5.1 Introduction

Ignall and Schrage [37] proposed an algorithm to optimize makespan for multi-machine and \( n \) jobs. As the problem size increases, NP-completeness of flowshop problems necessitates the development of heuristics to get near optimal solutions. Campbell et al. [16] proposed a heuristic algorithm to minimize the makespan. In their work, they split the given \( m \) machine problem into a series of equivalent two machine flowshop problem and solved it using Johnson’s method. Another heuristic was proposed by Nawaz et al. [50] for minimizing makespan time.

Apart from the makespan objective, other significant objectives like total flow time, tardiness and idle time of machines have been proposed by different authors. Ho and Chang [36] have attempted not only minimizing makespan but also minimizing total flow time and machine idle time. Chou and Lee [21] attempted to solve two machine flow-shop bi-criteria scheduling problem with release dates for the jobs, in which the objective function was to minimize the weighted sum of total flow time and makespan. Rajendran and Ziegler [59] considered the objective of minimizing the total weighted flow times of jobs. Rajendran and Chaudhuri [58] proposed a heuristic algorithm to minimize the total flow time by approaching the problem in multistage.
Maggu and Das [44] introduced the concept of transportation time in going from one stage to the other for a system in which an infinite number of transport agents were available and no transport agent was required to return to stage 1 from stage 2. Chandramouli [18] derived a simple heuristic algorithm for a 3-Machine and n-job flow-shop scheduling problem in which jobs were attached with weights to indicate their relative importance. Khodadadi [41] considered the problem of three machines in tandem with two transportation times in transporting the items from machine 1 to machine 2 and from machine 2 to machine 3. It was assumed that machine 1 starts processing the next item immediately after finishing the preceding one.

In addition to this, we have considered the loading and unloading times of all the jobs for the transport agent as the loading and unloading times may not be negligible if the size of items is large. Therefore, the chapter under consideration deals with the problem of three machines in tandem including the loading times, transportation times and unloading times for jobs to be transported. It also considers breakdown time of machines and weights of jobs according to their importance in the sequence. A heuristic approach has been made for finding optimal schedule.

5.2 Problem Description

Let us consider $n$ items $(I_1, I_2, \ldots, I_n)$ being processed through three machines $(A, B$ and $C)$ in the order $ABC$. Transport agents are available, who transport an item processed at machine $A$ to the machine $B$ and then from machine $B$ to machine $C$. Let $t_i$ and $g_i$ be the transportation times for item $i$ to be carried from machine $A$ to $B$ and $B$ to $C$ respectively. Let item $i$ to be transported
from machine $A$ to machine $B$ requires loading and unloading times denoted by $l_{ab_i}$ and $u_{ab_i}$ respectively and from machine $B$ to machine $C$ requires loading and unloading times denoted by $l_{bc_i}$ and $u_{bc_i}$ respectively. The problem is to find an optimal schedule of items to minimize the total production time for completing all the items.

5.3 Development of Procedure

We develop the procedure with the help of following lemma and theorem to provide an optimal schedule:

Lemma. If $\min\{A_i + l_{ab_i} + t_i + u_{ab_i}\} \geq \max\{B_i + l_{ab_i} + t_i + u_{ab_i}\}$, then $CA_p + l_{ab_p} + t_p + u_{ab_p} \geq CB_{p-1}$.

Proof: Consider the statement $P(q)$, for an arbitrary number $q$, defined as:

$P(q): CA_{q+1} + l_{ab_{q+1}} + t_{q+1} + u_{ab_{q+1}} \geq CB_q \ (q = 1, 2, \ldots)$

For any arbitrary natural number $q$

$CA_1 = A_1$
$CB_1 = A_1 + l_{a_1} + t_1 + u_{a_1} + B_1$
$CA_2 + l_{ab_2} + t_2 + u_{ab_2} = A_1 + A_2 + l_{ab_2} + t_2 + u_{ab_2}$

Since $\min\{A_i + l_{ab_i} + t_i + u_{ab_i}\} \geq \max\{B_i + l_{ab_i} + t_i + u_{ab_i}\}$

$\Rightarrow CA_2 + l_{ab_2} + t_2 + u_{ab_2} \geq CB_1$.

Hence $P(q)$ be true for $q = 1$.

