Computer Program of Line Balancing under the Multiple Workers in Each Station (LBMW)

Fumio AKAGI*, Hirokazu OSAKI* and Susumu KIKUCHI*

(Received February 4, 1981)

Synopsis

An assembly line with no paralleling of work elements and work stations is called a serial line. The cycle time of the serial line must be at least equal to the maximum work element time. To lower the cycle time beyond the limit and increase the production rate, one may permit the paralleling of work elements or work stations.

So in this paper we propose the parallel assignment method for achieving a higher production rate. In this method, work elements are assigned to work stations under the multiple upper time limits which are the products of the various numbers of workers and the limiting cycle time.

Further we develop the computer program of the proposed method and provide an illustrative problem and computational results.

1. Introduction

The typical assembly line is serial with no paralleling of work elements and work stations [1]. The serial line has one worker in each work station. And the serial line balancing is to assign the work elements to the work stations so as to make the work content at each of the stations as close as possible to one limiting cycle time, i.e., an upper time limit of the stations. Then the sum of the times for work elements assigned to any one station (i.e., station time) does not exceed the upper time limit.

* Department of Industrial Science
Therefor the cycle time, which is the largest value among the stations times, must be at least equal to the maximum work element time. Hence the production rate depending on the cycle time is restricted by the maximum work element time. This consequently confines the wide application of line balancing methods.

An alternative way to attack the increase of the production rate (hence lowering the cycle time) is by assigning the multiple workers to a definite station, that is, paralleling the station work. The effect of this assignment is to allow the multiple workers to perform the same station work, thereby increasing the upper time limit by the number of workers at the station. In this case, the problem is how to obtain the best possible combination of the number of workers and work elements in each station so that the efficiency of line balancing may be maximized.

Then we propose the method of assigning work elements to work stations under the multiple upper time limits which are the products of the number of workers and the limiting cycle time. We call this method the line balancing method under multiple workers in each station (LBMW).

Further we develop the computer program of line balancing with the proposed method. And an illustrative problem and computational results are provided to explain the proposed method.

2. Notations

We use the following notations for the assembly line terminology.

- \( n \) : number of work elements
- \( w_k \): \( k \)th work element in the element list
- \( t_k \): performance time of \( w_k \)
- \( P_k \): set of work elements preceding to \( w_k \)
- \( M_k \): performance restrictions of \( w_k \)
- \( k=1,\ldots,n \) : work element serial number
- \( T = \sum t_k, k=1,\ldots,n \) : total work content per unit product
- \( t_{\text{max}} = \max\{t_k, k=1,\ldots,n\} \) : maximum work element time
- \( P_u \) : limiting cycle time (depending on production rate)
- \( N \) : number of work stations
- \( n_i \) : number of work elements in the \( i \)th station
- \( w_{ij} \): \( j \)th work element in the \( i \)th station
- \( t_{ij} \): performance time of \( w_{ij} \)
- \( j=1,\ldots,n_i \) : work element serial number in the \( i \)th station
- \( i=1,\ldots,N \) : station serial number
- \( T_i = \sum t_{ij}, j=1,\ldots,n_i \) : station time of the \( i \)th station
Computer Program of Line Balancing, LBMW

3. Method proposed

Work elements are assigned to work stations which have one or more workers performing the same station work. So we add the other notations as follows:

- \( m_i \): number of workers in the \( i \)th station
- \( m = \sum_{i=1}^{N} m_i \): total number of workers on the line
- \( M_{\text{max}} \): admitted maximum number of workers at the \( i \)th station
- \( Z = 1, \ldots, M_{\text{max}} \): serial number of stages

where

- \( m_i \in \{1, \ldots, M_{\text{max}}\} \)
- \( P = \max \{ T_{i, i=1, \ldots, N} / m_i \} \)
- \( E_b = \frac{T}{N \times P} \)

In this method the problem is how to obtain the combination of \( m_i \) and \( \{ \omega_j, j=1, \ldots, n_i \} \) in each station so that the ratio of \( T_i / (m_i \times P_u) \) may be maximized.

Work elements, their performance time, precedence relations, and other constraints are assumed to be given. Further \( P_u \) and \( M_{\text{max}} \) must be given to execute the proposed method LBMW. The selection of assignable work elements is based on the preceding relations and the largest candidate rule [2].

