The sequence operation in production flows

Associate Professor Cătălin Angelo Ioan, PhD
„Danubius" University of Galati

Abstract: The sequence operation in production flows appears in the usual practice for the installations waiting time decreasing when a lot of pieces use the same technology line in the same direction.

Keywords: algorithm, permutation, iteration

Jel Classification: C7 , C02, C51 ,

Let two installations U_1 and U_2 who process n pieces $P_1,\ldots,P_n$ ($n\geq 2$) in the same order (first U_1 and after U_2). We shall consider that U_1 and U_2 are available from the process beginning and the waiting time to come in execution for U_2 does not implies other prices. In addition we shall suppose that the pieces do not have a finish ending date.

Let note with t_{ij} the processing time of the j-th piece on the i-th installation.

The problem consists in a determination of the pieces execution beginning order such that the waiting time of the installation U_2 to be minimum.

Let the matrix $T=(t_{ij})\in M_{2n}(\mathbf{R})$ of the time processing. The classical algorithm consists in the following steps:

Step 1 We choose the least element on the first row. This will give us the first piece who will come in execution.

Step 2 We cut the previous column and we choose the least element on the second row. This will give us the last piece who will come in execution.

Step 3 We cut the previous column and we go again at the first step. After this we will obtain the second piece who will come in execution, and after we go again at the second step and we find the penultimate piece and so on.

The algorithm will continue till we shall finish all the pieces.

Example

Let two installations U_1 and U_2 who process six pieces $P_1,\ldots,P_6$ in the same order (first U_1 and after U_2). We shall consider that U_1 and U_2 are available from the process beginning and the waiting time to come in execution for U_2 does not implies other prices. In addition we shall suppose that the pieces do not have a finish ending date. We can see the execution times in the following table:

Installation /Piece	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**
U_1	16	27	12	4	22	33
U_2	10	15	2	10	11	24

For the beginning we choose the piece P_4 because the element 4 is the lowest from the first row.

After we cut the corresponding column, we obtain:

Installation/Piece	**P_1**	**P_2**	**P_3**	**P_5**	**P_6**
U_1	16	27	12	22	33
U_2	10	15	2	11	24

The least element in the second row is 2, therefore the last piece will be P_3.

In the first row of the table:

Installation/Piece	**P_1**	**P_2**	**P_5**	**P_6**
U_1	16	27	22	33
U_2	10	15	11	24

we find that the least element in the first row is 16, therefore the second piece is P_1.

In the second row of the table:

Installation/Piece	**P_2**	**P_5**	**P_6**
U_1	27	22	33
U_2	15	11	24

we find that the least element in the second row is 11, therefore the fifth piece is P_5.

In the first row of the remaining table:

Installation/Piece	**P_2**	**P_6**
U_1	27	33
U_2	15	24

we find that the least element in the first row is 27, therefore the third piece is P_2 and finally the fourth piece is the remaining P_6.

The order of execution is therefore:

P_4,P_1,P_2,P_6,P_5,P_3

The Gantt diagram illustrates the execution order of all pieces:

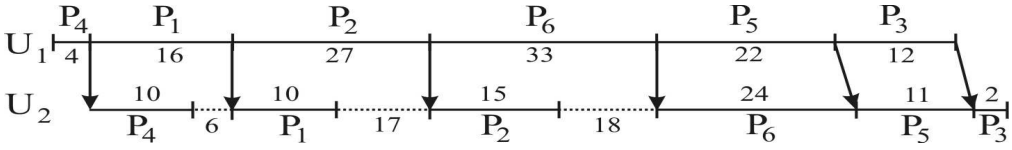

Fig. 1

The total waiting time for U_2 is the sum of the lower quantities in the figure 1 and is therefore: 6+17+18=41.

The total time of the process is 4+16+27+33+22+12+2=116.

If the process is cycling we can see that at the finish of execution, the piece P_4 will wait only 1 time unit till it will be process again on U_2.
We shall try in what follows to find all solutions of this problem to see if the method presented above is optimal.

Let consider a permutation $\sigma=\begin{pmatrix} 1 & 2 & ... & n \\ i_1 & i_2 & ... & i_n \end{pmatrix} \in S_n$ – the group of permutation of n elements and an order of pieces indexed by σ: $P_{i_1}, P_{i_2}, ..., P_{i_n}$.

