Batch scheduling for a single machine processing parts of a single item with increasing processing time to minimize total actual flow time

Sukoyo¹, TMA Ari Samadhi, Bermawi P. Iskandar, and Abdul Hakim Halim²†
Bandung Institute of Technology, Bandung 40132, INDONESIA
Email: sukoyo@mail.ti.itb.ac.id¹
ahakimhalim@lspitb.org²

Abstract. This research addresses a batch scheduling model for a single machine processing parts of a single item to minimize total actual flow time, defined as interval time from the arrival time to the due date. The part processing time is assumed to be increasing linearly with increasing the waiting time of the part from the batch completion time to its due date. The model consists of the following decision variables: the number of batches (N), batch sizes, starting times of processing, and sequence of the resulting batches. The problem is formulated as a non-linear programming model for which a relaxation is applied by considering variable N to be a parameter. The next step is to solve the model for several values of N, started from N = 1 and increased one by one until a stopping rule is satisfied. This paper is provided with numerical examples to show the characteristics of the model and how effective the proposed algorithm solves the problem.

Keywords: batch scheduling, single machine, actual flow time, increasing processing time.

1. INTRODUCTION

Research on batch scheduling generally assumes that processing time is a constant value and in practice, this assumption is not always true. If a batch has been finished before its due date, additional processing time from its completion time until its due date is needed for maintaining the qualities of parts in the completed batch. This means that processing time could also be dependent upon time interval from its batch completion time until its due date. Longer the interval between its completion time and its due date requires longer time for the additional process. This condition leads to the fact that processing time is not a parameter but a variable decision.

As the capacity of a manufacturing system is limited, completing a batch of a single item processed on a single machine before its common due date due could not be avoided. To deliver all completed parts exactly on their common due date in the just in time environment, the batch processed last should be sequenced in order to be completed exactly at their common due date. If the batch sequence number is counted backwardly from the batch processed last (or scheduled first backwardly), then additional processing time or waiting time from the batch completion time to the common due date will increase as the batch sequence number increases.

The increasing processing time phenomena in scheduling problems have been investigated in many literatures (for examples, see Holloway and Nelson, 1974; Dror, 1992; and Ravinder and Schultz, 1992). Chen et.al. (2004) has also given a literature review on scheduling problems with conditional processing time situations. The conditional processing times could be viewed as a decreasing processing time due to learning effect (Wang et.al., 2008; Yang and Kuo, 2007) or increasing processing time as consequences from naturally a deterioration process of production system (Oron, 2008, Kuo and Yang 2007, Ji and Chen, 2006, or sequence of processing (Mosheiov and Sarig, 2007, Bachman and Janiak, 2004). However, most of this research concerning the conditional processing time are generally only on scheduling problems for jobs.

This research investigates batch scheduling for a single item processed by a single machine in which processing time of a part increases proportionally with waiting time interval from its batch completion time until the common due date. The objective function for this batch scheduling problems is to minimize total actual flow time which is defined as interval between batch arrival time to its due date (Halim, 1993). The batch scheduling problems with increasing processing times could be formulated into a non linear programming model. Decision variables of this scheduling problem are the number of batches (N), batch sizes (Q), and a sequence of the resulting batches. To obtain the optimal solution of the batch scheduling problems, we relax the decision variable N to be a parameter. Using an optimization software package, we could obtain optimal solution. Based on the relaxation approach, the optimal solution is searched starting with an initial solution for N = 1. Then step by step,
we increase the value of \( N \) by 1 and find the solution for respective \( N \). Every time the solution for an \( N \) is obtained, the solution is compared to the previous solution. If the resulting total actual flow time is decreased then the best solution is updated. This iteration will be stopped if the solution could not be improved.

This paper will consist several sections. The description of batch scheduling problems dealt in this research are explained in the second section. While the proposed mathematical model is given in the third section. An experiment to demonstrate an application of the batch scheduling problem with increasing processing times and their characteristics is shown in the fourth section. The last section will present conclusions and the future research concerning the batch scheduling problems with increasing processing time.

2. BATHC SCHEDULING

2.1. Basic Model of Batch Scheduling

The motivation behind a batching decision is to share a single setup to process a number of parts. Parts sharing a single setup are called as a batch. Grouping parts into batches could reduce setup times, but it could increase flow time (see Baker [1974] and others for the definition of flow time) as finished parts should stay in the batch until all parts in the batch are competed. In other words, total flow time for a batch is defined as a multiplication of the batch processing time and the number of parts (the batch size) in the batch. It means that there is a trade off between reducing setup time and increasing total flow time in a batch scheduling problem.