The procedure of LBMW is as follows.

Step 1. Set the initial data.
- \( R = \{ \omega_k, k=1, \ldots, n \} \): set of still unassigned work elements
- \( i = 0 \). Go to step 2.

Step 2. Proceeding to the next station.
- \( i = i + 1 \)
- \( l = 0 \). Go to step 3.

Step 3. Proceeding to the next upper time limit at \( i \)th station.
- \( l = l + 1 \)
- \( n_i(l) = 0 \), \( T_i(l) = 0 \). Go to step 4.

Step 4. Select the maximum work element \( \omega_a \) among the ones which satisfy all the following four conditions.

1. \( \omega_a \in R \)
   - \( \omega_a \) is one of the unassigned work elements.
2. \( \{ \omega_a \mid P_a \cap R = \emptyset \} \)
   - \( \omega_a \) is one of the workable work elements with precedence relations.
3. \( \{ \omega_a \mid t_a < l \times P_u - T_i(l) \} \)
\( \omega_a \) is one of the assignable work elements with the bound of slack times.

\[
(4) \{ \omega_a \mid (M = M_a \cap \mathcal{E}), j = 1, \ldots, n_a(l) \} \cup (M = \emptyset) \}
\]

\( \omega_a \) is one of the assignable work elements with the performance restrictions.

If \( \omega_a \) can be selected, go to step 5. If not, go to step 6.

Step 5. Assign the selected work element \( \omega_a \) to \( i \)th station with \( i \)th upper time limit and the following calculations are done.

\[
n_i(l) = n_i(l) + 1, \]
\[
j = n_i(l), \]
\[
\omega_{i,j}(l) = \omega_a, t_{i,j}(l) = t_a, T_i(l) = T_i(l) + t_a, R = R - \{ \omega_a \}.
\]

Return to step 4.

Step 6. Reset the assigned work elements with \( l \)th limit.

\[
R = R + \{ \omega_{i,j}(l), j = 1, \ldots, n_i(l) \}.
\]

If \( l < M_{\text{imax}} \), return to step 3. If \( l \geq M_{\text{imax}} \), go to step 7.

Step 7. Select the best combination of the number of workers and work elements at \( i \)th station. First,

\[
l_0 = \{ l \mid \max \{ T_i(l) / (l \times \mu), l = 1, \ldots, M_{\text{imax}} \}
\]

and using \( l_0 \), the following calculations are done.

\[
m_i = l_0, n_i = n_i(l_0), T_i = T_i(l_0), \]
\[
\omega_{i,j} = \omega_{i,j}(l_0), t_{i,j} = t_{i,j}(l_0), j = 1, \ldots, n_i(l_0), \]
\[
R = R - \{ \omega_{i,j}, j = 1, \ldots, n_i \}.
\]

If \( R \neq \emptyset \), return to step 2. If \( R = \emptyset \), go to step 8.

Step 8. Compute the balance and stop the procedure.

\[
N = i
\]
\[
P = \max \{ T_i / m_i, i = 1, \ldots, N \}, \quad E_b = T / (\sum m_i \times P) = T / (m \times P).
\]

4. Program

The work assignment method under the condition of multiple upper time limits is programmed in FORTRAN IV. The program is the form of subroutine and its name is LBMW.

SUBROUTINE LBMW(PU,MIMAX,NW,NWK,TWK,KINDP,NAMEP,MACHIN,NSTN,TI,MI,NI,NWIJ)

