Equation (7) determines the relationship between the start time of consecutive batches on the first machine $BPM_1$. Equation (8) guarantees the schedule feasibility, that is, the start time of the batch processed first on $BPM_1$ will be on after time zero. Equation (9) ensures the material balance in the shop. Equations (10) and (11) restrict the batch sizes and the number of batches to positive values.

## 3. Results and Discussion

### *3.a. Solution Procedure*

The problem in this study is complicated because we must simultaneously solve batching and scheduling subproblems—the former involves determining the number of batches and the size of each batch and the latter determining the schedule of processing the resulting batches in a flow shop. Hence, a heuristic procedure is necessary. The TAF of the parts through the shop will be minimized if the solutions of both subproblems are minimized, considering the problem's constraints.

In a BPM, the number of parts in a batch is limited by capacity; thus, the maximum batch size is $c$, where $c$ is the BPM's capacity. The batch size must be an integer, so the minimum batch size equals 1. The number of resulting batches will be minimum if the batch size is maximized, as $N = n/c$, where *n* is the total demand. Again, as the number of batches must be an integer, if it is not an integer, then $n/c$ must be rounded up. The number of resulting batches will be maximum if the batch size is minimum, $N = n/1$. The number of resulting batches cannot exceed the total demand ($n$).

Theorem: *The scheduling period has N batches, and the batch sizes equal $Q_i$ (i = 1, 2, ..., N). The batches are to be processed in a flow shop with m BPMs (k = 1, 2, ..., m), $BPM_1 - BPM_2 - \cdots - BPM_m$ in this machine order. The setup time required before processing equals s, and the processing time on each BPM are $t_1$, $t_2$,..., $t_m$ where $t_1 \neq t_2 \neq \cdots \neq t_m$. The highest processing time is $t_z$ corresponding to $BPM_z$ (k = z). The completed parts should be sent on a due date, d. With backward scheduling, $F^a$ is minimized if and only if the batches are arranged in a non-increasing order of batch sizes, ranging from the position closest to the due date, d:*

$$Q_1 \geq Q_2 \geq Q_3 \geq \cdots \geq Q_N$$

Proof:

Equation (2) can be rewritten as Equation (12).

$$F^a = \sum_{k=1}^{m} t_k \sum_{i=1}^{N} Q_i + (s_z + t_z) \sum_{i=2}^{N} (i-1) Q_i \quad (12)$$

Equation (12) shows that $F^a$ will be minimum if $\sum_{k=1}^{m} t_k \sum_{i=1}^{N} Q_i$ and $(s_z + t_z) \sum_{i=2}^{N} (i - 1)Q_i$ are minimum. The values of parameter setup time ($s_k$and $s_z$) and the processing time ($t_k$ and $t_z$) are fixed and known. The variable decision is the size of each batch, $Q_i$, and it is clear that the batch arrangement does not affect the value of $\sum_{i=1}^{N} Q_i$. Increasing the value of *i* will increase the value of (*i*-1). It is clear that $\sum_{i=2}^{N} (i - 1)Q_i$ will be the minimum only if $Q_i$ decreases. This means that the batch sizes must be in a non-increasing order. Thus, a minimum $F^a$ can be obtained if the batch size, $Q_i$ decreases with increasing *i*. In other words, the batch that must be placed at the position closest to the due date (*i*=1) is the most giant batch, and the size continues to decrease until the last position (*i*=*N*). Thus, the following order is obtained:

$$Q_1 \geq Q_2 \geq Q_3 \geq \cdots \geq Q_N$$

If the number of batches and the size of each batch are determined, the remaining problem is deciding the schedule for processing the resulting batches in the flow shop. This study differs from the classical flow shop studies in that we choose TAF as the objective and adopt the backward-scheduling approach. The next step is to generate a schedule, i.e., to determine the starting and completion times of each batch on each *BPM* based on the backward-scheduling approach, provisions of the flow shop, and highest operating time. In backward scheduling, the schedule determination starts from batch $b_{m(1)}$ , i.e., the completion time of the first operation (*i* = 1) on the last machine (*k* = *m*) must coincide with the due date. Thus, the batch schedule$b_{m(1)}$ is obtained, i.e., $C_{m(1)} = d$ and $B_{m(1)} = C_{m(1)} - t_m$.

