

Multi-Items Batch Scheduling Model for a Batch Processor to Minimize Total Actual Flowtime of Parts through the Shop

Nita PA Hidayat^{1*}, Andi Cakravastia², TMA Ari Samadhi² and Abdul Hakim Halim²

Abstract: This study is inspired by a batch scheduling problem in metal working industry which guarantees to satisfy a due date. The actual flowtime adopts the backward scheduling approach and considers the due date. Using the actual flowtime as the objective, means that the solution is oriented to satisfy the due date, and simultaneously to minimize the length of time of the parts spending in the shop. This research is to address a problem of scheduling batches consisting of multiple items of parts processed on a batch processor where the completed parts must be delivered several time at different due dates. We propose an algorithm maximize the utilization of the batch processor and to schedule the resulting batches in backwardly non increasing batch size.

Keywords: Backward scheduling, batch scheduling, batch processor, batch size, actual flowtime.

Introduction

A number of parts processed simultaneously with sharing a setup time could be defined as a batch, and a batch processor is a machine that processes one batch at a time. Today the batch processor is encountered in many different environments, such as burn-in operations in semiconductor industry (Cheng, *et al.* [1]), heating and pressure operations in aeronautical industry (Malapert, *et al.* [2]), hardening and soaking operations in automobile gear manufacturing (Gokhalea and Mathirajan [3]), and drying operations in lumber industry (Gaudreault, *et al.* [4]). The batch processor is also used in iron and steel industry for heating ingots up to a proper processing temperature (Gong, *et al.* [5]), microbiological laboratory for producing agar that is used in food testing (Chakhlevit, *et al.* [6]), and bicycle rim manufacturing (Damodaran, *et al.* [7]). Those pieces of research work have addressed problems of batch scheduling on the batch processor.

The solution for batch scheduling problems on a batch processor is determined through two phases, that are batching and scheduling. Decisions on the batch size are made in the batching phase i.e., grouping the parts into some batches, while arranging the resulting batches into a specific sequence is performed in the scheduling phase. The size of a batch represents a number of parts in the batch, and it is limited by capacity of the batch processor. Grouping parts in the

batching phase should be based on a certain rule. Guo, Chengtao [8] has grouped parts based on the order of processing machines which is required by the parts. While Dazère-Peres and Monch [9] and Chakhlevitch *et al.* [6] have grouped parts based on the type of family of the parts, all parts in the same batch must come from the same type of family.

The entire parts in the same batch are processed simultaneously during a certain period of time called batch processing time, although each part requires a certain processing time which may differ from others. In Cheng *et al.* [1] and Condotta *et al.* [10] the entire parts in the batch have the same processing time, and the processing time of batch equals the processing time of parts. While in Noroozi, *et al.* [11], Behnamian *et al.* [12], Khasan, *et al.* [13] and Parsa, *et al.* [14] the processing time for each part is different. They set the processing time of batch equals the longest processing time of parts in the batch because they assume that an excessive duration of processing the parts could not lead to any defect. On the contrary, in Bellanger, *et al.* [15] each part has the processing time and a certain limit of processing time. They allow the real processing time of parts exceed its processing time as long as it does not exceed its certain limit of processing time because the excessive processing time from its limit may cause any defects. Therefore, only parts whose processing time intervals intersect can be formed into a batch.

Those studies adopt the forward scheduling approach, i.e., sequencing jobs starts from the time zero, then moving forward to the due date until all jobs have been scheduled. The schedule is feasible, but there is no guarantee it could meet the due date. On the other hand, Halim *et al.* [16] defined the actual flowtime of parts as the time that the parts spend in the shop from the starting time of processing until

¹ Faculty of Technology, Department of Industrial Engineering, Universitas Jendral Achmad Yani, Jl. Terusan Gatot Subroto, P.O BOX 807, Bandung 40285, Indonesia. Email: nita.ph@gmail.com.

² Faculty of Industrial Technology, Department of Industrial Engineering, Jl. Ganesha 10, Bandung 40132, Indonesia. Email: andi@mail.ti.itb.ac.id, asamadhi@mail.ti.itb.ac.id, ahakimhalim@gmail.com.

* Corresponding author

their due date as the delivery time of the processed parts, and the actual flowtime adopts a backward scheduling approach i.e., sequencing jobs starts from the due date then moving backward to the time zero until all jobs have been scheduled. The backward scheduling approach can satisfy the due date, but it may lead to an infeasible schedule. This research is inspired by a batch scheduling problem in metal working industry which guarantees to satisfy the due date as a commitment to its customer, thus adopting backward scheduling approach is a must. It is assumed that the resulted schedule is always feasible, and the infeasible schedule case will be discussed in future research.

The actual flowtime has been applied to problems of batch scheduling by Halim *et al.* [17, 18, 19], Sukoyo *et al.* [20] and Zahedi *et al.* [21] but they conduct batch scheduling research for job processors i.e., the machine that processes one job at a time. Hidayat *et al.* [22, 23] develop the actual flowtime of parts for batch scheduling problems on batch processors, and adopt it as an objective where parts to be processed are a single item with a common due date. Those model is further developed by Hidayat *et al.* [24] into a condition where the parts should be delivered at different due dates. Halim *et al.* [25] also develop the actual flowtime of parts for batch scheduling problems on batch processors into a condition where a batch could have different set up time from those of the other batches.

This research is concerned on batch scheduling problems at the coating stage performed on a batch processor in metal working industry. The length of time for the coating process is determined by coating thickness of part. When the parts are processed less or more than the time required for coating, then the part will not in accordance with the product specifications. Therefore, only parts which have the same processing time can be grouped into the same batch. We define that the parts which have the same requirement of coating thickness are compatible and have the same type of item. Therefore, grouping parts into batches should be based on the same type of item. The metal working industry produces some different products and demanded at different due dates. Each part requires a certain coating thickness depending on the product that will be formed by the parts, so that at the coating stage, there are some parts require different thickness and demanded at different due dates. Accordingly, we need to develop the model in Hidayat, *et al.* [24] into a condition of multiple items of parts with multiple due dates (abbreviated as MIMD). The model is developed in two steps; the first step is that the single item is developed into multiple items of parts with a common due date, and at the

second step the developed model is further developed into multiple items of parts with multiple due dates.

Methods

The MIMD problem can be explained as follows. Let there be numbers, $n_1, n_2, n_3, \dots, n_r$ of multiple items parts demanded at due dates, $d_1, d_2, d_3, \dots, d_r$ respectively. The parts must be processed on a batch processor, and t_g is the part processing time of item g . The batch processor needs to be set up before processing a batch, the set up times depends on the item type of parts in the batch, and s_g is the set up time for item g . The set up time is independent to the sizes of batches and the position of batches in the shop. The problems are both to determine batch size of the multiple items parts and to sequence the formed batches so as to minimize the total actual flowtime of parts through the shop (F^a).

The following the symbols and notations are used in this paper.

Indexes

- g : part item, $g=1, 2, \dots, k$
- h : time interval in a scheduling period, $h = 1, 2, \dots, r$. It is counted from the end position on a time scale.
- i : position of a batch on a production schedule which is counted from the end position on a time scale.

Sets

- b_i : set of parts in a batch sequenced at position i ($i = 1, 2, \dots, N$)
- b_{hi} : set of parts in a batch sequenced at position i ($i=1, 2, \dots, N_h$) within interval H_h

Parameters

- c : capacity of batch processor
- d : common due date
- d_h : the h^{th} due date
- H_h : time period defined as:
 $H_h = d_h - d_{h+1}$ for $h = 1, \dots, (r - 1)$
 $H_h = d_h$ for $h = r$
- k : number of item types
- m : number of machine
- n_g : number of parts item g demanded at common due date
- n : total part demanded at common due date, $n = \sum_{g=1}^k n_g$
- n_{hg} : number of parts item g demanded at due date d_h
- n_h : number of parts demanded at due date d_h , $n_h = \sum_{g=1}^k n_{hg}$
- r : number of intervals in the whole scheduling period