4.1. Usage

Argument list

<table>
<thead>
<tr>
<th>ARGUMENT</th>
<th>I/O</th>
<th>TYPE</th>
<th>SIZE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>I</td>
<td>REAL</td>
<td>1</td>
<td>( Pu ), the limiting cycle time</td>
</tr>
<tr>
<td>MIMAX</td>
<td>I</td>
<td>INTEGER</td>
<td>1</td>
<td>( M_{\text{imax}} ), admitted number of workers at a station</td>
</tr>
<tr>
<td>NW</td>
<td>I</td>
<td>INTEGER</td>
<td>1</td>
<td>( n ), number of work elements</td>
</tr>
<tr>
<td>NWK</td>
<td>I</td>
<td>INTEGER</td>
<td>NW</td>
<td>( \omega_k ), work element number</td>
</tr>
<tr>
<td>TWK</td>
<td>I</td>
<td>REAL</td>
<td>NW</td>
<td>( t_k ), time durations of work element</td>
</tr>
<tr>
<td>KINDP</td>
<td>I</td>
<td>INTEGER</td>
<td>NW</td>
<td>number of pre work elements</td>
</tr>
<tr>
<td>NAMEP</td>
<td>I</td>
<td>INTEGER</td>
<td>NW \times \text{KINDP}</td>
<td>( P_k ), pre work element number</td>
</tr>
<tr>
<td>MACHIN</td>
<td>I</td>
<td>INTEGER</td>
<td>NW</td>
<td>( M_k ), performance restriction</td>
</tr>
</tbody>
</table>

DEFINITION
4.2. Suggestion on using

Subroutine SWWEBR, SAWEBBS, SAWMAC and MAXGRP are used in LBMW.
These are used to select the maximum work element \( w_a \) at step 4 in the procedure of the method.

The program list of LBMW is shown in Table 1.

5. Illustrative problem and computational results

The assembly work of the small electric switches is used as an example to illustrate the proposed method. The total assembly work has been analyzed and divided into work elements. The list of work elements has been developed and shown in Table 2.

The line balancing consists of two procedures [3]:

(1) Minimize the number of workers on the line given the limiting cycle time.

(2) Minimize the cycle time given the number of workers on the line.

LBMW can be applied to both procedures.

We assume that the production schedule needs two thousands of the switches per day (420 minutes). So the cycle time of the line must be lower than 420/2000, or 0.210 minutes, and the limiting cycle time \( P_u \) is 0.210 minutes.

But the work element \( w_j \) (01) takes 0.323 minutes, which is longer than \( P_u \). In this case, the establishment of the assembly line is impossible by the serial balancing methods. So the LBMW will be applied to solve this problem.

As \( t_{max}/P_u=0.323/0.210=1.58 \), \( M_{max} \) must be at least equal to 2, and we set \( M_{max}=4 \).

5.1. Assignment under the given \( P_u=0.210 \)

Giving \( P_u=0.210 \) and \( M_{max}=4 \), LBMW is called. The step of obtaining the solution is shown in Table 3. Computational results in this case are shown in Table 4. The obtained line is constructed by 3 work stations (\( N=3 \)) and 5 workers (\( m=5 \)). The cycle time of that is 0.201 and the efficiency of line balancing is 0.909 (i.e., \( P=0.201 \) and \( E_b=0.909 \)).

In detail the 1st station has 3 workers (\( m_1=3 \)), 3 work elements (\( n_1=3 \), \( w_{1j}=(01,02,04) \)), and the station time of \( 0.602 (T_1=0.602) \). The 2nd station is \( m_2=1 \), \( n_2=2 \), \( w_{2j}=\{05,03\} \), and \( T_2=0.184 \). At the 3rd station
5.2. Assignment to lower the cycle time for 5 workers

To lower the cycle time and improve $E_b$, we use the cycle time of the previous obtained line for the next $P_u$. Giving $P_u=0.201$ (obtained in 5.1.) and $M_{max}=4$, LBMW is called. As the obtained line has 5 workers, the higher balance for $m=5$ is obtained. Computational results in this case are $N=2$, $m=5$, $P=0.186$ and $E_b=0.928$. The 1st station has 4 workers ($n_1=4$), 5 work elements ($n_2=5$, $W_{1j}=[01,02,04,05,03]$) and the station time $0.786$ ($T_1=0.786$). The 2nd station has $m_2=1$, $n_2=1$, $W_{2j}=06$ and $T_2=0.126$. They are shown in Table 5.

Further we use $P=0.186$ for $P_u$ and perform the same procedure. As the line has 5 workers, the higher balance for $m=5$ is obtained. Computational results in this case are $N=2$, $m=5$, $P=0.184$, $E_b=0.991$, shown in Table 6. The 1st station is 4 workers ($m_1=4$), 4 work elements ($n_1=4$, $W_{1j}=[01,02,04,05]$), and $T_1=0.728$. And the 2nd station is $m_2=1$, $n_2=2$, $W_{2j}=[06,03]$ and $T_2=0.184$.