Then, the first operating schedule for the previous *BPM* is determined from *k* = (*m*-1) until *k* = 1, referring to the flow shop provisions based on backward scheduling: $C_{k(1)} = B_{k+1(1)}$ and $B_{k(1)} = C_{k(1)} - t_k$ for *k* = 1, 2,..., (*m*-1). Subsequently, the schedule on $BPM_z$, which is the machine with the highest processing time, is determined. For $BPM_z$, there will be no waiting time and idle time. Hence, $C_{ki} = B_{k(i-1)} - s$ and $B_{ki} = C_{ki} - t_k$, for *k* = z and *i* = 2, 3,..., *N*. Schedules on other *BPM* will refer to $BPM_z$ schedule.

<u>Algorithm for the FsCdd model:</u>

Step:

0. Set *n, c, m, s*, and *d*.
   For *k* = 1, 2, 3, ..., *m,* set $t_k$
1. Calculate $N' = \frac{n}{c}$ go to 2
2. If *N'* is an integer, then *N* = *N'*, go to 3. Otherwise, *N*= rounding up of *N'* batches, then go to 3
3. Arrange *N* batches according to Theorem, and $Q_1 \geq Q_2 \geq \cdots \geq Q_N$. Go to 4.
4. Set *k* = *m*, *i* = 1, and $C_{ki} = d$. Calculate $B_{ki} = C_{ki} - t_k$, and go to 5.
5. Set *k*=*k-1* and $C_{ki} = B_{(k+1)i}$, calculate $B_{ki} = C_{ki} - t_k$, go to 6.
6. If *k=1*, then go to 7. Otherwise, go to 5.
7. If $t_k$ is the highest processing time, then set *z* =*k,* and go to 9. Otherwise, go to 8.
8. Set *k*=*k+1*, and go to 7.
9. Set *i*=*i+1*; calculate $C_{ki} = B_{k(i-1)} - s$ and $B_{ki} = C_{ki} - t_k$, and go to 10.
10. If *i*=*N*, then go to 11; otherwise, go to 9.
11. If *z=1*, then set *k*=1 and go to 12. Otherwise, go to 15.
12. Set $C_{ki} = B_{(k+1)i}$, calculate $C_{(k+1)i} = B_{(k+1)i} + t_{(k+1)}$, and go to 13.
13. If *i=2*, then go to 14, otherwise set *i*=*i*-1, and go to 12.
14. If *k*=(*m-1)*, then go to 25, otherwise set *k*=*k*+1; *i*=*N*, and go to 12.
15. If *z*=*m*, then set *k*=*m*, go to 16. Otherwise, go to 19.
16. Set $B_{ki} = C_{(k-1)i}$; compute $B_{(k-1)i} = C_{(k-1)i} - t_{(k-1)}$, and go to 17.
17. If *i=2*, then go to 18, otherwise set *i*=*i*-1, and go to 16.
18. If *k=2*, then go to 25. Otherwise, set *k*=*k*-1; *i*=*N*, and go to 16.
19. Set $C_{ki} = B_{(k+1)i}$; calculate $C_{(k+1)i} = B_{(k+1)i} + t_{(k+1)}$, go to 20.
20. If *i=2*, then go to 21. Otherwise, set *i*=*i*-1, and go to 19.
21. If *k*=(*m-1)*, then set *k*=*z* and *i*=*N,* go to 22. Otherwise, set *k*=*k*+1 and *i*=*N,* and go to 19.
22. Set $B_{ki} = C_{(k-1)i}$; calculate $B_{(k-1)i} = C_{(k-1)i} - t_{(k-1)}$, and go to 23.
23. If *i=2*, then go to 24. Otherwise, set *i*=*i*-1, and go to 22.
24. If *k=2*, then go to 25. Otherwise, set *k*=*k*-1; *i*=*N*, and go to 22.
25. Calculate $F^a$, and STOP.