- s_i : setup time of batch sequenced at position i
- s_{hi} : setup time of batch sequenced at position i within interval H_h
- s_g : setup time of item g
- t_i : processing time of batch sequenced at position i
- t_{hi} : processing time of batch sequenced at position i within interval H_h
- t_g : part processing time of item g
- Variables**
- B_i : starting time of processing batch sequenced at position i
- B_{hi} : starting time of processing batch sequenced at position i within interval H_h
- D_h : number of parts processed within interval H_h
- F^a : total actual flowtime of parts through the shop
- G_i : type of item part in the batch sequenced at position i
- K_h : number of shortage parts within interval H_h
- N_g : number of batches which consisting of part item g
- N : number of batches in the whole scheduling period
- N_{hg} : number of batches which consisting of part item g within interval H_h
- N_h : number of batches within interval H_h , $N_h = \sum_{g=1}^k N_{hg}$
- Q_i : number of parts in batch sequenced at position i
- Q_{hi} : number of parts in batch sequenced at position i within interval H_h
- A_{hg} : number of parts item g requested to be processed within interval H_h
- Z_{ig} : binary variable equals 1 if parts in the batch sequenced at position i are members of item g , otherwise 0
- Z_{hig} : binary variable equals 1 if parts in the batch sequenced at position i within interval H_h are members of item g , otherwise 0

Actual Flowtime as a Performance Measurement

The actual flowtime of parts is defined as the time required by parts to be on a shop. It is measured from the processing starting times to their due date or to their delivery time (Halim, *et al.* [16]). In an ideal condition, the part spends in the shop when it is processed. So that the part's actual flowtime equals to the part processing time. It can be happened if the part

arrives exactly at its processing starting time and delivers right after completing the process. It is assumed that the arrival of parts as a batch can be setup exactly at its processing starting time. The actual flowtime of parts will be higher than its processing time if those parts are completed before their due date. In this case, the parts require longer time on the shop, since they have to wait until they deliver to the customer at the due date. Therefore, minimizing the actual flowtime of parts is essentially minimizing the waiting time before the delivery time (due date). Hence, the time duration spends by those parts in the shop is minimized.

Using the definition in Halim, *et al.* [16], FP_{ij}^a denotes the actual flowtime of part P_j in batch b_i , and P_j is j^{th} part in batch b_i , then the actual flowtime of part P_j is as follows.

$$FP_{ij}^a = d - B_i \quad j = 1, 2, \dots, b \quad i = 1, 2, \dots, N \quad (1)$$

The actual flowtime of batch b_i (denoted by Fb_i^a) states the length of time the batch b_i spends on the shop from its arrival time until its due date (see. equation (2))

$$Fb_i^a = d - B_i \quad i = 1, 2, \dots, N \quad (2)$$

A concept of the actual flowtime for parts of single item demanded on a common due date have been developed in Hidayat, *et al.* [22]. Figure 1 shows the sequence of the N batches processed on a single batch processor. Fb_i^a is determined in equation (3), and the total actual flowtime of batches through the shop (expressed by FS^a) is determined in equation (4).

$$Fb_i^a = \{\sum_{j=1}^i (t_j + s_j)\} - s_i \quad (3)$$

$$FS^a = \sum_{i=1}^N \{\sum_{j=1}^i (t_j + s_j)\} - s_i \quad (4)$$

The actual flowtime of each part in the same batch equals the actual flowtime of batch, and total actual flowtime of parts in a batch is a multiplication of the actual flowtime of the batch by the number of parts in the batch. The total actual flowtime of parts through the shop for a single item with a common due date on a batch processor is determined in equation (5).

$$F^a = \sum_{i=1}^N \{\sum_{j=1}^i (t_j + s_j) - s_i\} Q_i \quad (5)$$

Model Development

The first model was developed for multiple items of parts with a common due date problem (MICD). The MICD is a development of a single item with common due date problem (abbreviated as SICD) in Hidayat, *et al.* [22]. The difference between those two lies in the number of item type of parts, while

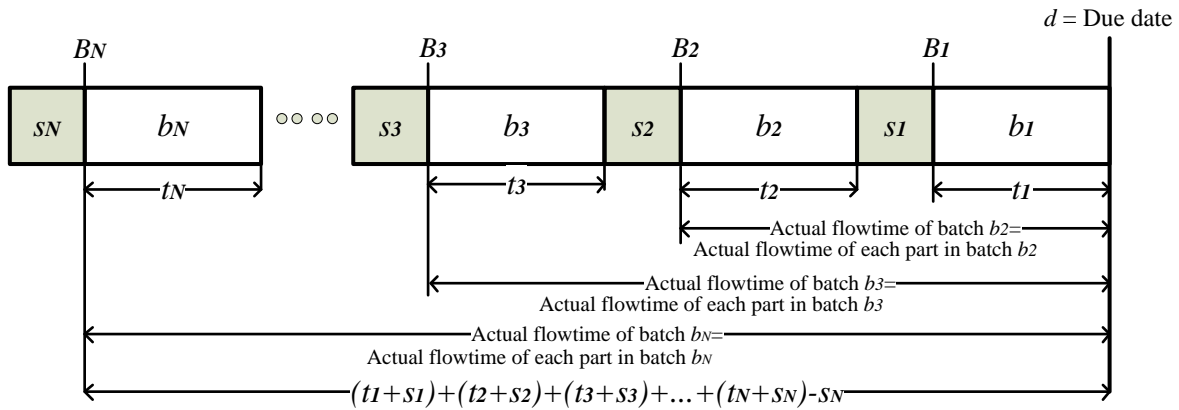


Figure 1. Illustration of the actual flowtime

simultaneously on a common due date. Under the condition of multiple items, the processed part consist of several different items and each item has different processing time and set up time.

It is not allowed to place any parts in any batch if the parts from other item are already in the batch. The processing time and setup time of each batch depend on the item type of parts in the batch. Therefore, we need a binary variable to indicate the item type of parts in a batch i.e. Z_{ig} . Where if batch b_i consist of part item g then $Z_{ig} = 1$, otherwise $Z_{ig} = 0$. It establishes several conditions that must be met, as follows.

- (1) The processing time of batch b_i is $t_i = \sum_{g=1}^k Z_{ig} \cdot t_g \quad \forall i = 1, 2, \dots, N$
- (2) The setup time of batch b_i is $s_i = \sum_{g=1}^k Z_{ig} \cdot s_g \quad \forall i = 1, 2, \dots, N$
- (3) The number of batches which consist of part item g is $N_g = \sum_{i=1}^N Z_{ig} \quad \forall g = 1, 2, \dots, k$
- (4) The number of demanded parts which come from item g is $n_g = \sum_{i=1}^N Q_i \cdot Z_{ig} \quad \forall g = 1, 2, \dots, k$

The entire batch on the shop must be processed within $[0, d]$, which is the available time for processing the parts. It can be seen from Figure 1 that the required time for the processing of N batches equals to $\sum_{i=1}^N (t_i + s_i) - s_N$. Definitely we need $\sum_{i=1}^N (t_i + s_i) - s_N \leq d$ as a constraint. Processing of the last batch should be completed exactly at the common due date. The last processed batch is the first batch on the schedule, so that we need $B_1 + t_1 = d$ as a constraint. In equation (6), it can be seen that F^a depends on variable decision Q_i , parameters t_i and s_i , the item type of part does not determine F^a . Therefore F^a for MICD problem is the same as F^a for SICD problem although the characteristic of the parts is developed from a single item into multiple items.

Problem MICD is formulated as model M1.

Model M1:

$$\text{minimize } F^a = \sum_{i=1}^N \{ \sum_{j=1}^i (t_j + s_j) - s_i \} Q_i \quad (6)$$

Subject to:

$$t_i = \sum_{g=1}^k Z_{ig} t_g \quad \forall i = 1, 2, \dots, N \quad (7)$$

$$s_i = \sum_{g=1}^k Z_{ig} s_g \quad \forall i = 1, 2, \dots, N \quad (8)$$

$$N_g = \sum_{i=1}^N Z_{ig} \quad \forall g = 1, 2, \dots, k \quad (9)$$

$$n_g = \sum_{i=1}^N Q_i \cdot Z_{ig} \quad \forall g = 1, 2, \dots, k \quad (10)$$

$$\sum_{i=1}^N (t_i + s_i) - s_N \leq d \quad (11)$$

$$B_1 + t_1 = d \quad (12)$$

$$0 < Q_i \leq c \quad \forall i = 1, 2, \dots, N \quad (13)$$

$$N > k \quad (14)$$

$$Z_{ig} = \begin{cases} 1 & \forall g = 1, 2, \dots, k; \forall i = \\ 0 & 1, 2, \dots, N \end{cases} \quad (15)$$