Flow time of a batch in Dobson (1987) is defined with the assumption that all batches are ready to be processed at the beginning time of the scheduling period and the approach for scheduling batches the so-called forward scheduling. This assumption is not true in the real situation as the ready times of processing batches are controllable. In addition, the delivery time should at their respective due date. Therefore, Halim (1993) proposed a new scheduling criteria called actual flow time. The total actual flow time of all parts in a batch is counted by multiplying the batch processing time with the batch size. Adopting the actual flow time as the scheduling criterion means the solution is based on the so-called backward approach. Figure 1 shows the differences between the flow time and the actual flow time in dealing with batch scheduling problems.

By referring to Figure 1, we may formulate the total actual flow time for \( N \) batches as follows:

\[
F^a = \sum_{i=1}^{N} \left( d_i - B_{[i]} \right) Q_{[i]}
= \sum_{i=1}^{N} \left( (i-1)s + \sum_{m=1}^{i} t_{m} Q_{[m]} \right) Q_{[i]}
\]  

(1)

The decision variables for the batch scheduling problem are how many number of batches (\( N \)), how many parts grouped in each respective batches (\( Q_{[i]} \)), and what the sequence of the resulting batches. The objective is to minimize the total actual flow time as shown in Equation (1). The constraints that should be considered are the available production time, the total number of parts to be batched should be equal to the number of demanded parts, the completion time of the batch scheduled first from the common due date direction should be exactly at the due date, the number of batches should be in integer values, and batch size could not be as negative values.

Halim (1993) has solved this batch scheduling problem using Lagrange Relaxation method. The solution shows that LPT scheduling rules (started from due date) adopted to schedule the \( N \) batches with constant setup time will result minimum actual flow time.

2.2. Increasing Processing Time

In the batch scheduling problem for a single machine processing a single item, finished batches should wait at the shop floor until all parts in the batch have been completed. In such a situation, the quality of finished parts in a batch will naturally be deteriorated during their batch waiting time intervals, thus additional processing times are required to maintain the quality. These additional processes are to ensure that all parts in all the batches could be delivered at their due date in the required quality. In other words, processing time of a part depends upon the waiting time (time interval) between the batch completion time and its due date. The waiting times for respective batches will depend on the resulting schedule. The first backwardly scheduled batch will therefore have no waiting time as its completion time is exactly at the due date. The second batch should wait for its delivery as long as the sum of setup time and batch processing time for the first batch, and so on. This study assumes that processing times of batches will increase linearly with the increase of waiting times of respective finished batches.
Figure 1. The flow time and actual flow time in batch scheduling

Figure 2. Batch positions in a schedule

Figure 2 will be used to formulate the increasing processing time situation in a batch scheduling problem. We define the following notations:

- \( c \): completion time.
- \( d \): due date,
- \( T \): initial processing time per part,
- \( \delta \): increasing rate of processing time per part per unit waiting time,

From Figure 2, we can formulate processing time for an item for batch at first position \( t_{[1]} \), second position \( t_{[2]} \) and third position \( t_{[3]} \) as follows:

\[
\begin{align*}
\quad t_{[1]} &= T + \delta(d - c_{[1]}) = T \\
\quad t_{[2]} &= T + \delta(d - c_{[2]}) = T + \delta(s + t_{[1]}Q_{[1]}) \\
\quad t_{[3]} &= T + \delta(d - c_{[3]}) = T + \delta(2s + t_{[1]}Q_{[1]} + t_{[2]}Q_{[2]})
\end{align*}
\]

Therefore processing time per item for batch at position \( i \) could be formulated as follow:
This paper uses the following notations.

Decision variables
- \( N \) : the number of batches,
- \( Q_{[i]} \) : size of batch (unit) at position \( i \),
- \( B_{[i]} \) : starting time for processing batch at position \( i \).

Parameters
- \( i, m \) : index for batch position : 1,2,..., \( N \),
- \( D \) : demand (unit),
- \( d \) : due-date,
- \( T \) : initial processing time in constant values (unit time/unit),
- \( \delta \) : increasing rate of processing time (unit time/unit/ unit waiting time),
- \( s \) : setup time per batch.