*3.b. Numerical Examples*

Let there be 70 parts of a single item to be processed on $BPM_1, BPM_2, BPM_3$, and $BPM_4$ in this machine order. The capacity of each BPM is 20 parts, the setup time on each BPM is equal to 1, and the due date equals 200*.* Herein, we present three numerical examples that differ in the processing times on a BPM (Table 1).

**Table 1** Processing time on a BPM ($t_k$)

| Case No. | *k* | | | | The highest processing time |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 1. | 20 | 10 | 15 | 5 | $t_1 = 20$ |
| 2. | 5 | 20 | 10 | 15 | $t_2 = 20$ |
| 3. | 15 | 10 | 20 | 5 | $t_3 = 20$ |

For the proposed algorithm, there are four batches where the batch sizes are $Q_1 = Q_2 = Q_3 = 20$ and $Q_4 = 10$, respectively. Table 2 lists the starting and completion times of batches on each BPM. Figure 4 shows the Gantt chart for Case 3.

**Table 2** Results of the calculation using the proposed algorithm for the FsCdd model

| | | | |
|---|---|---|---|
| Case-1: $t_1 = 20; t_2 = 10; t_3 = 15; t_4 = 5; d = 200, n = 70, c = 20$ and $s = 1$ | | | |
| $C_{41} = 200; B_{41} = 195$ | $C_{31} = 195; B_{31} = 180$ | $C_{21} = 180; B_{21} = 170$ | $C_{11} = 170; B_{11} = 150$ |
| $C_{42} = 179; B_{42} = 174$ | $C_{32} = 174; B_{32} = 159$ | $C_{22} = 159; B_{22} = 149$ | $C_{12} = 149; B_{12} = 129$ |
| $C_{43} = 158; B_{43} = 153$ | $C_{33} = 153; B_{33} = 138$ | $C_{23} = 138; B_{23} = 128$ | $C_{13} = 128; B_{13} = 108$ |
| $C_{44} = 137; B_{44} = 132$ | $C_{34} = 132; B_{34} = 117$ | $C_{24} = 117; B_{24} = 107$ | $C_{14} = 107; B_{14} = 87$ |
| $F^a = \sum_{i=1}^{4}(d - B_{1i})\,Q_i = (200 - 150)20 + (200 - 129)20 + (200 - 108)20 + (200 - 87)10 = 5.390$ | | | |
| Case-2: $t_1 = 5; t_2 = 20; t_3 = 10; t_4 = 15; d = 200, n = 70, c = 20$ and $s = 1$ | | | |
| $C_{41} = 200; B_{41} = 185$ | $C_{31} = 185; B_{31} = 175$ | $C_{21} = 175; B_{21} = 155$ | $C_{11} = 155; B_{11} = 150$ |
| $C_{42} = 179; B_{42} = 164$ | $C_{32} = 164; B_{32} = 154$ | $C_{22} = 154; B_{22} = 134$ | $C_{12} = 134; B_{12} = 129$ |
| $C_{43} = 158; B_{43} = 143$ | $C_{33} = 143; B_{33} = 133$ | $C_{23} = 133; B_{23} = 113$ | $C_{13} = 113; B_{13} = 108$ |
| $C_{44} = 137; B_{44} = 122$ | $C_{34} = 122; B_{34} = 112$ | $C_{24} = 112; B_{24} = 92$ | $C_{14} = 92; B_{14} = 87$ |
| $F^a = \sum_{i=1}^{4}(d - B_{1i})\,Q_i = (200 - 150)20 + (200 - 129)20 + (200 - 108)20 + (200 - 87)10 = 5.390$ | | | |
| Case-3: $t_1 = 15; t_2 = 10; t_3 = 20; t_4 = 5;\ d = 200, n = 70, c = 20$ and $s = 1$ | | | |
| $C_{41} = 200; B_{41} = 195$ | $C_{31} = 195; B_{31} = 175$ | $C_{21} = 175; B_{21} = 165$ | $C_{11} = 165; B_{11} = 150$ |
| $C_{42} = 179; B_{42} = 174$ | $C_{32} = 174; B_{32} = 154$ | $C_{22} = 154; B_{22} = 144$ | $C_{12} = 144; B_{12} = 129$ |
| $C_{43} = 158; B_{43} = 153$ | $C_{33} = 153; B_{33} = 133$ | $C_{23} = 133; B_{23} = 123$ | $C_{13} = 123; B_{13} = 108$ |
| $C_{44} = 137; B_{44} = 132$ | $C_{34} = 132; B_{34} = 112$ | $C_{24} = 112; B_{24} = 102$ | $C_{14} = 102; B_{14} = 87$ |
| $F^a = \sum_{i=1}^{4}(d - B_{1i})\,Q_i = (200 - 150)20 + (200 - 129)20 + (200 - 108)20 + (200 - 87)10 = 5.390$ | | | |