Constraint (7) shows that the processing time of batch b_i depends on the item type of parts in batch b_i . It is the consequence of only parts which have the same item type is grouped into the same batch. It also applies to the setup time of batch b_i in constraint (8). Constraint (9) ensures that the number of batches which consist of item g equals to the sum of the binary variable for each type of item. Constraint (10) ensures a balance of materials on the shop. It is necessary to ensure that the number of demanded parts of item g equals the total of the size of batches which consist of item g . Constraint (11) shows that the entire batches must be processed within an interval $[0, d]$. Constraint (12) states that the completion time of the first batch must be equal to the common due date. Constraint (13) is the upper limit and lower limit for the size of each batch. Constraint (14) states that the number of the batch on the shop is higher than the number of the item, this indicates that each item can form more than one batch. Constraint (14) is the value of the binary variable. It is apparent that if index g is eliminated, model M1 becomes SICD model in Hidayat, et al. [22]

The second model is developed for MIMD problem, which is the development of a single item with multi due dates problem (abbreviated as SIMD) in Hidayat,

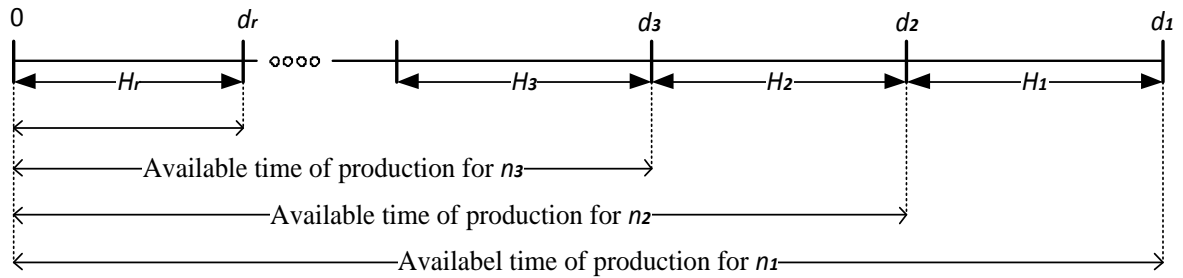


Figure 2. Available time of production

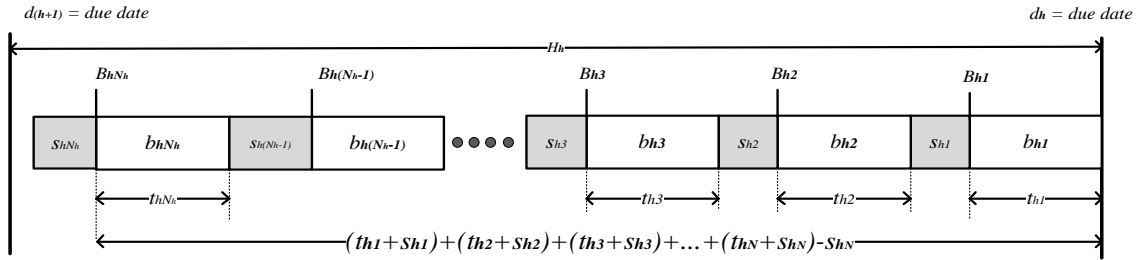


Figure 3. N_h batches within interval H_h

et al. [24]. The difference between those two lies in the number of item type of parts, while the similarity lies in the parts that should be delivered several times at different due dates. Under condition of multi due dates, the scheduling period is divided into r partitions of interval H_h ($h=1,2,\dots,r$) which is the interval between two consecutive due dates. Under the condition of multiple items of parts, there are n_h demanded parts in interval H_h which consist of n_{h1} parts of item 1, n_{h2} parts of item 2 and n_{hk} parts of item k . Therefore a binary variable i.e. Z_{hig} is needed. This variable indicates the item type of parts in a batch b_{hi} , where if batch b_{hi} consists of item g then $Z_{hig} = 1$, otherwise $Z_{hig} = 0$. It establishes several conditions that must be met, as follow.

- (1) The processing time of batch b_{hi} is $t_{hi} = \sum_{g=1}^k Z_{hig} \cdot t_g \quad \forall h = 1, 2, \dots, r \quad \forall i = 1, 2, \dots, N_h$
- (2) The set up time of batch b_{hi} is $s_{hi} = \sum_{g=1}^k Z_{hig} \cdot s_g \quad \forall h = 1, 2, \dots, r \quad \forall i = 1, 2, \dots, N_h$
- (3) The number of batches which consist of part item g is $N_{hg} = \sum_{i=1}^{N_h} Z_{hig} \quad \forall h = 1, 2, \dots, r \quad \forall g = 1, 2, \dots, k$

The number of demanded parts item g is $n_{hg} = \sum_{i=1}^{N_h} Q_{hi} \cdot Z_{hig}$ and $n_h = \sum_{g=1}^k n_{hg} \quad \forall h = 1, 2, \dots, r \quad \forall g = 1, 2, \dots, k$

The ideal condition is when the entire n_h parts can be processed within intervals H_h , but if it can not be met then the part can be processed in other intervals before d_h . From Figure 2 it can be seen that if n_1 could not entirely be satisfied in H_1 then the shortage of n_1 will be fulfilled in other intervals H_2, H_3, \dots, H_r . Each interval H_h can process its demanded parts and

the shortage of other interval's demanded parts, unless interval H_1 , therefore, the condition of $D_h \geq n_h$ can happen in interval H_2, H_3, \dots, H_r .

Definitely, we need constraints which are formulated as $\sum_{h=a+1}^r (D_h - n_h) \geq 0 \quad \forall a = 1, 2, \dots, (r - 1)$ and $\sum_{h=1}^r (D_h - n_h) = 0$. Figure 3 shows that the required time for processing of N_h batches within interval H_h equals to $(\sum_{i=1}^{N_h} t_{hi} + \sum_{i=1}^{(N_h-1)} s_{hi})$. This means we need $\sum_{i=1}^{N_h} t_{hi} + \sum_{i=1}^{N_h-1} s_{hi} \leq H_h$ as a constraint.

Hidayat, *et al.* [19] have developed F^a model for SIMD problem, as follows.

$$F^a = [\sum_{h=1}^r [\sum_{i=1}^{N_h} \{ \sum_{j=1}^i (s_{hj} + t_{hj}) - s_{hi} \} Q_{hi}]] + [\sum_{a=1}^{r-1} \{ \sum_{h=a+1}^r (D_h - n_h) \} (d_a - d_{a+1})] \quad (16)$$

It can be seen that the item type of part does not determine F^a . Therefore F^a for MIMD problem is the same as F^a for SIMD problem although characteristic of the parts is developed from single item into multiple items.

Problem MIMD is formulated as model M2.

Model M2:

$$\text{Minimize } F^a = [\sum_{h=1}^r [\sum_{i=1}^{N_h} \{ \sum_{j=1}^i (s_{hj} + t_{hj}) - s_{hi} \} Q_{hi}]] + [\sum_{a=1}^{r-1} \{ \sum_{h=a+1}^r (D_h - n_h) \} (d_a - d_{a+1})] \quad (17)$$

Subject to:

$$\sum_{i=1}^{N_h} t_{hi} + \sum_{i=1}^{(N_h-1)} s_{hi} \leq H_h \quad \forall h = 1, 2, \dots, r \quad (18)$$

$$s_{hi} = \sum_{g=1}^k Z_{hig} \cdot s_g \quad \forall i = 1, 2, \dots, N_h \quad \forall h = 1, 2, \dots, r \quad (19)$$

$$t_{hi} = \sum_{g=1}^k Z_{hig} \cdot t_g \quad \forall i = 1, 2, \dots, N_h \quad \forall h = 1, 2, \dots, r \quad (20)$$

$$N_{hg} = \sum_{i=1}^{N_h} Z_{hig} \quad (21)$$

$$\forall g = 1, 2, \dots, k \quad \forall h = 1, 2, \dots, r$$

$$D_{hg} = \sum_{i=1}^{N_h} Q_{hi} \cdot Z_{hig} \quad (22)$$

$$\forall g = 1, 2, \dots, k \quad \forall h = 1, 2, \dots, r$$

$$\sum_{h=a+1}^r (D_h - n_h) \geq 0 \quad \forall a = 1, 2, \dots, (r - 1) \quad (23)$$