Let $P(q)$ be true for $q = m$, i.e.,

$CA_{m+1} + l_{ab_{m+1}} + t_{m+1} + u_{ab_{m+1}} \geq CB_m$. 
Optimization in Scheduling Problems

Now

\[ CB_{m+1} = \max \left\{ CA_{m+1} + l_{ab_{m+1}} + t_{m+1} + u_{ab_{m+1}} , CB \right\} + B_{m+1} \]

\[ = CA_{m+1} + \left( l_{ab_{m+1}} + t_{m+1} + u_{ab_{m+1}} + B_{m+1} \right). \]

But \( CA_{m+2} + l_{ab_{m+2}} + t_{m+2} + u_{ab_{m+2}} = CA_{m+1} + A_{m+2} + l_{ab_{m+2}} + t_{m+2} + u_{ab_{m+2}} \)

And \( A_{m+2} + l_{ab_{m+2}} + t_{m+2} + u_{ab_{m+2}} \geq B_{m+1} + l_{ab_{m+1}} + t_{m+1} + u_{ab_{m+1}} \)

Hence \( CA_{m+2} + l_{ab_{m+2}} + t_{m+2} + u_{ab_{m+2}} \geq CB_{m+1}. \)

Therefore, \( P(q) \) is true for \( q = m+1. \)

**Theorem** An optimal sequence is obtained by sequencing the item \( i-1, i, i+1 \) such that:

\[
\begin{align*}
\min \{ A_i + l_{ab_i} + t_i + u_{ab_i} + B_i + l_{bc_i} + g_i + u_{bc_i} + l_{ab_{i+1}} + \\
+ t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}} + C_{i+1} \} < \min \{ A_{i+1} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + g_{i+1} + \\
+ u_{bc_{i+1}}, l_{ab_i} + t_i + u_{ab_i} + B_i + l_{bc_i} + g_i + u_{bc_i} + C_i \}
\end{align*}
\]

**Proof:** Let \( S \) and \( S' \) denote the sequences of items given by:

\[
S = (I_1, I_2, \ldots, I_{i-1}, I_i, I_{i+1}, I_{i+2}, \ldots, I_n)
\]

\[
S' = (I'_1, I'_2, \ldots, I'_{i-1}, I'_i, I'_{i+1}, I'_{i+2}, \ldots, I'_n)
\]

Let \( (X_p, X'_p) \) and \( (CX_p, C'X_p) \) be respectively the processing time and completion time of any item \( p \) on machine \( X (= A \text{ or } B \text{ or } C) \) for the sequences \((S, S'). \) Let \( (t_p, t'_p) \) and \( (g_p, g'_p) \) denotes the transportation times of item \( p \) to transport it from machine \( A \) to machine \( B \) and from machine \( B \) to machine \( C \)
respectively for the sequences \((S,S')\). Let \((l_{ab_p}, t'_{ab_p})\) and \((u_{ab_p}, u'_{ab_p})\) be respectively the loading and unloading times of an item \(p\) in transporting it from machine \(A\) to machine \(B\) and \((l_{bc_i}, l'_{bc_i})\) and \((u_{bc_i}, u'_{bc_i})\) be the loading and unloading time in transporting item \(p\) from machine \(B\) to machine \(C\) respectively for the sequences \((S,S')\).

The completion time of \(p^{th}\) item on machines \(B\) and \(C\) is given by

\[
CB_p = \max \left( CA_p + l_{ab_p} + t_p + u_{ab_p} , CB_{p-1} \right) + B_p
= CA_p + l_{ab_p} + t_p + u_{ab_p} + B_p
\]

\[
CC_p = \max \left\{ CB_p + l_{bc_p} + g_p + u_{bc_p} , CC_{p-1} \right\} + C_p
= \max \left\{ CA_p + l_{ab_p} + t_p + u_{ab_p} + B_p + l_{bc_p} + g_p + u_{bc_p} , CC_{p-1} \right\} + C_p
\]

(5.1)

Now, we will choose the sequence \(S\) if

\[
CC_n < C'C_n
\]