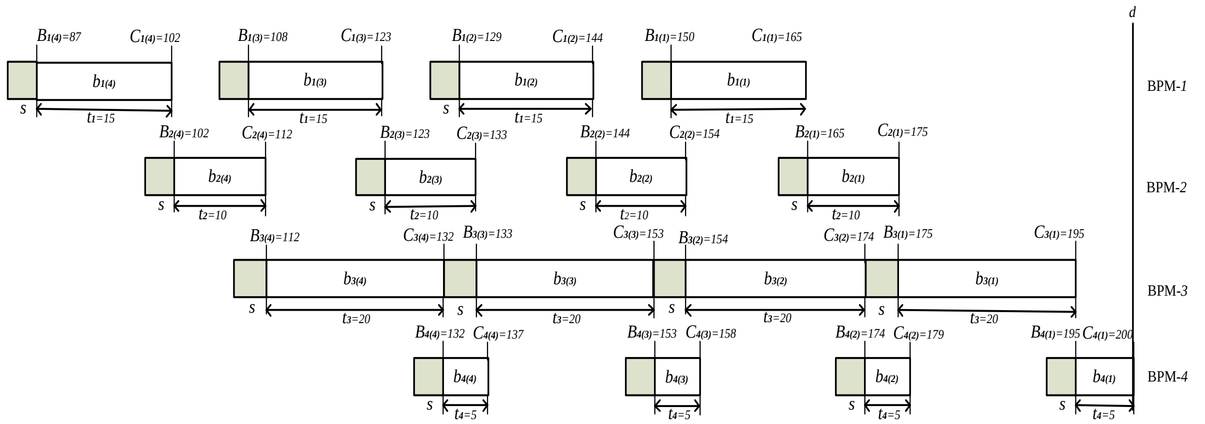


**Figure 4** Gantt chart of case-3

From Table 2, we can see that the machine with the highest processing time ($BPM_z$) will require the most production time. Therefore, time is essential at $BPM_z$, as indicated by the absence of idle time at $BPM_z$. In all cases, every time $BPM_z$ finishes processing a batch,

$BPM_z$ immediately sets up and processes the next batch. In contrast to other BPMs that experience idle time (e.g., $BPM_4$ in case-3 experiences idle time of 15—the difference in processing time between $BPM_3$ and $BPM_4$, i.e., $t_3 - t_4$ =15). The idle time at $BPM_2$ is 10, the difference in processing time $BPM_3$ with $BPM_2$, ($t_3 - t_2$ =10). When the idle time at $BPM_1$ is 5, that is, the difference in processing time between $BPM_1$ and $BPM_2$ ($t_1 - t_2 = 5$).