$$\sum_{h=1}^r (D_h - n_h) = 0 \quad (24)$$

$$\sum_{i=1}^{N_h} Q_{hi} = D_h \quad \forall h = 1, 2, \dots, r \quad (25)$$

$$B_{h1} + t_{h1} = d_h \quad \forall h = 1, 2, \dots, r \quad (26)$$

$$0 \leq Q_{hi} \leq c, \forall i = 1, \dots, N_h; \quad \forall h = 1, 2, \dots, r \quad (27)$$

$$Z_{hig} = \begin{cases} 1 \\ 0 \end{cases} \quad (28)$$

$$\forall h = 1, 2, \dots, r; \forall i = 1, 2, \dots, N; \forall g = 1, 2, \dots, k$$

$$D_{hg} > 0 \quad (29)$$

$$\forall h = 1, 2, \dots, r; \forall g = 1, 2, \dots, k$$

Constraint (18) shows that in each interval, the entire of N_h formed batches must be processed within interval H_h . Constraint (19) shows that the setup time of batch b_{hi} depends on the item type of parts in batch b_{hi} . It also applies to the processing time of batch b_{hi} in constraint (20). Constraint (21) to ensure that the sum of the binary variable for each type of item in each interval H_h equals to the number of batches which consist of item g in each interval H_h . Constraint (22) is necessary to ensure a balance of material in each interval. It shows that in each interval, the number of parts in all of formed batches which consists of item g , must be equal to the number of processed parts of item g . Constraint (23) is necessary due to conditions of multi due date. It shows that the number of processed parts within a time interval of H_r up to H_h must be greater or equals to the number of demanded parts within the time interval. Constraint (24) is necessary to ensure a balance of materials on the shop. It shows that in the scheduling period, the processed parts must be equal to the demanded parts. Constraint (25) is necessary to ensure a balance of material in each interval. It shows that in each interval the number of processed parts must be equal to the number of parts in all of the formed batches. Constraint (26) states that the completion time of the first batch in each interval must be equals the due date of that interval (d_h). Constraint (27) states the upper and lower bounds for the size of each batch. Constraint (28) is the value of the binary variable. Constraint (29) states the number of processed parts of item g in each interval H_h must be positive.

It is apparent that if the index g is eliminated, model M2 becomes SIMD model in Hidayat, *et al.* [24]. If the index h is eliminated model M2 becomes model M1. Subsequently, if the index h and g are eliminated, model M2 becomes SICD model in Hidayat, *et al.* [22].

Solution Procedure

F^a will be minimum if the batching and the scheduling phase provide a minimum solution. The size of

each batch could not exceed the capacity of the batch processor so that the highest of batch size equals to the batch processor capacity, and the lowest is one. At the maximum batch size, the number of formed batches will be minimum, which around up the result of the total demand divided by the capacity, (n_g/c) . Rounding up is needed because the number of the batch must be an integer while the results of (n_g/c) is not always an integer. The decimal point indicates the unaccommodated parts in a batch. Rounding up also indicates an idle batch, in which the size of that batch is lower than the processor capacity.

Theorem 1. For a shop with N batches and each of which consists of the same single item. Each batch has processing time t_i , setup time s_i and size of the batch Q_i ($i = 1, 2, \dots, N$). Using backward scheduling, the total actual flowtime of parts through the shop will be minimum if the batches are arranged so that

$$\frac{(t_1 + s_1)}{Q_1} \leq \frac{(t_2 + s_2)}{Q_2} \leq \frac{(t_3 + s_3)}{Q_3} \leq \dots \leq \frac{(t_N + s_N)}{Q_N}$$

Proof. Suppose there are two sequences of N batches. The difference between the two lies only in that the first sequence. It shows that batch b_m is in the m^{th} position and batch $b_{(m+1)}$ is in the $(m + 1)^{th}$ position. For the second sequence shows that the batch b_m in the $(m + 1)^{th}$ position and the batch $b_{(m+1)}$ is in the m^{th} position. The total actual flowtime of parts through the shop under the first and the second sequences respectively are F^{1a} and F^{2a} .

It is obtained that

$$F^{1a} - F^{2a} = (t_m + s_m)Q_{(m+1)} - (t_{(m+1)} + s_{(m+1)})Q_m$$

The first sequence will be less than or equal to the second sequence if and only if:

$$F^{1a} - F^{2a} \leq 0$$

$$(t_m + s_m)Q_{(m+1)} - (t_{(m+1)} + s_{(m+1)})Q_m \leq 0 \quad \text{or}$$

$$\frac{(t_m + s_m)}{Q_m} \leq \frac{(t_{(m+1)} + s_{(m+1)})}{Q_{(m+1)}}$$

The best sequence will be obtained if this procedure is continued until all the batches are in an increasing order of $(t_i + s_i)/Q_i$, for $i = 1, 2, \dots, N$

There are N generated batches, that will be scheduled on a batch processor. First, arrange the N batches in any order, so that a sequence of batch b_i for $i = 1, 2, \dots, N$ is obtained. Next, sort the sequence of batches increasingly in $(t_i + s_i)/Q_i$, for $i = 1, 2, \dots, N$ (see Theorem 1).