Further we use $P=0.184$ for $P_u$ and perform the same procedure. In this case the obtained line has one more workers than the previous line. This means that the cycle time can not be lower than 0.184 for 5 workers on the line.

As the results, three different work assignments which satisfy the production rate 2000 units/day were obtained. One of them may be selected and adopted by the other measurements except the efficiency.

References


---

Table 2. List of work elements of a switch.

<table>
<thead>
<tr>
<th>NWK</th>
<th>TWK</th>
<th>KINDP</th>
<th>NAMEP</th>
<th>MACHIN</th>
<th>$w_k$</th>
<th>$t_k$ (min.)</th>
<th>$P_k$</th>
<th>$M_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0.323</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>0.183</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>0.088</td>
<td>0</td>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>0.126</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>0.126</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>0.126</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$t_{max}=0.323$, $T=0.912$

Table 3. Steps of obtaining the solution under $P_u=0.210$ and $M_{max}=4$.

<table>
<thead>
<tr>
<th>L</th>
<th>$T_{11}(L)$</th>
<th>$T_{11}(L)/L=0.001$</th>
<th>$P_{11}(L)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.153</td>
<td>0.729</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0.311</td>
<td>0.769</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0.127</td>
<td>0.925</td>
<td>1 2 4</td>
</tr>
<tr>
<td>4</td>
<td>0.755</td>
<td>0.936</td>
<td>1 2 4 5 3</td>
</tr>
<tr>
<td>2</td>
<td>0.184</td>
<td>0.876</td>
<td>5 3</td>
</tr>
<tr>
<td>3</td>
<td>0.313</td>
<td>0.492</td>
<td>5 6 3</td>
</tr>
<tr>
<td>4</td>
<td>0.310</td>
<td>0.169</td>
<td>5 6 3</td>
</tr>
</tbody>
</table>

In this case the obtained line has one more workers than the previous line.
Table 4. Computational results.

<table>
<thead>
<tr>
<th>N</th>
<th>T1</th>
<th>M1</th>
<th>W1</th>
<th>TW1</th>
<th>T2</th>
<th>M2</th>
<th>W2</th>
<th>TW2</th>
<th>T3</th>
<th>M3</th>
<th>W3</th>
<th>TW3</th>
<th>P</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>.502</td>
<td>3</td>
<td>01</td>
<td>.323</td>
<td>02</td>
<td>.153</td>
<td>04</td>
<td>.126</td>
<td>05</td>
<td>.350</td>
<td>.201</td>
<td>.909</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>.144</td>
<td>1</td>
<td>05</td>
<td>.126</td>
<td>03</td>
<td>.350</td>
<td>06</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>.126</td>
<td>1</td>
<td>06</td>
<td>.126</td>
<td>06</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td>.126</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Computational results.

<table>
<thead>
<tr>
<th>N</th>
<th>T1</th>
<th>M1</th>
<th>W1</th>
<th>TW1</th>
<th>T2</th>
<th>M2</th>
<th>W2</th>
<th>TW2</th>
<th>T3</th>
<th>M3</th>
<th>W3</th>
<th>TW3</th>
<th>P</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>.728</td>
<td>4</td>
<td>01</td>
<td>.323</td>
<td>02</td>
<td>.153</td>
<td>04</td>
<td>.126</td>
<td>05</td>
<td>.126</td>
<td>.196</td>
<td>.923</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>.126</td>
<td>1</td>
<td>06</td>
<td>.126</td>
<td>03</td>
<td>.350</td>
<td>03</td>
<td>.350</td>
<td>03</td>
<td>.350</td>
<td>.126</td>
<td>.126</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Computational results.

