Applied and Computational Mathematics
2016; 5(1): 1-9
Published online January 15, 2016 (http://www.sciencepublishinggroup.com/j/acm)
doi: 10.11648/j.acm.20160501.11
ISSN: 2328-5605 (Print); ISSN: 2328-5613 (Online)

New Algorithm for N-jobs on M-machine Flow Shop Scheduling Problems

Geleta Tadele Mohammed

Department of Mathematics, College of Natural Sciences, Arba Minch University, Arba MInch, Ethiopia

Email address:
geletatadele@gmail.com, geleta.tadele@amu.edu.et, geletatadele@yahoo.com

To cite this article:
Geleta Tadele Mohammed. New Algorithm for N-jobs on M-machine Flow Shop Scheduling Problems. *Applied and Computational Mathematics*. Vol. 5, No. 1, 2016, pp. 1-9. doi: 10.11648/j.acm.20160501.11

Abstract: Because of flow shop scheduling is one of the most important problems in the area of production management, In this paper, what I have did is that, I have developed a new algorithm for n-jobs m-machine flow shop scheduling problem for special case of n-jobs m-machine flow shop scheduling problem. And also, there is a good self explanatory example to explain the algorithm very well.

Keywords: N-jobs M-machine, Flow Shop Scheduling Problems

1. Introduction and Statement of the Problem

In this paper the n-jobs m-machines flow shop scheduling problem where processing times of all jobs are deterministic and known before processing is started is studied with the introduction of concepts of optimal value of processing time multiple, assignment of optimal due date and determination of optimal sequence of jobs by minimizing total squared values of lateness. And also the work has been supported by numerical examples and, where n, m are positive integers.

First, for m=1, the problem was studied by cheng [1]:

All jobs become available for processing at the same time, and they require t_i units for processing for $i = 1{:}\,n$. Where n is the number of jobs.

If d_i denotes the assigned due date of job i and expressed by

d_i= kt_i , $i = 1{:}\,n$, where k is processing time multiple.

Since cost will certainly be incurred whenever a job cannot be completed exactly on its due date, be it early or tardy. It is natural to have minimization of total missed due dates as the objective, for which the total squared values of lateness L^2 is defined as the performance measure.

The subscript i denote the job occupying the i^{th} position for an arbitrary sequence of jobs. If L_i C_i and d_i respectively denotes the lateness, completion time and assigned due date of the job in position i, then

$$L^2=\sum_{i=1}^{n} {L^2}_i = \sum_{i=1}^{n}(C_i - d_i)^2$$

This, $L^2 = \sum_{i=1}^{n}\left[\sum_{j=1}^{i}(t_j) - kt_i\right]^2$, where C_i=$\sum_{j=1}^{i} t_j$

Note: Here,

- The purpose of square is to consider as we need to minimize both of early and tardiness at the same time.
- t_i is the processing time of i^{th} position job on the given (only one) machine.
- d_i= kt_i , $i = 1{:}\,n$, where k is processing time multiple is the approximation of due dates of each jobs. i.e. In this case due dates of each job is not given, so, we have determine them optimally.
- C_i=$\sum_{j=1}^{i} t_j$ is Completion time of i^{th} position job is the sum of all processing time of jobs before i^{th} position and i^{th} position itself on machine (here we have only one machine).
- Hence, cheng [5] obtained an optimal value of processing time multiple as:

$$k^* = \frac{\sum_{i=1}^{n} t_i \sum_{j=1}^{i} t_j}{\sum_{i=1}^{n} [t_i]^2}$$

After we have obtained the optimum value of k which is k^*, then we can obtain due dates of each jobs. Then, we can use E.D.D (earliest due date) rule to obtain the optimum arrangement or sequence of jobs. This is all about the work done by cheng on n-jobs 1-machine scheduling problem.

Secondly, for m=2, the problem was studied by Ikram [2]:

In short, Ikram's work was an extension of change [1] work for two machines, with having the following core condition:

- It is applied for the minimum processing times of all jobs on the first machine is greater than or equal to the maximum of processing times of all jobs on the second machine.

In this study, I extend the work of change [1] and Ikram [2] to special class of n-jobs m-machine scheduling problem.