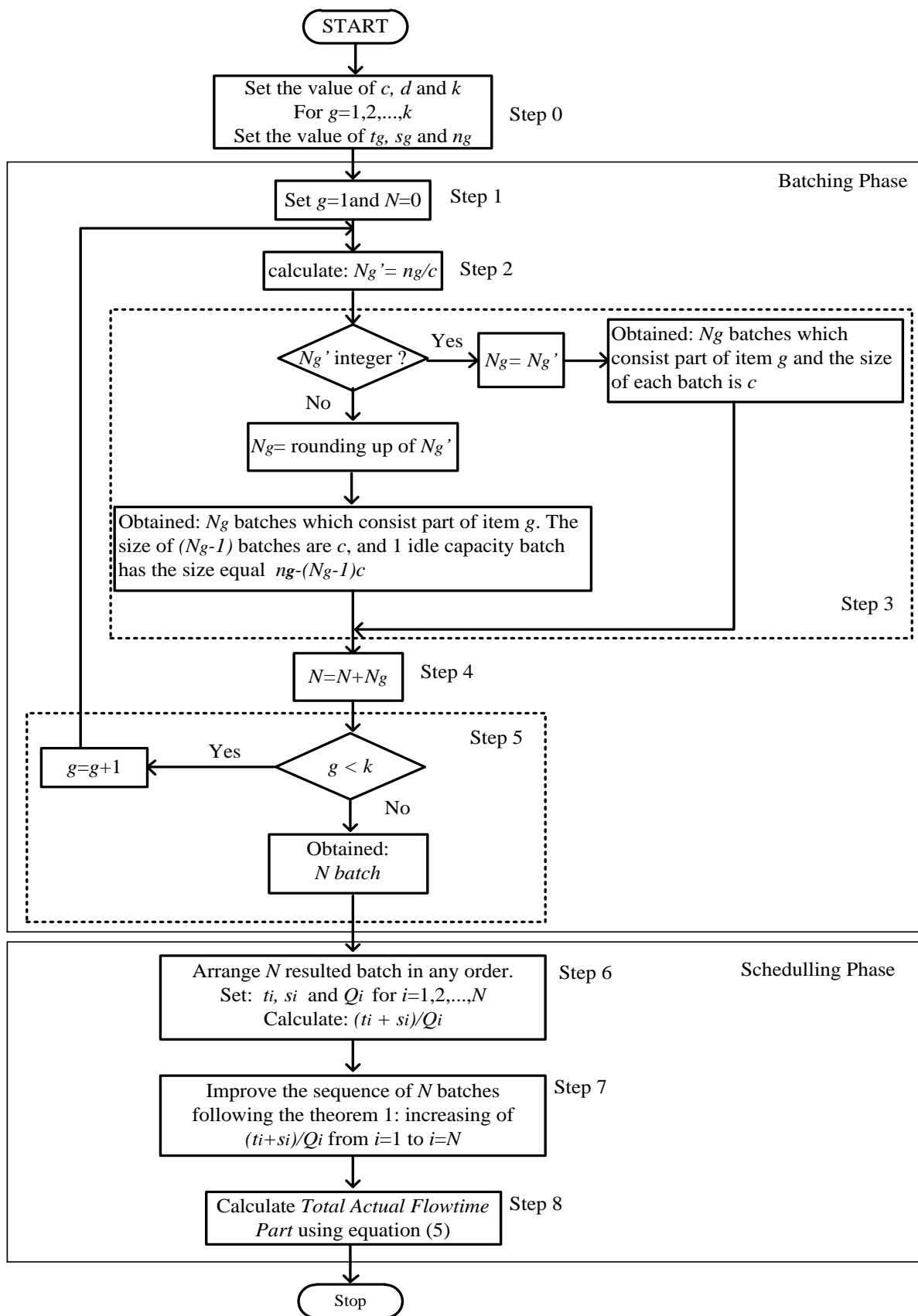
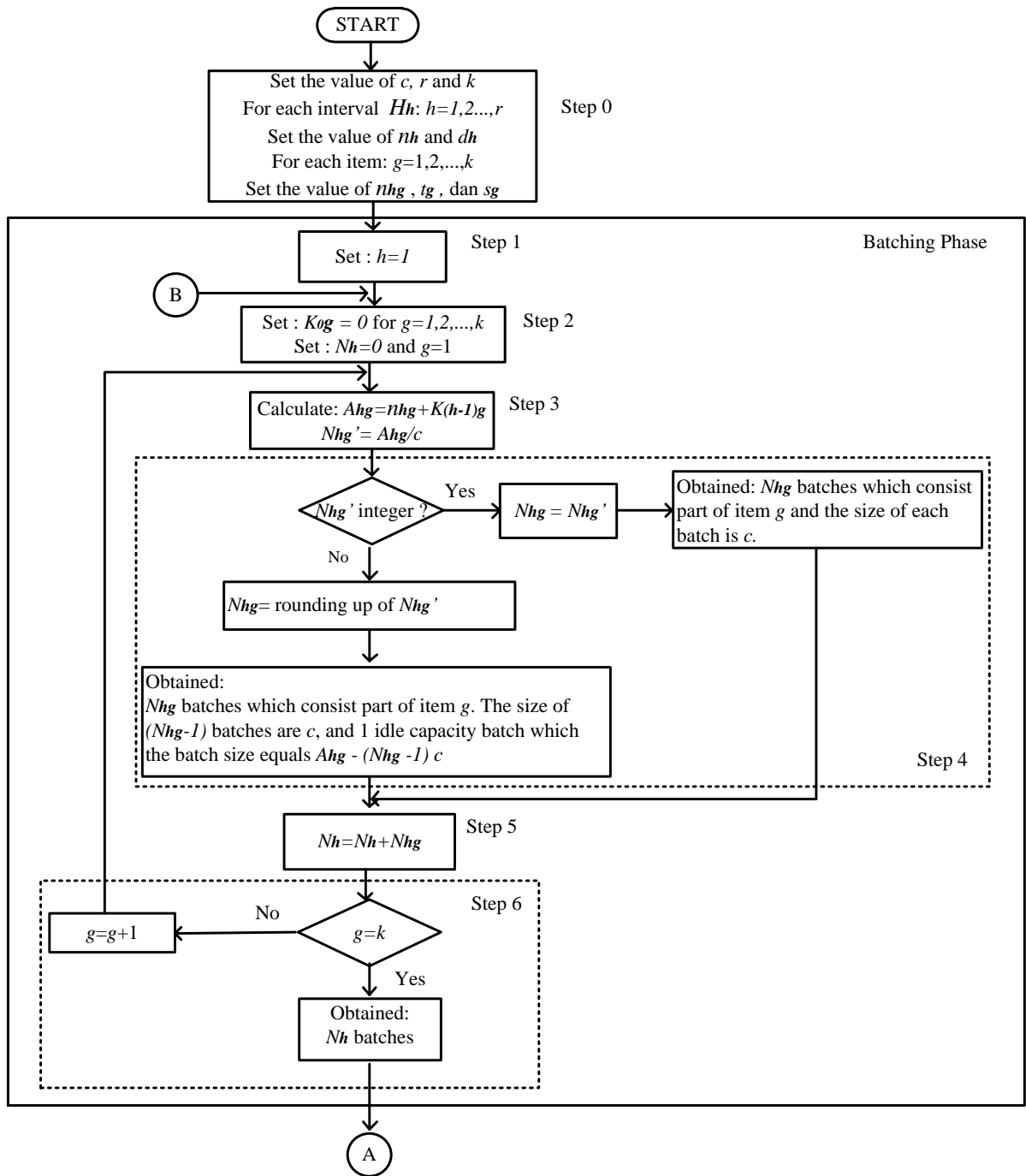


Figure 4. Flowchart for Algorithm A1



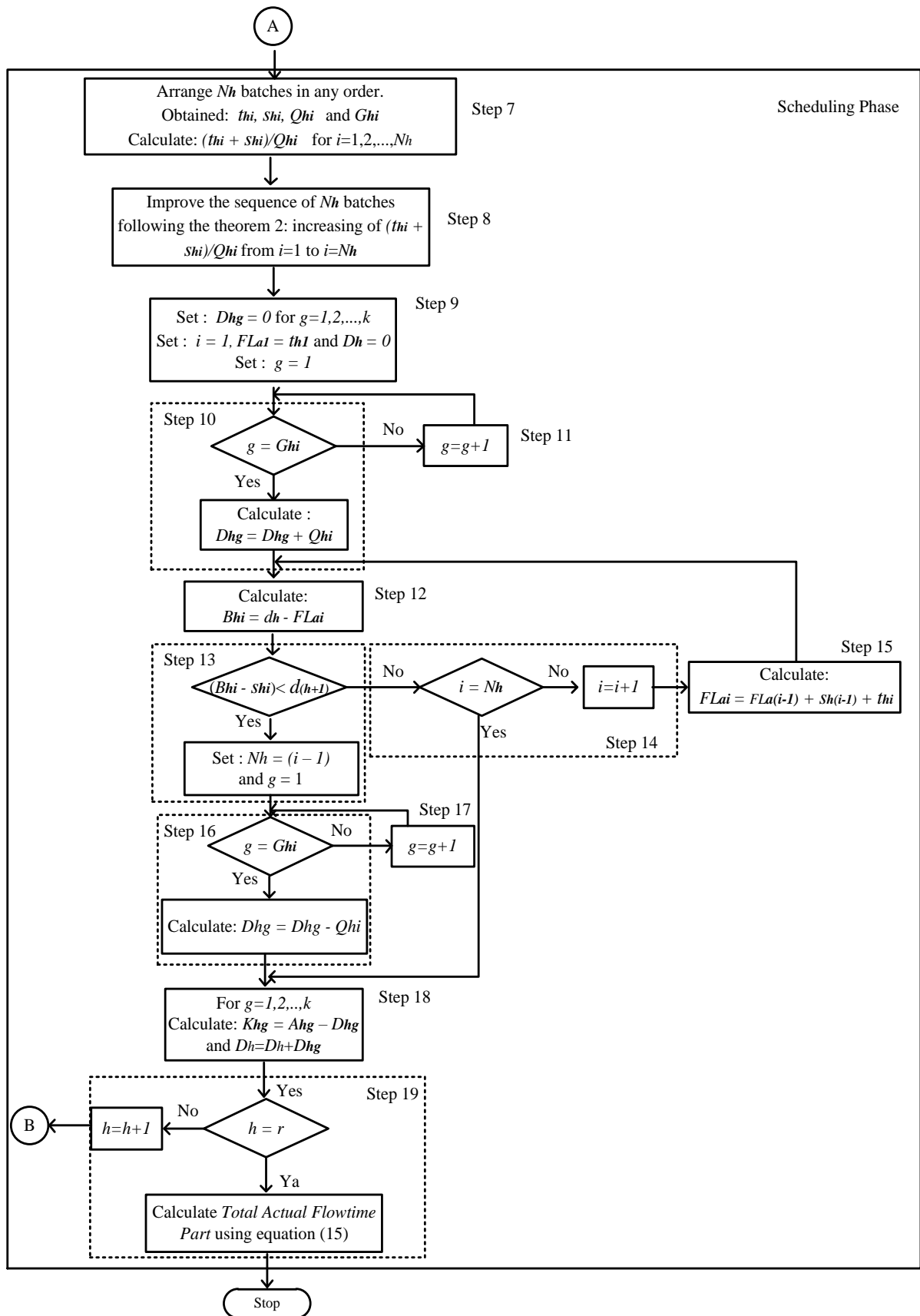


Figure 5. Flowchart for Algorithm A2

Algorithm A1 to solve problem MICD is developed, and it is presented as a flowchart in Figure 4.