(5.2)
i.e., if

\[
\max \left( CA_n + l_{ab_n} + t_n + u_{ab_n} + B_n + l_{bc_n} + g_n + u_{bc_n} , CC_{n-1} \right) + C_n
< \max \left( C'A_n + l'_{ab_n} + t'_n + u'_{ab_n} + B'_n + l'_{bc_n} + g'_n + u'_{bc_n} , C'C_{n-1} \right) + C'_n
\]

Now

\[
CA_n + l_{ab_n} + t_n + u_{ab_n} + B_n + l_{bc_n} + g_n + u_{bc_n}
= C'A_n + l'_{ab_n} + t'_n + u'_{ab_n} + B'_n + l'_{bc_n} + g'_n + u'_{bc_n}
\]

and \(C_n = C'_n\), so the result Eq. (5.2) will be true if:

\[
CC_{n-1} < C'C_{n-1}
\]

(5.3)

Proceeding in this way we get that inequality Eq. (5.2) is true if:

\[
CC_p < C'C_p \left( p = i+1, i+2, \ldots, n \text{ and } i = 1, 2, \ldots, n-1 \right)
\]

(5.4)
We now calculate the values of $CC_{i+1}$ and $C'C_{i+1}$

\[
CC_{i+1} = \max \left\{ CB_{i+1} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}}, CC_i \right\} + C_{i+1}
\]

\[
= \max \left\{ CA_{i+1} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}}, CC_i \right\} + C_{i+1}
\]

\[
CC_{i+1} = \max \left\{ CA_{i+1} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}}, CB_{i} + l_{bc_{i}} + g_{i} + u_{bc_{i}} + C_{i}, CC_{i-1} + C_{i} \right\} + C_{i+1}
\]

\[
= \max \left\{ CA_{i+1} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + \right.
 \left. g_{i+1} + u_{bc_{i+1}}, \max \left\{ CA_i + l_{ab_i} + t_i + u_{ab_i}, CB_{i-1} \right\} + \right.
 \left. B_i + l_{bc_i} + g_i + u_{bc_i} + C_i, CC_{i-1} + C_i \right\} + C_{i+1}
\]

\[
CC_{i+1} = \max \left\{ CA_{i+1} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}}, \right.
 \left. CA_i + l_{ab_i} + t_i + u_{ab_i} + B_i + l_{bc_i} + g_i + u_{bc_i} + C_i, CC_{i-1} + C_i \right\} + C_{i+1}
\]

(5.5)

Similarly

\[
C'C_{i+1} = \max \left\{ C'A_{i+1} + A'_i + A'_{i+1} + l'_{ab_{i+1}} + t'_{i+1} + u'_{ab_{i+1}} + B'_{i+1} + \right.
 \left. l'_{bc_{i+1}} + g'_{i+1} + u'_{bc_{i+1}}, C'_{i+1} + A'_i + l'_{ab_i} + t_i + u'_{ab_i} + \right.
 \left. B'_i + l'_{bc_i} + g'_i + u'_{bc_i}, C'_i + C'_{i+1}, C'C_{i-1} + C'_i + C'_{i+1} \right\}
\]

(5.6)

Comparing the sequences $S$ and $S'$, it is obvious that

\[
CA_{i+1} = C'A_{i+1}, CC_{i-1} = C'C_{i-1}
\]

\[
X_i = X'_{i+1}, X_{i+1} = X'_i (X = A, B or C)
\]
Optimization in Scheduling Problems

\[ t_i = t'_i, t_{i+1} = t'_i, g_i = g'_i, g_{i+1} = g'_i \]  \hspace{1cm} (5.7)

\[ l_{ab} = l'_{ab}, \quad l'_{ab}; \quad l_{bc} = l'_{bc}; \quad l'_{bc} \]  

\[ g_{ab} = g'_ab, \quad g_{ab}; \quad g_{bc} = g'_bc; \quad g_{bc} = g'_bc \]

Writing Eq. (5.4) for \( p = i + 1 \) and using Eq. (5.7), we get

\[
\max \left\{ CA_{i+1} + A_i + A_{i+1} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}} + C_{i+1}, CA_{i+1} + A_i + l_{ab} + t_i + u_{ab} + B_i + l_{bc} + g_i + u_{bc} + C_i \right\} < \max \left\{ CA_{i+1} + A_i + A_{i+1} + l_{ab} + t_i + u_{ab} + B_i + l_{bc} + g_i + u_{bc} + C_i, CA_{i+1} + A_i + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + B_{i+1} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}} + C_{i+1}, CC_{i+1} + C_i + C_{i+1} + C_i \right\}
\]