In all cases, the waiting time appears only at $BPM_4$ which carries out the last process in the flow shop sequence. Waiting time occurs in batches $b_{42}, b_{43}$, and $b_{44}$ because $C_{42}, C_{43}$ and $C_{44}$ are not equal to $d$. At $BPM_1$, $BPM_2$ and $BPM_3$, there is no waiting time because it is assumed that the batch arrivals at $BPM_1$ can be arranged at the time when the parts perform the first operation on $BPM_1$. Then, after finishing processing on $BPM_1$, the second process on $BPM_2$, $C_{1i} = B_{2i}$, is immediately performed, and so on until the last processing. This is possible because of the backward-scheduling approach.

Although the processing time configuration of each $BPM$ differs among cases, the value of $F^a$ in all cases is identical, i.e., 5,390. This happens because demand ($n$) and BPM ($c$) capacity in each case are equal. Hence, for the batching subproblem, we get the same batch number and the batch size. From Theorem, identical batching results yield identical batch arrangements in each case, namely, $Q_1 = Q_2 = Q_3 = 20$ and $Q_4$ =10, and $B_{1(i)}$ All cases are the same—$B_{1(1)}$ =150, $B_{1(2)}$ =129, $B_{1(3)}$ =108, and $B_{1(4)}$=87. Equation (2) shows that minimum $F^a$ is only determined by the decision variables $Q_i$ and $B_{1(i)}$. Equation (12) shows that the value $F^a$ is the sum of $\sum_{k=1}^{m} t_k \sum_{i=1}^{N} Q_i$ and $(s_z + t_z) \sum_{i=2}^{N}(i-1)Q_i$. Thus, it is clear that $F^a$ is identical across the cases.

*3.c. Model Validation*

We tested the model using Lingo software, and we used several sets of data with arbitrary selection. This was done considering that some researchers have discussed BS problems for flow shops with BPMs. However, any previous work has not adopted the objective of minimizing the TAF and the backward-scheduling approach. From the test, in all cases, the number of formed batches is minimum as $N = n/c$ and it is an integer. The number of batches must be an integer; hence, if it is not an integer, $n/c$ is rounded up. The batch arrangement satisfies Theorem and the value of $F^a$ is minimum, equals to Equation (12). Table 3 lists the results of the data calculation.

**Table 3** Calculation results of hypothetical data

| No | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $s_k$ | $n$ | $c$ | $N$ | $Q_i$ | $F^a$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1. | 8 | 19 | 15 | 16 | 4 | 76 | 15 | 6 | $Q_i$ =15 for i=1,..., 5; $Q_6 = 1$ | 7,973 |
| 2. | 10 | 11 | 30 | 11 | 2 | 87 | 11 | 8 | $Q_i$ =11 for i=1,..., 7; $Q_8 = 10$ | 15,026 |
| 3. | 22 | 21 | 15 | 10 | 3 | 79 | 17 | 5 | $Q_i$ =17 for i=1,..., 4; $Q_5 = 11$ | 9,022 |
| 4. | 20 | 30 | 22 | 15 | 1 | 82 | 18 | 5 | $Q_i$ =18 for i=1,..., 4; $Q_5 = 10$ | 11,722 |
| 5. | 23 | 27 | 28 | 26 | 4 | 88 | 18 | 5 | $Q_i$ =18 for i=1,..., 4; $Q_5 = 16$ | 14,656 |
| 6. | 5 | 16 | 28 | 23 | 2 | 88 | 12 | 8 | $Q_i$ =12 for i=1,..., 7; $Q_8 = 4$ | 14,736 |
| 7. | 14 | 6 | 24 | 15 | 3 | 74 | 10 | 8 | $Q_i$ =10 for i=1,..., 7; $Q_8 = 4$ | 10,792 |
| 8. | 23 | 21 | 17 | 20 | 3 | 69 | 16 | 5 | $Q_i$ =16 for i=1,..., 4; $Q_5 = 5$ | 8,605 |
| 9. | 30 | 12 | 19 | 24 | 4 | 75 | 10 | 8 | $Q_i$ =10 for i=1,..., 7; $Q_8 = 5$ | 14,705 |
| 10. | 27 | 22 | 30 | 18 | 1 | 76 | 16 | 5 | $Q_i$ =16 for i=1,..., 4; $Q_5 = 12$ | 11,836 |