Algorithm A1

- Step 0. Set the value of c , d , and k
Set the value of n_g , t_g , and s_g ,
for $g = 1, 2, \dots, k$.
- Step 1. Set $g = 1$ and $N=0$, proceed to Step 2
- Step 2. Calculate $N'_g = \frac{n_g}{c}$, proceed to Step 3
- Step 3. If N'_g integer then $N_g = N'_g$ and proceed to Step 4 (obtained N_g batches which consist part of item g and the size of each batch is c);
Otherwise $N_g =$ rounding up of N'_g and proceed to Step 4 (obtained N_g batches which consist part of item g .
The size of (N_g-1) batches is c , and one idle capacity batch which the batch size equals $n_g - (N_g - 1)c$).
- Step 4. Set $N = N + N_g$, proceed to Step 5
- Step 5. If $g < k$ then $g = g+1$ and proceed to Step 2; otherwise proceed to Step 6 (Obtained N batches)
- Step 6. Arrange N resulted batches in any order, set t_i, s_i and Q_i for $i = 1, 2, \dots, N$. Calculate $(t_i + s_i)/Q_i$ and proceed to Step 7
- Step 7. Improve the sequence of N batches with folowing the Theorem 1:
 $\frac{(t_1+s_1)}{Q_1} \leq \frac{(t_2+s_2)}{Q_2} \leq \frac{(t_3+s_3)}{Q_3} \leq \dots \leq \frac{(t_N+s_N)}{Q_N}$
and proceed to Step 8.
- Step 8. Calculate F^a using equation (6) and then stop.

From Figure 2, it can be seen that under the condition of multi due dates, the demanded parts (n_h) can be fulfilled in some intervals before its due date (d_h) unless n_r . Definitely in each interval H_h we need to distinguish between the demanded part (n_{hg}), the processed parts (D_{hg}), and the required part to process (A_{hg}) for $g = 1, 2, \dots, k$. The difference between the required parts to process and the processed parts is the shortage of parts in interval H_h , i.e., $K_{hg} = A_{hg} - D_{hg}$. Then for the next interval $H_{(h+1)}$ the required parts to process is equal to the sum of demanded parts and the shortage of parts in the previous interval, i.e., $A_{(h+1)g} = n_{(h+1)g} + K_{hg}$, and the resulting batches ($N_{(h+1)g}$) is equal to the round up of $A_{(h+1)g}/c$.

Theorem 2. For a shop with r intervals H_h ($h=1, 2, \dots, r$) and each interval has N_h batches, each bathces consists of the same single item. Each batch has processing time t_{hi} , set up time s_{hi} and size of batch Q_{hi} ($i=1, 2, \dots, N_h$). Using backward scheduling, the total actual flowtime of parts through the shop will be minimum if the batches in each interval H_h are arranged so that

$$\frac{(t_{h1}+s_{h1})}{Q_{h1}} \leq \frac{(t_{h2}+s_{h2})}{Q_{h2}} \leq \frac{(t_{h3}+s_{h3})}{Q_{h3}} \leq \dots \leq \frac{(t_{hN_h}+s_{hN_h})}{Q_{hN_h}}$$

Proof. Suppose in the interval H_h there are two sequences of N_h batches. The difference between the two lies only in that the first sequence shows that batch b_{hv} is in the v^{th} position and batch $b_{h(v+1)}$ is in the $(v + 1)^{th}$ position, while the second sequence shows that the batch b_{hv} in the $(v + 1)^{th}$ position and the batch $b_{h(v+1)}$ is in the v^{th} position. The total actual flowtime of parts through the interval under the first and the second sequences respectively are F_h^{1a} and F_h^{2a} . It is obtained that

$$F_h^{1a} - F_h^{2a} = (t_{hv} + s_{hv})Q_{h(v+1)} - (t_{h(v+1)} + s_{h(v+1)})Q_{hv}$$

The first sequence will be better than or equivalent to the second sequence ($F_h^{1a} \leq F_h^{2a}$) if and only if:

$$(t_{hv} + s_{hv})Q_{h(v+1)} \leq (t_{h(v+1)} + s_{h(v+1)})Q_{hv}, \text{ or}$$

$$\frac{(t_{hv}+s_{hv})}{Q_{hv}} \leq \frac{(t_{h(v+1)}+s_{h(v+1)})}{Q_{h(v+1)}}$$

The best sequence will be obtained if this procedure is continued until all the batches are in increasing order of $(t_{hi} + s_{hi})/Q_{hi}$, for $i = 1, 2, \dots, N_h$.

$$\frac{(t_{h1}+s_{h1})}{Q_{h1}} \leq \frac{(t_{h2}+s_{h2})}{Q_{h2}} \leq \frac{(t_{h3}+s_{h3})}{Q_{h3}} \leq \dots \leq \frac{(t_{hN_h}+s_{hN_h})}{Q_{hN_h}}$$

Algorithm A2 to solve problem MIMD is developed, and it is presented as a flowchart in Figure 5.

Algorithm A2

- Step 0. Set the value of c, r dan k
Set the value of d_h for each interval H_h ,
 $h=1, 2, \dots, r$
Set the value of n_{hg}, t_g and s_g for $g=1, 2, \dots, k$
- Step 1. Set $h = 1$ then proceed to Step 2
- Step 2. Set $K_{0g} = 0$ for $g=1, 2, \dots, k$;
Set $N_h = 0$ and $g = 1$ then proceed to Step 3
- Step 3. Calculate $A_{hg} = n_{hg} + K_{(h-1)g}$ and $N'_{hg} = \frac{A_{hg}}{c}$
then proceed to Step 4
- Step 4. If N'_{hg} integer then $N_{hg} = N'_{hg}$, and proceed to Step 5 (obtained N_{hg} batches and the size of each batch are c);
otherwise $N_{hg} =$ rounding up of N'_{hg} then proceed to Step 5 (obtained N_{hg} batches which consist part of item g .
The size of $(N_{hg}-1)$ batches are c , and 1 idle capacity batch which the batch size equals $A_{hg} - (N_{hg} - 1)c$)
- Step 5. Calculate $N_h = N_h + N_{hg}$ then proceed to Step 6
- Step 6. If $g = k$ then proceed to Step 7 (obtained N_h batches);
otherwise $g = g + 1$ then proceed to Step 3

- Step 7. Arrange N_h batches in any order (obtained t_{hi} , s_{hi} , Q_{hi} and G_{hi}); Calculate: $\frac{(t_{hi}+s_{hi})}{Q_{hi}}$ for $i=1,2,\dots,N_h$ then proceed to Step 8
- Step 8. Improve the sequence of N_h batches with following the Theorem 2, then proceed to Step 9
- Step 9. Set $D_{hg} = 0$ for $g = 1,2,\dots, k$; Set $i=1, FL_{h1}^a = t_{h1}$, and $D_h = 0$ then proceed to Step 10
- Step 10. Set $g=1$, then proceed to Step 11
- Step 11. If $g = G_{hi}$ then calculate $D_{hg} = D_{hg} + Q_{hi}$, and proceed to Step 13; otherwise proceed to Step 12
- Step 12. $g = g + 1$, then proceed to Step 11
- Step 13. Calculate $B_{hi} = d_h - FL_{hi}^a$ and proceed to Step 14
- Step 14. If $(B_{hi} - s_{hi}) < d_{(h+1)}$ then $N_h = i - 1$ and proceed to Step 17; otherwise proceed to Step 15
- Step 15. If $i = N_h$ then proceed to Step 18; otherwise $i = i + 1$, and proceed to Step 16
- Step 16. Calculate $FL_{hi}^a = FL_{h(i-1)}^a + s_{h(i-1)} + t_{hi}$, and proceed to Step 10
- Step 17. Calculate $D_{hg} = D_{hg} - Q_{hi}$, and proceed to Step 18
- Step 18. Calculate $K_{hg} = A_{hg} - D_{hg}$ and $D_h = D_h - D_{hg}$ for $g = 1,2,\dots, k$, proceed to Step 19
- Step 19. If $h \neq r$ then $h = h + 1$, proceed to Step 3; otherwise calculate F^a using equation (17), then Stop

Results and Discussions

In this section we give numerical examples for MICD and MIMD consecutively.

For the MICD case, we have the numbers of multiple items parts demanded as many as 30, 20 and 25; and the due date is 1.000 unit time. Additionally, the capacity of batch processor is 20 parts, while the processing time and the setup time of each item are presented in Table 1.

The calculation is done using Algorithm A1. Table 2 shows the result of MICD numerical example and the Gantt chart is presented in Figure 6.