The following assumptions are also adopted in formulating a proposed model of the batch scheduling problems:

a. This scheduling problem is for a single item processed on a machine.
b. Processing time is increasing proportionally with the increase of waiting time.
c. The beginning time of scheduling period is \( t = 0 \).
d. Batch size is real positive value.
e. The position of a batch is backwardly counted from the due date direction.

The batch scheduling problem with increasing processing time could be presented into a non-linear programming as follows.

\[
t_{[m]} = T + \delta \left( (m-1)s + \sum_{k=0}^{m-1} t_{[k]}Q_{[k]} \right) \quad (5)
\]

where \( t_{[0]} = T, Q_{[0]} = 0, m = 1,2,..., N \).

Equation (5) shows a formulation of increasing processing time in the batch scheduling discussed in this research. By substituting equation (5) into equation (1), we obtain a formulation of total actual time for \( N \) batches with increasing processing time as follow:

\[
F^a = \sum_{i=1}^{N} \left( (i-1)s + \sum_{k=0}^{m-1} \left[ T + \delta \left( (m-1)s \right. \right. \right. \\
\left. \left. \left. + \sum_{k=0}^{m-1} t_{[k]}Q_{[k]} \right) \right] \right) Q_{[i]} \quad (6)
\]

3. MODEL FORMULATION AND SOLUTION

3.1. Model Formulation

Equation (7) formulates an objective function of minimizing the total actual flow time that is evaluated as summation of multiplication between batch size and its processing time from starting processing time to due date for \( N \) batches. Available production time constraint to execute those \( N \) batches is shown in equation (8) and (9).

\[
\text{Min} \sum_{i=1}^{N} \left( (i-1)s + \sum_{k=0}^{m-1} \left[ T + \delta \left( (m-1)s \right. \right. \right. \\
\left. \left. \left. + \sum_{k=0}^{m-1} t_{[k]}Q_{[k]} \right) \right] \right) Q_{[i]} \quad (7)
\]

with constraints

\[
(N-1)s + \sum_{k=0}^{m-1} \left( T + \delta \left( (m-1)s \right. \right. \left. + \sum_{k=0}^{m-1} t_{[k]}Q_{[k]} \right) \right) Q_{[i]} \leq d \quad (8)
\]

\[
B_{[1]} + TQ_{[1]} = d \quad (9)
\]

\[
\sum_{i=1}^{N} Q_{[i]} = D \quad (10)
\]

\[
Q_{[0]} = 0 \quad (11)
\]

\[
t_{[0]} = T \quad (12)
\]

\[
Q_{[i]} \geq 0, \; N \geq 1, i = 1,2,..., N \quad (13)
\]

Equation (7) formulates an objective function of minimizing the total actual flow time that is evaluated as summation of multiplication between batch size and its batch flow time from starting processing time to due date for \( N \) batches. Available production time constraint to execute those \( N \) batches is shown in equation (8) and (9).

Equation (8) is the constraint to explain that processing time required to finish all batches should be less or same with available production interval from beginning time of scheduling \( (t = 0) \) until the due date. While equation (9) shows that the first batch should be finished exactly at due date. Total parts in all batches should same with the demand quantity. This constraint is presented in equation (10).

Equation (11) declares there is no batch in zero position. Equation (12) is to set initial processing time to standard value. Non negative constraints for decision variables are stated in equation (13) such as batch size should be in real value and number of batches should be in integer.

3.2. Solution Method

To solve the non-linear programming model of batch scheduling with increasing processing time, we apply relaxation approach. The decision variable \( N \) is relaxed to becoming a parameter. Using the relaxation approach, we could use general optimization software to find the optimal solution on batch sizes \( Q_{[i]} \) for certain value of \( N \) from the non-linear programming model presented in equation (7) to (13).

Feasible minimum value of \( N \) is 1 and the maximum value of \( N \) occurs when all parts in the first batch \( (Q_{[1]} = D) \) and sizes of other batches are zero. Maximum \( N \) value
(\(N_{\text{max}}\)) could be determined as follows:

\[
N_{\text{max}} = \left[ \frac{(d - (D \times T))}{s} \right] + 1
\] (14)

Optimal solution of the batch scheduling with increasing processing time is started with \(N = 1\) and then its solution becomes initial solution. Step by step, value of \(N\) is increased by one and its solution will be compared with the best solution of previous value of \(N\). This searching process will be stopped when the solution of current value of \(N\) has longer total actual flow time than the best solution of previous step.

Based on that relaxation approach, we develop detail heuristic procedures to solve the batch scheduling problem with increasing processing time as follow

1. Calculate \(N_{\text{max}}\) according to equation (14). If \(N_{\text{max}} > 1\) then go to step (b), other wise STOP because there is no feasible solution.