Subtracting last term from both sides and further subtracting \( CA_{i+1} + A_i + A_{i+1} + l_{ab} + t_i + u_{ab} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + l_{bc} + g_i + u_{bc} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}} + B_i + B_{i+1} + C_i + C_{i+1} \) from each side, we get

\[
\max \left\{ -l_{ab} - t_i - u_{ab} - l_{bc} - g_i - u_{bc} - B_i - C_i, -A_{i+1} - l_{ab_{i+1}} - t_{i+1} - u_{ab_{i+1}} - l_{bc_{i+1}} - g_{i+1} - u_{bc_{i+1}} - B_{i+1} \right\} < \max \left\{ -l_{ab_{i+1}} - t_{i+1} - u_{ab_{i+1}} - l_{bc_{i+1}} - g_{i+1} - u_{bc_{i+1}} - B_{i+1} - C_i, -A_i - l_{ab} - t_i - u_{ab} - l_{bc} - g_i - u_{bc} - B_i \right\}
\]

\[
\min \left\{ A_i + l_{ab} + t_i + u_{ab} + l_{bc} + g_i + u_{bc} + B_i, l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}} + B_{i+1} + C_{i+1} \right\} < \min \left\{ A_{i+1} + l_{ab_{i+1}} + t_{i+1} + u_{ab_{i+1}} + l_{bc_{i+1}} + g_{i+1} + u_{bc_{i+1}} + B_{i+1}, l_{ab} + t_i + u_{ab} + l_{bc} + g_i + u_{bc} + B_i + C_i \right\}
\]  \hspace{1cm} (5.10)
5.3.1 Algorithm

We can summarize the utility of above theorem into following steps to give us a decomposition algorithm, i.e., numerical method to obtain optimal schedule minimizing total elapsed time for a 3-machine, n-job sequencing problem where setup and transportation times are taken into account. Our problem can be represented in tableau form as follows:

<table>
<thead>
<tr>
<th>Item i</th>
<th>Machine A (A_i)</th>
<th>l_{ab_i}</th>
<th>t_i</th>
<th>u_{ab_i}</th>
<th>Machine B (B_j)</th>
<th>l_{bc_j}</th>
<th>g_i</th>
<th>u_{bc_j}</th>
<th>Machine C (C_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A_1</td>
<td>l_{ab_1}</td>
<td>t_1</td>
<td>u_{ab_1}</td>
<td>B_1</td>
<td>l_{bc_1}</td>
<td>g_1</td>
<td>u_{bc_1}</td>
<td>C_1</td>
</tr>
<tr>
<td>2</td>
<td>A_2</td>
<td>l_{ab_2}</td>
<td>t_2</td>
<td>u_{ab_2}</td>
<td>B_2</td>
<td>l_{bc_2}</td>
<td>g_2</td>
<td>u_{bc_2}</td>
<td>C_2</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>n</td>
<td>A_n</td>
<td>l_{ab_n}</td>
<td>t_n</td>
<td>u_{ab_n}</td>
<td>B_n</td>
<td>l_{bc_n}</td>
<td>g_n</td>
<td>u_{bc_n}</td>
<td>C_n</td>
</tr>
</tbody>
</table>

where A_i, B_j, C_k are the service times on machines A, B and C respectively. l_{ab_i}, u_{ab_i} and l_{bc_j}, u_{bc_j} are respectively the loading and unloading times for transport agent in transporting it from machines A to B and B to C respectively. t_i, g_i are the transportation times of item i in carrying it from machines A to B and B to C respectively satisfying the structural relationship:

i) \( \min(A_i + l_{ab_i} + t_i + u_{ab_i}) \geq \max(B_i + l_{ab_i} + t_i + u_{ab_i}) \)

ii) \( \min(C_i + l_{bc_i} + g_i + u_{bc_i}) \geq \max(B_i + l_{bc_i} + g_i + u_{bc_i}) \)
The result of above theorem gives the following procedure for an optimal sequence:

Step 1: Assume there are two fictitious machines $G$ and $H$ in place of $A$ and $B$ respectively. Assume that the service times for these fictitious machines $G_i$ and $H_i$ are given by

$$G_i = A_i + l_{ab_i} + t_i + u_{ab_i} + l_{bc_i} + g_i + u_{bc_i} + B_i$$

$$H_i = l_{ab_i} + t_i + u_{ab_i} + l_{bc_i} + g_i + u_{bc_i} + B_i + C_i$$

Step 2: Applying Johnson’s (1954) rule to the fictitious machine times $G$ and $H$ constructed in step 1, we obtain the optimal sequence.