<table>
<thead>
<tr>
<th>N</th>
<th>T1</th>
<th>M1</th>
<th>W1</th>
<th>TW1</th>
<th>T2</th>
<th>M2</th>
<th>W2</th>
<th>TW2</th>
<th>T3</th>
<th>M3</th>
<th>W3</th>
<th>TW3</th>
<th>P</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>.728</td>
<td>4</td>
<td>01</td>
<td>.323</td>
<td>02</td>
<td>.153</td>
<td>04</td>
<td>.126</td>
<td>05</td>
<td>.126</td>
<td>.196</td>
<td>.923</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>.126</td>
<td>1</td>
<td>06</td>
<td>.126</td>
<td>03</td>
<td>.350</td>
<td>03</td>
<td>.350</td>
<td>03</td>
<td>.350</td>
<td>.126</td>
<td>.126</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Program list.

```plaintext
C *** CALL UNDER THE MULTIPLE WORKERS IN EACH STATION ***
SUBROUTINE LIMN(OUT, N Mat, W G, TWK, KINDP, NAME, MACHEIN, P)
NOUT=N+1, M1=1, W1=1, TW1=1
DIMENSION WAK(50), TWK(50), KINDP(50), NAMEP(50), MACHEIN(50)
DIMENSION TI(50), W1(50), TW1(50), W2(50), TW2(50)
DIMENSION TIL(10), WIL(10), TWIL(10), WIJ(10), TWIJ(10)
DIMENSION GROUP(50), WASSP(50)
DO 12 I=1, N
12 IF(WIL(I).EQ.0) GO TO 15
14 CONTINUE
GO TO 50
15 CONTINUE
LS=0
310 LS=LS+1
IF(LS.EQ.50) GO TO 515
TTEST=PU*FLOAT(LS)
TIL(LS)=0.0
NIL(LS)=0
MACRES=7
700 CONTINUE
CALL SWEER(OUT, KINDP, NAMEP, MACRES, WASSP, GROUP)
IF(LS.EQ.0) GO TO 710
CALL SWEES(OUT, MACRES, GROUP, TEST, TWK, WASSP, WASSP)
IF(WASSP.LE.0) GO TO 519
CALL SAVAC(MACRES, WASSP, WASSP, MACHEIN)
IF(WASSP.LE.0) GO TO 511
WASSP=WASSP+1
NIL(LS)=NIL(LS)+1
TIL(LS)=TIL(LS)+TWK
TTEST=TIL(LS)+TWK
IF(TIL(LS).LE.0) GO TO 15
NIL(LS)=NIL(LS)+1
W1(I)=W1(I)+1
IF(W1(I).EQ.50) GO TO 315
TTEST=TTEST+T4(MACET)
W4EST=TTEST+10000.0
TEST=TEST+1
IF(MACRES.EQ.10) MACRES=MACRES+1
GO TO 15
315 IF(LS.EQ.50) GO TO 330
END=END+TIL(LS)/(LS*PU)
```

C *** SELECT THE WORKABLE WORK ELEMENTS WITH PRECEDENT RELATION ***
DIMENSION NET(NS),I=1,NET(LS)
END

C *** SELECT THE ASSIGNABLE WORK ELEMENTS WITH THE BOUND OF SLACK TIMES ***
SUBROUTINE SAWMACS(NUMGRP,NGRP,TW,TW,K,NUMASC,NASSGP) 
DIMENSION NASSGP(50),MACHIN(50)
IF(MACHIN(NAS).LE.0) RETURN
END

C *** SELECT ASSIGNABLE WORKS WITH THE MACHINE RESTRICTIONS ***
FUNCTION MAXGRP(NUMASS,NASSGP,TWK)
DIMENSION NASSGP(50),TWK(50)
MAXGRP=NASSGP(1)
BIG=TWK(MAXGRP)
DO 10 I=1,NUMASS 
    IF(TWK(NAS).LE. BIG) GO TO 10 
    N=N-1 
    NASSGP(I)=NASSGP(I)
10 CONTINUE 
RETURN 
END

C *** LARGEST CANDIDATE RULE ***
FUNCTION MAXGRP(NUMASS,NASSGP,TWK)
DIMENSION NASSGP(50),TWK(50)
MAXGRP=NASSGP(1)
BIG=TWK(MAXGRP)
DO 10 I=1,NUMASS 
    IF(TWK(NAS).LE. BIG) GO TO 100 
    MAXGRP=N 
    BIG=TWK(N)
100 CONTINUE 
RETURN 
END