## 4. Conclusions

The BS problem for the $m$-BPM flow shop with the backward-scheduling approach was solved by dividing it into batching and scheduling subproblems. To attain $F^a$ minimum, the solutions of two subproblems must be minimal. In the batching subproblem, the number of resulted batches $(N)$ is minimum when $N$ equals the rounded-up value of $n/c$. Rounding up is done if $n/c$ is not an integer so that there is one batch with size $n$- $(N$-1) $c$ and ($N$-1) batches where the batch sizes are $c$. On the other hand, if $n/c$ is an integer, all batch sizes equal $c$. The solution in the scheduling subproblem will be minimum if the sequencing of $N$ resulting batches follows the Theorem; i.e., $Q_1 \geq Q_2 \geq \cdots \geq Q_N$ . The starting and completion times of each batch on each BPM are determined in three simple steps. First, the completion time of the last batch processed on the last machine ($b_{m1}$) must be equal to the due date, $C_{m1} = d$ and $B_{m1} = C_{m1} - t_m$ . Second, completion time batch $b_{(m-1)1}$ processed on $BPM_{(m-1)}$ must be determined. In the flow shop, the sequence determined using the backward-scheduling approach ensures that $C_{(m-1)1} = B_{m1}$ and $B_{(m-1)1} = C_{(m-1)1} - t_{(m-1)}$ and so on until $C_{11} = B_{21}$ and $B_{11} = C_{11} - t_1$. Third, the schedule at $BPM_z$ which has the highest processing time, must be determined, and $C_{z2} = B_{z1} - s$ and $B_{z2} = C_{z2} - t_3$, and so on are obtained until $C_{zN} = B_{z(N-1)} - s$ and $B_{zN} = C_{zN} - t_z$. Finally, $C_{ki}$ and $B_{ki}$ schedules for $i$ = 2, 3,..., $N$ for $k$ = 1,2,..., $m$, are determined referring to the $BPM_z$ schedule, flow shop provisions, and the backward-scheduling approach. These three steps were devised based on backward scheduling. The proposed model is limited to problems of single item parts with an common due date. Further research can be to cover the condition of multiple items demanded at multiple -due dates.

## Acknowledgments

This work was supported by funding from Direktorat Riset dan Pengabdian Masyarakat, Deputi Bidang Penguatan Riset dan Pengembangan, Kementerian Riset dan Teknologi/ Badan Riset dan Inovasi Nasional Republik Indonesia. ID Proposal: 112de821-2cea-4b30-908c-477a9d95f273.

## References

Chen, H., Zhou S., Li, X., Xu, R., 2014. A Hybrid Differential Evolution Algorithm for A Two-Stage Flow Shop on Batch Processing Machines with Arbitrary Release Times and Blocking. *International Journal of Production Research*, Volume 52(19), pp. 5714–5734

Chou, F.D., Wang, H.M., 2012. Minimizing Total Weighted Tardiness on Parallel Batch-processing Machine Scheduling Problems with Varying Machine Capacities. *Applied Mechanics and Materials*, Volume 110, pp. 3906–3913

Fu, Q., Sivakumara, A.I., Li, K., 2012. Optimization of Flow-shop Scheduling with Batch Processor and Limited Buffer. *International Journal of Production Research*, Volume 50(8), pp. 2267–2285