2. Solve the non linear programming model (equation (7) to (13)) with \(N = 1\) to determine total actual flow time and optimal batch size. Set current solution of total actual flow time as the best solution \((Z')\) and optimal batch size as \(Q_{\text{[i]}}^*\). Go to step (c).

3. Increase number of batches, \(N = N + 1\), and set initial value of \(Q_{\text{[i]}} = D/N\), \(i = 1, 2, \ldots, N\). Go to step (d).

4. Based on the current value of \(N\) and initial value of \(Q_{\text{[i]}}\), solve the non linear programming model (equation (7) to (13)) to determine optimal batch size \((Q_{\text{[i]}})\). Calculate total actual flow time according \(Q_{\text{[i]}}\) and then set current solution of total actual flow time as \(Z\). Go to step (e).

5. If \(Z < Z'\) then \(Z = Z'\); \(Q_{\text{[i]}} = Q_{\text{[i]}}^*\), \(N_{\text{optimal}} = N\). Go to (f).

6. If \(N = N_{\text{max}}\). Then got to step (g), other wise go to step (c).

7. STOP, optimal solution has been found.

4. EXPERIMENT AND MODEL ANALYSIS

An experiment is done to demonstrate how to solve the batch scheduling problem with increasing processing time formulated into a non-linear programming model. Before the non linear programming model is applied, we perform model verification and validation. When the proposed non-linear model is used for non-increasing processing time \((\delta = 0)\), then its solution should be return to solution of the original model of batch scheduling (Halim, 1993). Results of model verification and validation are shown in Table 1. Those results are proven that solution of the proposed model is exactly same with solution of the original model. It means the proposed model of batch scheduling with increasing processing time was valid and verified.

After the validation and verification of the proposed model was proven, then we applied it to solve two cases with different increasing rate of processing time \((\delta)\). Results of searching optimal solution of those cases are given in Table 2. In Table 2, we show step by step solution to find minimum total actual flow time \((F^*)\) and optimal batch size \((Q_{\text{[i]}}^*)\). When \(\delta = 0.001\) the searching was stop at \(N = 5\) in which the optimal solution is at \(N = 4\). While when \(\delta = 0.005\) the searching was stop at \(N = 4\) in which the optimal solution is at \(N = 3\).

<table>
<thead>
<tr>
<th>(D)</th>
<th>(t)</th>
<th>(D)</th>
<th>(s)</th>
<th>Solutions of Original model ((\text{Halim, 1993}))</th>
<th>(Q_{\text{[i]}}^*)</th>
<th>Solutions of Proposed Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.5</td>
<td>50</td>
<td>2</td>
<td>(Q_{\text{[1]}} = 26.667) (Q_{\text{[2]}} = 22.667) (Q_{\text{[3]}} = 18.667) (Q_{\text{[4]}} = 16.667) (Q_{\text{[5]}} = 10.667) (Q_{\text{[6]}} = 6.667)</td>
<td>1</td>
<td>5000.0 (Q_{\text{[1]}} = 100)</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>3346.7</td>
<td>6</td>
<td>3346.7</td>
</tr>
<tr>
<td>120</td>
<td>2</td>
<td>7</td>
<td>3345.1</td>
<td>(Q_{\text{[1]}} = 26.286) (Q_{\text{[2]}} = 22.286) (Q_{\text{[3]}} = 18.286) (Q_{\text{[4]}} = 14.286) (Q_{\text{[5]}} = 10.286) (Q_{\text{[6]}} = 6.286)</td>
<td>1</td>
<td>5000.0 (Q_{\text{[1]}} = 100)</td>
</tr>
<tr>
<td>200</td>
<td>1</td>
<td>200</td>
<td>1</td>
<td>40000.0 (Q_{\text{[1]}} = 200)</td>
<td>1</td>
<td>40000.0 (Q_{\text{[1]}} = 200)</td>
</tr>
<tr>
<td>220</td>
<td>3</td>
<td>28566.7</td>
<td>(Q_{\text{[1]}} = 76.667) (Q_{\text{[2]}} = 66.667)</td>
<td>3</td>
<td>28566.7 (Q_{\text{[1]}} = 76.667) (Q_{\text{[2]}} = 66.667) (Q_{\text{[3]}} = 56.667)</td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>6</td>
<td>27458.3</td>
<td>(Q_{\text{[1]}} = 58.333) (Q_{\text{[2]}} = 48.333) (Q_{\text{[3]}} = 38.333) (Q_{\text{[4]}} = 28.333) (Q_{\text{[5]}} = 18.333) (Q_{\text{[6]}} = 8.333)</td>
<td>6</td>
<td>27458.3 (Q_{\text{[1]}} = 58.333) (Q_{\text{[2]}} = 48.333) (Q_{\text{[3]}} = 38.333) (Q_{\text{[4]}} = 28.333) (Q_{\text{[5]}} = 18.333) (Q_{\text{[6]}} = 8.333)</td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Step by step solution for batch scheduling with increasing processing time