5.4 Flowshop Scheduling Involving Job Weights and Break-down Intervals of Machines

Let job $i$ be assigned with the weight $w_i$ according to its relative importance for performance in the given sequence. The performance measure studied is weighted mean flow time defined by:

$$\bar{F}_w = \frac{\sum_{i=1}^{n} w_i f_i}{\sum_{i=1}^{n} w_i}, \text{ where } f_i \text{ is the flow time of } i^{th} \text{ job.}$$

Let the machine break-down interval $(a,b)$ is already known to us, i.e., deterministic in nature and the break-down interval length is $b-a$, which is known. Then our aim is to find optimal or near optimal sequence of jobs to minimize the total elapsed time.
5.4.1 Algorithm

The given problem in the tabular form may be stated as follows:

<table>
<thead>
<tr>
<th>Item $i$</th>
<th>Machine $A$ ($A_i$)</th>
<th>$l_{ab_i}$</th>
<th>$t_i$</th>
<th>$u_{ab_i}$</th>
<th>Machine $B$ ($B_i$)</th>
<th>$l_{bc_i}$</th>
<th>$g_i$</th>
<th>$u_{bc_i}$</th>
<th>Machine $C$ ($C_i$)</th>
<th>$w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$A_1$</td>
<td>$l_{ab_1}$</td>
<td>$t_1$</td>
<td>$u_{ab_1}$</td>
<td>$B_1$</td>
<td>$l_{bc_1}$</td>
<td>$g_1$</td>
<td>$u_{bc_1}$</td>
<td>$C_1$</td>
<td>$w_1$</td>
</tr>
<tr>
<td>2</td>
<td>$A_2$</td>
<td>$l_{ab_2}$</td>
<td>$t_2$</td>
<td>$u_{ab_2}$</td>
<td>$B_2$</td>
<td>$l_{bc_2}$</td>
<td>$g_2$</td>
<td>$u_{bc_2}$</td>
<td>$C_2$</td>
<td>$w_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$n$</td>
<td>$A_n$</td>
<td>$l_{ab_n}$</td>
<td>$t_n$</td>
<td>$u_{ab_n}$</td>
<td>$B_n$</td>
<td>$l_{bc_n}$</td>
<td>$g_n$</td>
<td>$u_{bc_n}$</td>
<td>$C_n$</td>
<td>$w_n$</td>
</tr>
</tbody>
</table>

Then the steps are as follows:

Step 1: Modifying problem into two machines flow-shop problem using fictitious machine $G$ and $H$ as in algorithm 5.3.1, the modified problem in the tabular form is:

<table>
<thead>
<tr>
<th>Item $i$</th>
<th>Machine $G$</th>
<th>$G_i = (A_i + l_{ab_i} + t_i + u_{ab_i} + l_{bc_i} + g_i + u_{bc_i} + B_i)$</th>
<th>Machine $H$</th>
<th>$H_i = (l_{ab_i} + t_i + u_{ab_i} + l_{bc_i} + g_i + u_{bc_i} + B_i + C_i)$</th>
<th>Weight $w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$G_1$</td>
<td>$G_1$</td>
<td>$H_1$</td>
<td>$H_1$</td>
<td>$w_1$</td>
</tr>
<tr>
<td>2</td>
<td>$G_2$</td>
<td>$G_2$</td>
<td>$H_2$</td>
<td>$H_2$</td>
<td>$w_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$n$</td>
<td>$G_n$</td>
<td>$G_n$</td>
<td>$H_n$</td>
<td>$H_n$</td>
<td>$w_n$</td>
</tr>
</tbody>
</table>
Optimization in Scheduling Problems

Step 2: Find \( \min(G_i, H_i) \)

i) If \( \min(G_i, H_i) = G_i \) then define \( G'_i = G_i - w_i \) and \( H'_i = H_i \).

ii) If \( \min(G_i, H_i) = H_i \) then define \( G'_i = G_i \) and \( H'_i = H_i + w_i \).