Gokhalea, R., Mathirajan, M., 2011. Heuristic Algorithms for Scheduling of a Batch Processor in Automobile Gear Manufacturing. *International Journal of Production Research,* Volume 49(10), pp. 2705–2728

Gong, H., Tang, L., Duin, C.W., 2010. A Two-Stage Flowshop Scheduling Problem on a Batching Machine and a Discrete Machine with Blocking and Shared Setup Times. *Computers & Operations Research,* Volume 37(5), pp. 960–969

Halim, A.H., Ohta, H., 1993. Batch-scheduling Problems Through the Flow Shop with Both Receiving and Delivery Just in Time. *International Journal of Production Research*, Volume 31(8), pp. 1943–1955

Hidayat, N.P.A., Cakravastia, A., Samadhi, T.M.A., Halim, A.H., 2013. A Single Item Batch Scheduling Model on a Batch Processor to Minimize Total Actual Flowtime of Parts through the Shop. *In:* Proceeding Asia Pacific Industrial Engineering and Management System, Cebu, Philippines

Hidayat, N.P.A., Cakravastia, A., Samadhi, T.M.A., Halim, A.H., 2014. A Batch-scheduling Problem to Minimize Total Actual Flow time of Parts through the Shop which Has *m* Heterogenous Batch Processors. *In:* Proceedings Asia Pacific Industrial Engineering and Management System, Jeju Island, South Korea

Hidayat, N.P.A., Cakravastia, A., Samadhi, T.M.A., Halim, A.H., 2016. A Batch Scheduling Model for *m* Heterogeneous Batch Processor. *International Journal of Production Research*, Volume 54(4), pp. 1170–1185

Hidayat, N.P.A., Cakravastia, A., Samadhi, T.M.A., Halim, A.H., 2018. Multi-Items Batch Scheduling Model for a Batch Processor to Minimize Total Actual Flow time of Parts through the Shop. *Jurnal Teknik Industri*, Volume 20(1), pp. 73-88

Liao, L.M., Huang, C.J., 2011. Tabu Search Heuristic for Two-Machine Flowshop with Batch Processing Machines. *Computers & Industrial Engineering*, Volume 60(3), pp. 426–432

Matin, H.N., Salmasi, N., Shahvari, O., 2017. Makespan Minimization in Flow Shop Batch Processing Problem with Different Batch Compositions on Machines. *International Journal of Production Economics*, Volume 193, pp. 832–844

Maulidya, R., Wangsaputra, R., Halim, A.H., 2020. A Batch Scheduling Model for a Three-stage Hybrid Flowshop Producing Product with Hierarchical Assembly Structures. *International Journal of Technology*, Volume 11(3), pp.608–618

Dauzère-Pérès, S., Mönch, L., 2013. Scheduling Jobs on a Single Batch Processing Machine with Incompatible Job Families and Weighted Number of Tardy Jobs Objective. *Computers & Operations Research,* Volume 40(5), pp. 1224–1233

Rossi, A., Puppato, A., Lanzetta, M., 2013. Heuristics for Scheduling A Two-Stage Hybrid Flow Shop with Parallel Batching Machines: Application at a Hospital Sterilisation Plant. *International Journal of Production Research*, Volume 51(8), pp. 2363–2376

Surjandari, I., Rachman, A., Purdianta, D.A., Dhini, A., 2015. The Batch Scheduling Model For Dynamic Multi-Item, Multi-Level Production in an Assembly Job Shop with Parallel Machines. *International Journal of Technology*, Volume 1, pp. 84–96

Utama, D.M., Widodo, D.S., Wicaksono, W., Ardiansyah, L.R., 2019. A New Hybrid Metaheuristics Algorithm for Minimizing Energy Consumption in The Flow Shop Scheduling Problem. *International Journal of Technology,* Volume 10(2), pp. 320–331

Yusriski, R., Samadhi, T.M.A.A., 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. *In:* IOP Conference Series: Materials Science and Engineering, Volume 114(1), p. 012073