Let the table of execution times:

Installation /Piece	P_{i_1}	P_{i_2}	…	P_{i_k}	…	P_{i_n}
U_1	$d_{i_1 1}$	$d_{i_2 1}$	…	$d_{i_k 1}$	…	$d_{i_n 1}$
U_2	$d_{i_1 2}$	$d_{i_2 2}$	…	$d_{i_k 2}$	…	$d_{i_n 2}$

We define $g_1, g_2, \ldots, g_n \geq 0$ - the pauses before entering in execution of $P_{i_1}, P_{i_2}, \ldots, P_{i_n}$ on the installation U_2. We have obviously:

- $g_1 = d_{i_1 1}$ (*from the beginning of the process*)
- $g_2 = \max(d_{i_1 1} + d_{i_2 1} - d_{i_1 2} - g_1, 0)$
- $g_3 = \max(d_{i_1 1} + d_{i_2 1} + d_{i_3 1} - d_{i_1 2} - d_{i_2 2} - g_1 - g_2, 0)$

…

- $g_k = \max\left(\sum_{p=1}^{k} d_{i_p 1} - \sum_{p=1}^{k-1} d_{i_p 2} - \sum_{p=1}^{k-1} g_p, 0\right)$

…

- $g_n = \max\left(\sum_{p=1}^{n} d_{i_p 1} - \sum_{p=1}^{n-1} d_{i_p 2} - \sum_{p=1}^{n-1} g_p, 0\right)$

If we note: $B_{i_1 \ldots i_k} = \sum_{p=1}^{k} d_{i_p 1} - \sum_{p=1}^{k-1} d_{i_p 2}$ we have:

- $g_1 = B_{i_1}$
- $g_2 = \max(B_{i_1 i_2} - g_1, 0)$
- $g_3 = \max(B_{i_1 i_2 i_3} - g_1 - g_2, 0)$

…

- $g_k = \max\left(B_{i_1 \ldots i_k} - \sum_{p=1}^{k-1} g_p, 0\right)$
- $g_n = \max\left(B_{i_1 \ldots i_n} - \sum_{p=1}^{n-1} g_p, 0\right)$

The objective function is $z=\min\left(\sum_{k=2}^{n} g_k\right)$.

The algorithm will compute for all permutations $\sigma=\begin{pmatrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n \end{pmatrix} \in S_n$ the quantities $g_1,\dots,g_n$ and after the values of z.

For the example presented above we have the order: P_6, P_1, P_2, P_4, P_5, P_3 and the Gantt diagram is the following:

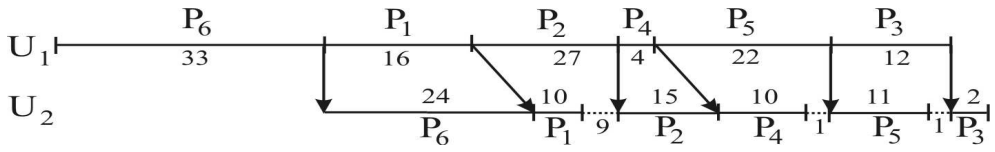

Fig. 2

The total waiting time for U_2 is the sum of the lower quantities in the figure 1 and is therefore: 9+1+1=11.

The total time of the process is 33+16+27+4+22+12+2=116 the same like in the previous method.

If the process is cycling we can see that at the finish of execution, the piece P_6 will wait 31 time unit till it will be process again on U_2.

After this example we can see that if we analyse all the permutations the total time seems to remain constant, but the waiting time for the second installation descrease very much. If the process cycle the total time in the first case is lower then in the second case presented upper, but if we made all the permutations we can find also a better choice. Indeed, after all permutations, we find the following order: P_1,P_2,P_3,P_4,P_5,P_6 and after the finish of the processhe piece P_1 will not wait any moment. After the second iteration we shall find that the total waiting time is: 17+11+22+16-24=42 and in the first method 43.

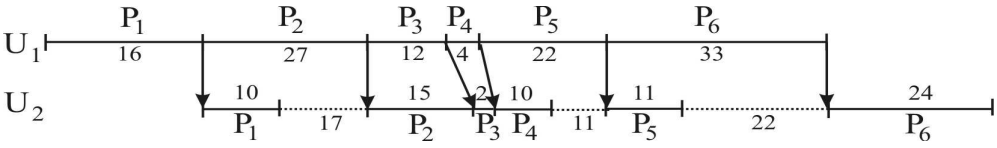

Fig. 3