Step 3: Define a new reduced problem in the tabular form as:

<table>
<thead>
<tr>
<th>Item ( i )</th>
<th>( G'_i )</th>
<th>( H'_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( G'_1 / w_1 )</td>
<td>( H'_1 / w_1 )</td>
</tr>
<tr>
<td>2</td>
<td>( G'_2 / w_2 )</td>
<td>( H'_2 / w_2 )</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>( n )</td>
<td>( G'_n / w_n )</td>
<td>( H'_n / w_n )</td>
</tr>
</tbody>
</table>

Determine the optimal sequence by using Johnson’s algorithm for this new reduced problem and see the effect of break-down interval \( (a, b) \) on different jobs.

Step 4: Formulate a new problem with processing time \( A'_i, B'_i \) and \( C'_i \) where

\[
A'_i = A_i + (b - a), \quad B'_i = B_i + (b - a) \quad \text{and} \quad C'_i = C_i + (b - a), \quad \text{if } (a, b) \text{ affected on job } i.
\]

and \( A'_i = A_i, \quad B'_i = B_i, \quad C'_i = C_i, \quad \text{if } (a, b) \text{ has no effect on job } i. \)

Step 5: Now repeat the procedure to get the optimal sequence.

This sequence is either optimal or near optimal for the original problem. By this sequence we can determine the total elapsed time and weighted mean-flow time.
5.5 Example Let a machine tandem queuing problem be given in the following tableau form:

<table>
<thead>
<tr>
<th>Item $i$</th>
<th>Machine A $(A_i)$</th>
<th>$l_{ab_i}$</th>
<th>$t_i$</th>
<th>$u_{ab_i}$</th>
<th>Machine B $(B_i)$</th>
<th>$l_{bc_i}$</th>
<th>$g_i$</th>
<th>$u_{bc_i}$</th>
<th>Machine C $(C_i)$</th>
<th>$w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

where the machine break-down interval is $(12,18)$.

**Solution:** Now $\min(A_i + l_{ab_i} + t_i + u_{ab_i}) = 14$

$$\max(B_i + l_{ab_i} + t_i + u_{ab_i}) = 14$$

Hence, structural condition (i) is satisfied. Now, using the step 1, the reduced problem is:

<table>
<thead>
<tr>
<th>Item $i$</th>
<th>Machine $G$</th>
<th>Machine $H$</th>
<th>Weight $w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$G_i = (A_i + l_{ab_i} + t_i + u_{ab_i} + l_{bc_i} + g_i + u_{bc_i} + B_i)$</td>
<td>$H_j = (l_{ab_i} + t_i + u_{ab_i} + l_{bc_i} + g_i + u_{bc_i} + B_i + C_i)$</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>30</td>
<td>2</td>
</tr>
</tbody>
</table>
Using steps 2 to 3 and applying Johnson’s rule, the optimal sequence is (2, 3, 1, 4). Now the effects of break-down interval (12,18) on sequence (2, 3, 1, 4) is read as follows:

<table>
<thead>
<tr>
<th>Item</th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$l_{ab}$</td>
<td>$t_i$</td>
<td>$u_{ab}$</td>
</tr>
<tr>
<td>2</td>
<td>0-6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6-10</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>10-14</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>14-23</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Hence, with effect of breakdown interval the original problem gets modified to a new problem (as per step 4).

<table>
<thead>
<tr>
<th>Item</th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$l_{ab}$</td>
<td>$t_i$</td>
<td>$u_{ab}$</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Now, repeating the procedure, we get the sequence (3, 2, 4, 1) which is optimal and the final table is:
Mean weighted flow time is

\[
\frac{34 \times 4 + (48 - 4) \times 5 + (60 - 10) \times 2 + (65 - 25) \times 3}{4 + 5 + 2 + 3} = 41.14 \text{ hrs.}
\]

Hence, the total elapsed time is 65 hrs and mean weighted flow time is 41.14 hrs.