***Table 1.** N-jobs m-machines scheduling problem.*

Jobs	Machines					
	A_1	A_2	A_3	...	$A_{(m-1)}$	A_m
1	A_{11}	A_{21}	A_{31}		$A_{(m-1)1}$	A_{m1}
2	A_{12}	A_{22}	A_{32}		$A_{(m-1)2}$	A_{m2}
3	A_{13}	A_{23}	A_{33}		$A_{(m-1)3}$	A_{m3}
.	.	.	.		.	.
.	.	.	.		.	.
.	.	.	.		.	.
n	A_{1n}	A_{2n}	A_{3n}	...	$A_{(m-1)n}$	A_{mn}

Note that: In the above table and throughout this paper, A_{ij} represents the processing time of job j on machine i.

The problem is to determine an optimal value of processing time multiple, assign due date and obtain optimal sequence of jobs which minimizes L^2 (total sum of squared values of lateness).

2. Problem Formulation

2.1. Formulation of Completion Time

- *Flow time* of job i is the time difference between its completion time on the last machine and starting time of it on the first machine. If we consider starting time of the first machine is fixed at zero, then simply flow time is the same for all jobs. But, completion time of job is dependent on sequence of jobs. i.e. Completion time of one job is different from sequence to sequence. A completion time of each job always occurs on the last machine.

Now, the formulation of completion time each job is the following:

Here, as in the case of Ikram [2], the order machines is fixed and it is $A_1 \to A_2 \to A_3 \to \cdots \to A_{(m-1)} \to A_m$. Then the corresponding completion time can be calculated as follows in the table below:

***Table 2.** Completion time of jobs in n-jobs m-machines scheduling problem.*

Jobs			Machines			
	A_1		A_2		...	A_m
	In	Out	In	Out	In	Out
1	0	A_{11}	A_{11}	$A_{11} + A_{21}$	$A_{11}\ A_{21}+\ldots+A_{(m-1)1}$	$A_{11}\ A_{21}+\ldots+A_{(m-1)1}+A_{m1}$
2	A_{11}	$A_{11}+A_{12}$	$A_{11}+A_{12}$	$A_{11}+A_{12}+A_{22}$	$A_{11}\ A_{12}+A_{22}+\ldots+A_{(m-1)2}$	$A_{11}\ A_{12}+A_{22}+\ldots+A_{(m-1)2}+A_{m2}$
3	$A_{11}+A_{12}$	$A_{11}+A_{12}+$ A_{13}	$A_{11}+A_{12}+$ A_{13}	$A_{11}+A_{12}+A_{13}+$ A_{23}	$A_{11}\ A_{12}+A_{13}+A_{23}\ldots+A_{(m-1)3}$	$A_{11}\ A_{12}+A_{13}+A_{23}+\ldots+A_{(m-1)3}+$ A_{m3}
.						.
.						.
.						.
n	$A_{11}\ A_{12}+$ $A_{13}+A_{14}+\ldots+$ $A_{1(n-1)}$	$A_{11}\ A_{12}+$ $A_{13}+A_{14}+\ldots$ $+A_{1(n-1)}$ $+A_{1n}$	$A_{11}\ A_{12}+$ $A_{13}+A_{14}+$ $\ldots+A_{1(n-1)}$ $+A_{1n}$	$A_{11}\ A_{12}+A_{13}+A_{14}$ $+\ldots+A_{1(n-1)}$ $+A_{1n}+A_{11}\ A_{12}+$ $A_{13}+A_{14}+\ldots+$ $A_{1(n-1)}$ $+A_{2n}$	$A_{11}\ A_{12}+A_{13}+A_{14}+\ldots+$ $A_{1(n-1)}$ $+A_{1n}+A_{11}\ A_{12}+A_{13}+A_{14}+\ldots+$ $A_{1(n-1)}$ $+A_{2n}+A_{3n}+A_{4n}+\ldots+A_{(m-1)n}$	$A_{11}\ A_{12}+A_{13}+A_{14}+\ldots+A_{1(n-1)}$ $+A_{1n}+A_{11}\ A_{12}+A_{13}+A_{14}+\ldots+$ $A_{1(n-1)}$ $+A_{2n}+A_{3n}+A_{4n}+\ldots+A_{(m-1)n}$ $+A_{mn}$

Now, the completion times for jobs 1, 2, and 3 are given below:

Job1: $A_{11}\ A_{21}+\ldots+A_{(m-1)1}+A_{m1}$

Job2: $A_{11}\ A_{12}+A_{22}\ +\ldots+A_{(m-1)2}+A_{m2}$

Job3: $A_{11}\ A_{12}+A_{13}+A_{23}+\ldots+A_{(m-1)3}+A_{m3}$

And also, the completion time of job done on i^{th} position place is given accordingly from the above table as:

$C_i = \left[\sum_{j=1}^{i} A_{1j}\right] + A_{2i} + A_{3i} + \cdots + A_{mi}$, for i=1: n

2.2. Problem Formulation and Assumption

In addition to the information we have about this problem above, let us consider n-jobs with deterministic processing times processed on m-machine and the same readiness times, the problem is to find the optimal processing time multiple k^* for due date assignment problem, the optimal sequence σ^* to minimize the total amount of missed due dates. It is found that k^* is constant for a given job set and σ^* should be in S.P.T (short processing time) rule.

The basic assumptions for the problem are as follows:

1. All processing times are deterministic and known before the starting time of all jobs.
2. The machine cannot simultaneously process two or more jobs at the same time.
3. Jobs Pre-emption is not required.

And also consider the following conditions:

$$Min\,\{A_{1i}\} \geq Max\{A_{2i}\} \quad (1.1)$$

$$Min\,\{A_{2i}\} \geq Max\{A_{3i}\} \quad (1.2)$$

$$Min\,\{A_{3i}\} \geq Max\{A_{4i}\} \quad (1.3)$$

.

.

.

$$Min\,\{A_{(m-1)i}\} \geq Max\{A_{mi}\} \quad (1.m)$$

And if $A_{1x} > A_{1y}$, then the following conditions holds true:

$$A_{1x}\, A_{2y} \ \geq\ A_{1y}\ A_{2x} \quad (2.0)$$

$$A_{1x}\, A_{3y} \ \geq\ A_{1y}\ A_{3x} \quad (2.1)$$

$$A_{1x}\, A_{4y} \ \geq\ A_{1y}\ A_{4x} \quad (2.2)$$

.
.
.

$$A_{1x}\, A_{my} \ \geq\ A_{1y}\ A_{mx} \quad (2.m)$$

Since all jobs spend more amount of time on the first machine, then d_j= kA_{1j} , $j = 1{:}\,n$, where k is processing time multiple. This is because of the condition (1.0) to (1.m).

Let σ denotes an arbitrary sequence, and $[j]$ represents jobs occupying j^{th} position of σ . If $L_{[j]}, C_{[j]}\ and\ d_{[j]}$ represents lateness, completion time and assigned due date oj the job in the j^{th}, then our objective function is to minimize:

$$L^2=\sum_{i=1}^{n} L^2{}_{[i]}=\sum_{i=1}^{n}(C_{[i]} - d_{[i]})^2 \quad (3)$$

Since the completion time of the j^{th} for any sequence σ is: $C_{[j]} = \left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]}$, for j =1: n and by using $d_{[j]}$= k$A_{[1j]}$, $j = 1{:}\,n$ equation (3) becomes:

$$L^2=\sum_{j=1}^{n}\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]} - \mathrm{k}A_{[1j]}]^2 \quad (4)$$

This is a function of k and to be minimized.

2.3. Optimal Due Date Assignment Procedure

Since the function to be minimized has degree 2, then from calculus concept one can find the value of k that minimize L^2 as follows by using chain rule:

$$\frac{dL^2}{dk} = \frac{d}{dk}\left[\sum_{j=1}^{n}\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]} - \mathrm{k}A_{[1j]}]^2\right]$$

$$=-2\sum_{j=1}^{n} A_{[1j]}\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]} - \mathrm{k}A_{[1j]}]^2$$

$$\frac{dL^2}{dk} = 0\ if\ and\ only\ if -2\sum_{j=1}^{n} A_{[1j]}\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]} - \mathrm{k}A_{[1j]}]^2 = 0$$

This implies that

$$\sum_{j=1}^{n} A_{[1j]} \sum_{i=1}^{j} A_{[1i]} + \sum_{j=1}^{n} A_{[1j]}\, A_{2j} + \sum_{j=1}^{n} A_{[1j]}\, A_{[3j]} + \sum_{j=1}^{n} A_{[1j]}\, A_{[4j]} + \ldots + \sum_{j=1}^{n} A_{[1j]}\, A_{[mj]} - k\sum_{j=1}^{n}[A_{[1i]}]^2 = 0$$