For the MIMD case; we have the numbers of multiple items parts demanded are 270, 260, 275, 310, 290 and 260; and the due dates are 10.000, 9.750, 9500, 9.250, 8.950 and 8.700 unit time respectively. The part processing time and the setup time of each item can be seen in Table 3. The calculation is done using Algorithm A2. The results are presented as gantt chart (see Figure 7) and tables (see Table 4 and Table 5).

Analysis

The proposed algorithm shows that we should maximize the utilization of the batch processor. We

Table 1. Data for MICD numerical example

g	1	2	3
n_g	30	20	25
t_g	20	10	30
s_g	7	5	9

Table 2. The result of MICD numerical example.

i	1	2	3	4	5
G_i	2	1	3	1	3
t_i	10	20	30	20	30
s_i	5	7	9	7	9
Q_i	20	20	20	10	5
$\left(\frac{t_i + s_i}{Q_i}\right)$	0.75	1.35	1.95	2.7	7.8
FL^a	10	35	72	101	138
F^a	4,040				

should also schedule backwardly the resulting batches in a non-increasing batch size.

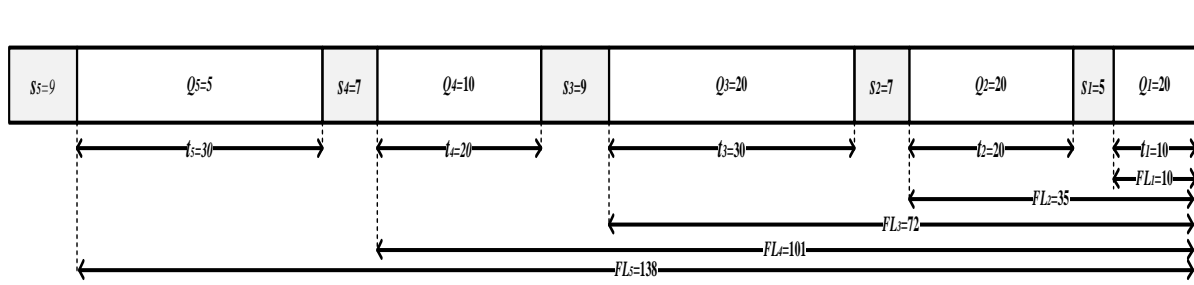
The batch size is always equal to the batch processor capacity unless the value of (n_g/c) and (A_{hg}/c) is rounded up. This step is required in order to get the minimal number of generated bathces. Moreover, that value shows the presence of an idle capacity batch. The maximum number of the idle batch for each item type is 1, and the maximum number of the idle batch on the shop equals to the number of item type.

The batch on the first position is always a full capacity one which has the lowest $(t_g + s_g)$. Under the common due date condition, the first position on the shop is the ideal position, in which the batch completion time exactly coincide with the due date. In this condition, that batch will be delivered directly after the process is completed, so that the waiting time of all parts equals to zero. Then, for the batch on the second until the last position will have waiting time before they are delivered. As the result of the placement of a full capacity batch of item which has the lowest of $(t_g + s_g)$, the total waiting time as well as F^a is reduced.

Conclusions

There are three steps that should be done to minimize the total actual flowtime. First, rounding up the total demand divided by the capacity of batch processor to find a minimum number of the batch. Second, the size of each batch equals to the batch processor capacity, unless there is an idle batch. Third, the batch on the first position (i.e closest to the due date) is a full capacity batch of item which has the lowest of the sum of processing and set up time.

$d=1.000$



6. Gantt Chart for MICD Numerical Example

Figure

Table 3. Data for MIMD numerical example

h	1			2			3			4			5			6		
d_h	10.000			9.750			9.500			9.250			8950			8700		
n_h	270			260			275			310			290			260		
g	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
n_{hg}	90	80	100	75	85	100	90	100	85	100	110	100	110	100	80	100	90	70
t_{hg}	50	30	40	50	30	40	50	30	40	50	30	40	50	30	40	50	30	40
s_{hg}	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

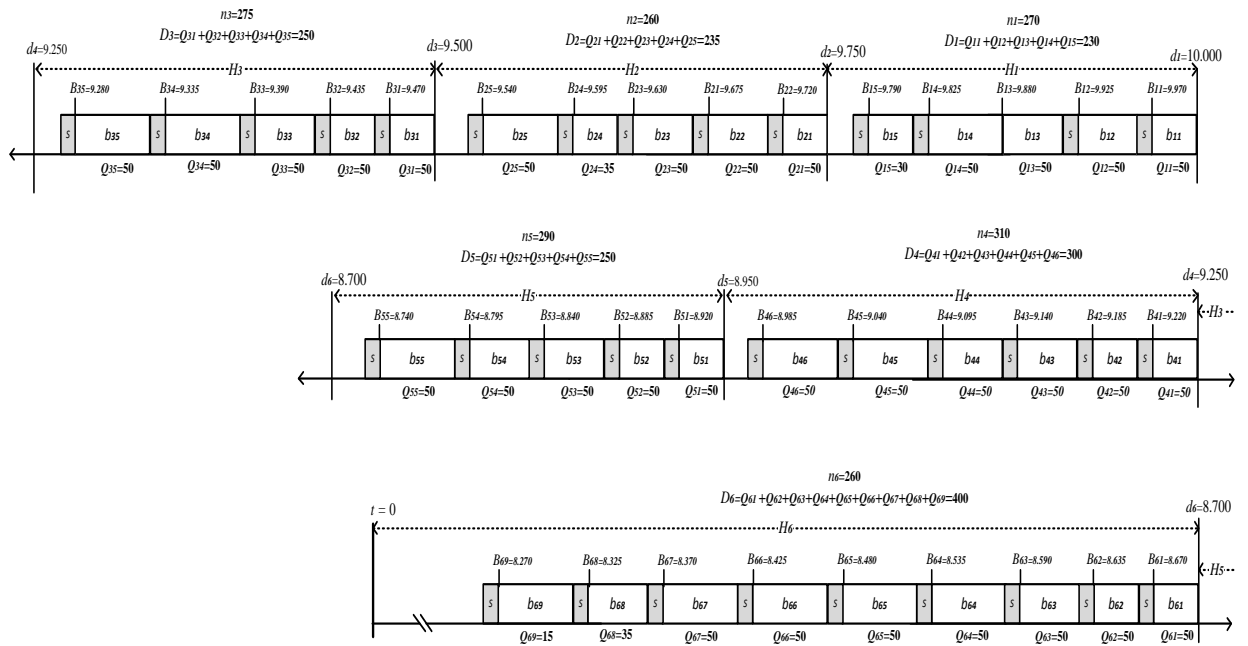


Figure 7. Gantt chart for MIMD numerical example

Table 4. The result of MIMD numerical example

h	i	G_{hi}	t_{hi}	s_{hi}	Q_{hi}	$\frac{t_{hi}+s_{hi}}{Q_{hi}}$	FL_{hi}^a	F_{hi}^a	F_h^a	F^a
1	1	2	30	5	50	0,7	30	1.500	26.300	
	2	3	40	5	50	0,9	75	3.750		
	3	3	40	5	50	0,9	120	6.000		
	4	1	50	5	50	1,1	175	8.750		
	5	2	30	5	30	1,17	210	6.300		
2	1	2	30	5	50	0,7	30	1.500	27.175	
	2	3	40	5	50	0,9	75	3.750		
	3	3	40	5	50	0,9	120	6.000		
	4	2	30	5	35	1	155	5.425		
	5	1	50	5	50	1,1	210	10.500		
3	1	2	30	5	50	0,7	30	1.500	29.500	
	2	2	30	5	50	0,7	65	3.250		
	3	3	40	5	50	0,9	110	5.500		
	4	1	50	5	50	1,1	165	8.250		
	5	1	50	5	50	1,1	220	11.000		
4	1	2	30	5	50	0,7	30	1.500	41.750	232.550
	2	2	30	5	50	0,7	65	3.250		
	3	3	40	5	50	0,9	110	5.550		
	4	3	40	5	50	0,9	155	7.750		
	5	1	50	5	50	1,1	210	10.500		
	6	1	50	5	50	1,1	265	13.250		
5	1	2	30	5	50	0,7	30	1.500	28.500	
	2	2	30	5	50	0,7	65	3.250		
	3	3	40	5	50	0,9	110	5.500		
	4	3	40	5	50	0,9	155	7.750		
	5	1	50	5	50	1,1	210	10.500		
6	1	2	30	5	50	0,7	30	1.500	79.325	
	2	2	30	5	50	0,7	65	3.250		
	3	3	40	5	50	0,9	110	5.500		
	4	1	50	5	50	1,1	165	8.250		
	5	1	50	5	50	1,1	220	11.000		
	6	1	50	5	50	1,1	275	13.750		
	7	1	50	5	50	1,1	330	16.500		
	8	3	40	5	35	1,29	375	13.125		
	9	1	50	5	15	3,67	430	6.450		