<table>
<thead>
<tr>
<th>N</th>
<th>(F^\alpha)</th>
<th>(Q_{(1)})</th>
<th>(Q_{(2)})</th>
<th>(Q_{(3)})</th>
<th>(Q_{(4)})</th>
<th>(Q_{(5)})</th>
<th>(Q_{(6)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5000.0</td>
<td>(Q_{(1)}=100)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3911.3</td>
<td>(Q_{(1)}=53.596)</td>
<td>(Q_{(2)}=46.404)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3601.6</td>
<td>(Q_{(1)}=39.078)</td>
<td>(Q_{(2)}=33.015)</td>
<td>(Q_{(3)}=27.906)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3485.8</td>
<td>(Q_{(1)}=32.800)</td>
<td>(Q_{(2)}=27.116)</td>
<td>(Q_{(3)}=22.216)</td>
<td>(Q_{4}=17.869)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3504.0</td>
<td>(Q_{(1)}=40.703)</td>
<td>(Q_{(2)}=28.349)</td>
<td>(Q_{(3)}=18.441)</td>
<td>(Q_{(4)}=10.144)</td>
<td>(Q_{5}=2.364)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5000.0</td>
<td>(Q_{(1)}=100)</td>
<td>(Q_{(2)}=Q_{(3)}=Q_{(4)}=Q_{(5)}=Q_{(6)}=0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. CONCLUSION

The phenomena on increasing processing time in line with waiting time from batch completion time until its due date has effects on determine decision variables of batch scheduling, e.g. number of batches \(N\), batch size \(Q\), and processing sequence of resulting batches. This batch scheduling problem could be formulated into a non linear programming model. By applying relaxation on decision variable \(N\), we could determine the optimal solution of the non linear programming model through iterations of heuristic procedure. The results of experiments showed that total actual flow time has a convex relation with number of batches \(N\).
REFERENCES


AUTHOR BIOGRAPHIES

Sukoyo is a senior lecturer and PhD candidate in Industrial Engineering and Management at the Institut Teknologi Bandung (ITB). His main research interest is in operation management. His current research focuses on an integrated production scheduling and maintenance for batch production facility in just-in-time environment. Sukoyo graduated from ITB in Industrial Engineering and Management program.

TMA Ari Samadhi is a Associate Professor at Industrial Engineering Department, Institut Teknologi Bandung (ITB), Bandung, Indonesia. He holds doctorate degree from The University of New South Wales, Australia. His main research interest is in computer integrated manufacturing. His email address is asamadhi@mail.itb.ac.id

Bermawi P. Iskandar is a Professor of Industrial Engineering at Industrial Engineering Department, Institut Teknologi Bandung (ITB), Bandung, Indonesia. He received his BS and MS degrees in Industrial Engineering from ITB, Indonesia, and his Doctorate degree from University of Queensland, Australia. His current research interests are maintenance and reliability, warranty, and industrial systems design. He published his papers in International Journal of Reliability and Application, RAIRO-Operations Research, Mathematical and Computer Modelling, Computer & Operations Research and several n Indonesian. His email address is bermawi@lspitb.org

Abdul Hakim Halim is a Professor of Industrial Engineering at Industrial Engineering Department, Institut Teknologi Bandung (ITB), Bandung, Indonesia. He received his BS and MS degrees in Industrial Engineering from ITB, Indonesia, and his Doctorate degree from Industrial Engineering Department, Osaka Prefecture University, Osaka, Japan. His research interests are in production scheduling, inventory control, FMS, and JIT systems. He published his papers in International Journal of Production Research, European
Journal of Operational Research, International Journal of Production Economics, and Production Planning and Control, as well as in Journal of Japan Industrial Management Association written in Japanese and several national journals on production systems written in Indonesian. His email address is ahakimhalim@lspitb.org