Table 5. The result of calculation for each interval

h	g	D_{hg}	D_h	$D_h - n_h$	n_h	n_{hg}	g
1	1	50	230	-40	270	90	1
	2	80				80	2
	3	100				100	3
2	1	50	235	-25	260	75	1
	2	85				85	2
	3	100				100	3
3	1	100	250	-25	275	90	1
	2	100				100	2
	3	50				85	3
4	1	100	300	-10	310	100	1
	2	100				110	2
	3	100				100	3
5	1	50	250	-40	290	110	1
	2	100				100	2
	3	100				80	3
6	1	215	400	140	260	100	1
	2	100				90	2
	3	85				70	3

The multi-stage scheduling problem is a complex problem. In the early stage, we have to understand the scheduling as a single stage. Therefore, discussion on the batch processor scheduling as a single stage is a must.

In this paper, we presented a simple system, i.e., a single stage with multi-item parts and multi due date. This study, is a step in a series of bigger research for solving the multi-stage scheduling which consist of job processor-batch processor – job processor. We also presented the scheduling for job processor using backward scheduling approach with flowtime as the performance criteria.

References

- Cheng, B., Wang, Q., Yang, S., and Hu, Xiaoxuan, An Improved Ant Colony Optimization for Scheduling Identical Parallel Batching Machines with Arbitrary Job Sizes, *Applied Soft Computing* 13, 2013, pp.765–772.
- Malapert, A., Gueret, C., and Rousseau, L. M., A Constraint Programming Approach for a Batch Processing Problem with Non Identical Job Sizes, *European Journal of Operational Research*, 221, 2012, pp. 533–545.
- Gokhalea, R. and Mathirajan, M., Heuristic Algorithms for Scheduling of a Batch Processor in Automobile Gear Manufacturing, *International Journal of Production Research*, 49(10), 2011, pp. 2705–2728.
- Gaudreault, J., Frayret, J.M., Rousseau, A., and D'Amours, S., Combined Planning and Scheduling in a Divergent Production System with Co-production: A Case Study in the Lumber Industry, *Computers & Operations Research*, 38, 2011, pp. 1238–1250.
- Gong, Hua., Tang, Lixin., and Duin, C.W., A Two-Stage Flowshop Scheduling Problem on a Batching Machine and a Discrete Machine with Blocking and Shared Setup Times, *Computers & Operations Research*, 37, 2010, pp. 960 – 969.
- Chakhlevitch, K., Glass, C.A., and Kellerer, H., Batch Machine Production with Perishability Time Windows and Limited Batch Size, *European Journal of Operational Research*, 210, 2011, pp. 39–47.
- Damodaran, P., Srihari, K., and Lam, S., Scheduling a Capacitated Batch Processing Machine to Minimize Makespan, *Robotics and Computer-Integrated Manufacturing* 23, 2007, pp. 208–216.
- Guo, Chengtao., Jiang, Zhibin., Zhang, Huai., and Li, Na., Decomposition-based Classified Ant Colony Optimization Algorithm for Scheduling Semiconductor Wafer Fabrication System, *Computers & Industrial Engineering*, 62, 2012, pp. 141–151.
- Dauzère-Pères, S. and Mönch, L., Scheduling Jobs on a Single Batch Processing Machine with Incompatible Job Families and Weighted Number of Tardy Jobs Objective, *Computers & Operations Research*, 40, 2013, pp. 1224–1233.
- Condotta, A., Knust, S., and Shakhlevich, N.V., Parallel Batch Scheduling of Equal-Length Jobs with Release and Due Dates, *Journal of Scheduling*, 13, 2010, pp. 463–477.
- Noroozi, A., Mokhtari, H., and Abadi, I. N. K., Research on Computational Intelligence Algorithms with Adaptive Learning Approach For Scheduling Problems with Batch Processing Machines, *Neurocomputing*, 101, 2013, pp. 190–203.
- Behnamiana, J., Ghomia, F.S.M.T., Jolai, F., and Amirtaheria, O., Minimizing Makespan on a Three Machine Flowshop Batch Scheduling Problem with Transportation Using Genetic Algorithm, *Applied Soft Computing*, 12, 2012, pp. 768–777.
- Kashan, A.H., Karimi, B., and Jolai, F., An Effective Hybrid Multi Objective Genetic Algorithm for Bi-criteria Scheduling on a Single Batch Processing Machine with Non-Identical Job Sizes, *Engineering Applications of Artificial Intelligence*, 23, 2010, pp. 911–922.
- Parsa, N.R., Karimi, B., and Kashan, A.H., A Branch and Price Algorithm to Minimize Makespan on a Single Batch Processing Machine with Non-Identical Job Sizes, *Computers & Operations Research*, 37, 2010, pp. 1720–1730.
- Bellanger, A., Oulamara, A., and Kovalyov, Minimizing Total Completion Time on a Batching Machine with Job Processing Time Compatibilities, *Electronic Notes in Discrete Mathematics* 36, 2011, pp. 1295–1302.
- Halim A.H., and Ohta, H., Batch-scheduling through the Flowshop with Both Receiving and Delivery Just in Time, *International of Production Research*, 31(8), 1993, pp. 1943-1955.
- Halim A.H., Miyazaki, S. and Ohta, H., Batch-Scheduling Problems to Minimize Actual Flowtime of Parts through the Shop under JIT Environment, *European Journal of Operation Research*, 72, 1994, pp. 529-544.
- Halim, A.H., Miyazaki, S. and Ohta, H., Lot Scheduling Problems of Multiple Items in the Shop with Both Receiving and Delivery Just in Time, *Production and Planning Control*, 5, 1994, pp. 175 – 184.
- Halim, A.H, Suryadhini, P.P. and Toha, I.S., Batch Scheduling to Minimize Total Actual Flowtime in a Two-Stage Flowshop with Dedicated Machines in the Second Stage, *Proceeding*

- of the 7th Asia Pacific Industrial Engineering and Management Systems Conference*, Bangkok, Thailand, 2006, pp 1043-1349.
20. Sukoyo, Samadhi, A.T.M.A, Iskandar, B..P, and Halim, A.H., Flow Shop Batch Scheduling for Single Item with Increasing Processing Time to Minimize Total Actual Flowtime, *2nd Asia Pacific Conference on Manufacturing Systems*, 2010, Yogyakarta, Indonesia.
 21. Zahedi, Samadhi, A.T.M.A, Suprayogi, and Halim, A.H., Integrated Batch Production and Maintenance Scheduling for Multiple Items Processed on a Deteriorating Machine to Minimize Total Production and Maintenance Cost with Due-date Constraint, *International Journal of Industrial Engineering Computation*, 7, 2016, pp.229-244.
 22. Hidayat, N.P.A., Cakravastia A., Samadhi, A. TMA., Halim, A.H., A Single Item Batch Scheduling Model on a Batch Processor to Minimize Total Actual Flowtime of Parts Through the Shop, *Proceeding Asia Pacific Industrial Engineering and Management System*, Cebu Island, Philipina, 2013.
 23. Hidayat, N. P.A., Cakravastia A., Samadhi, A. TMA., and Halim, A.H., A Batch-scheduling Problem to Minimize Total Actual Flowtime of Parts through the Shop which has m Heterogenous Batch Processors, *Proceeding Asia Pacific Industrial Engineering and Management System*, Jeju Island, South Korea, 2014.
 24. Hidayat, N. P.A., Cakravastia A., Samadhi, A. TMA., and Halim, A.H., A Batch Scheduling Model for m Heterogeneous Batch Processor, *International of Production Research*, 54(4), 2016, pp1170-1185.
 25. Halim, A.H., Ernawati, and Hidayat, N. P.A., A Model of Batch Scheduling for a Single Batch Processor with Addintional Setups to Minimize Total Inventory Holding Cost of Parts of a Single Item Requested at Multi-due-date, *Proceeding Asia Pacific Conference on Manufacturing Systems and International Manufacturing Engineering Conference*, Yogyakarta, Indonesia, 2017.