Which gives

$$\mathrm{k}^* = \frac{\sum_{j=1}^{n} A_{[1j]} \sum_{i=1}^{j} A_{[1i]} + \sum_{j=1}^{n} A_{[1j]} A_{2j} + \sum_{j=1}^{n} A_{[1j]} A_{[3j]} + \sum_{j=1}^{n} A_{[1j]} A_{[4j]} + \ldots + \sum_{j=1}^{n} A_{[1j]} A_{[mj]}}{\sum_{j=1}^{n}[A_{[1j]}]^2} \quad (5)$$

According to change [6], $\sum_{j=1}^{n} A_{[1j]}\, A_{2j}$, $\sum_{j=1}^{n} A_{[1j]}\, A_{[3j]}$, $\sum_{j=1}^{n} A_{[1j]}\, A_{[4j]}, \ldots, \sum_{j=1}^{n} A_{[1j]}\, A_{[mj]}$, $\sum_{j=1}^{n}[A_{[1i]}]^2$ are constant and independent of the sequence of the jobs.

And also, $\sum_{j=1}^{n} A_{[1j]} \sum_{i=1}^{j} A_{[1i]}$ is constant and independent of the sequence of the jobs.

Therefore, k* is constant and independent of the sequence of the jobs.

Hence, by using the value of k*, we assign the value of due dates of each to be

$d_{[j]}$= k*$A_{[1j]}$, $j = 1{:}\,n$

Theorem (Minimization of L^2 under a certain condition):

Our objective function $L^2 = \sum_{j=1}^{n}\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]} - \mathrm{k}A_{[1j]}]^2$ can be minimized by applying the rule that job x should be done before job y if the following conditions are satisfied:

$$A_{1x} > A_{1y} \quad (6.1)$$

$$A_{1x}\, A_{2y} \ \geq\ A_{1y}\ A_{2x} \quad (6.2)$$

$$A_{1x}\, A_{3y} \ \geq\ A_{1y}\ A_{3x} \quad (6.3)$$

$$A_{1x}\, A_{4y} \ \geq\ A_{1y}\ A_{4x} \quad (6.4)$$

$$A_{1x}\, A_{my} \ \geq\ A_{1y}\ A_{mx} \quad (6.m)$$

Proof:

Since, $L^2 = \sum_{j=1}^{n}[\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]} - \mathrm{k}A_{[1j]}]^2$, the extending this expression we can obtain that

$$L^2 = \sum_{j=1}^{n}\left\{[\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]}]^2 + \left[\mathrm{k}A_{[1j]}\right]^2 - 2\mathrm{k}A_{[1j]}\left(\left[\sum_{i=1}^{j} A_{[1i]}\right] + A_{[2j]} + A_{[3j]} + \cdots + A_{[mj]}\right)\right\}$$

By having re-arrangement of summation inside the bracket we have,

$$L^2=\sum_{j=1}^{n}[[\sum_{i=1}^{j} A_{[1i]}]+A_{[2j]}+A_{[3j]}+\cdots+A_{[mj]}]^2+\sum_{j=1}^{n}[\mathrm{k}A_{[1j]}]^2-2k\sum_{j=1}^{n}[A_{[1j]}([\sum_{i=1}^{j} A_{[1i]}]+A_{[2j]}+A_{[3j]}+\cdots+A_{[mj]})] \quad (7)$$

From the third term of Equation (10), we obtain: $\sum_{j=1}^{n}[A_{[1j]}([\sum_{i=1}^{j} A_{[1i]}]+A_{[2j]}+A_{[3j]}+\cdots+A_{[mj]})] = \sum_{j=1}^{n} A_{[1j]}\sum_{i=1}^{j} A_{[1i]}+\sum_{j=1}^{n} A_{[1j]}A_{2j}+\sum_{j=1}^{n} A_{[1j]}A_{[3j]}+\sum_{j=1}^{n} A_{[1j]}A_{[4j]}+\ldots+\sum_{j=1}^{n} A_{[1j]}A_{[mj]}$, which is constant and independent of the sequence of the jobs change [1].

And also, the middle term $\sum_{j=1}^{n}[\mathrm{k}A_{[1j]}]^2$ is constant (because of it is the sum of square quantity) and independent of sequence of the jobs.

Now, the remaining thing to prove is that $\sum_{j=1}^{n}[[\sum_{i=1}^{j} A_{[1i]}]+A_{[2j]}+A_{[3j]}+\cdots+A_{[mj]}]^2$ can be minimized by applying the conditions given above in the theorem?

The answer is yes!

Let σ_1 be a sequence of jobs in which job x and y are arranged in a position k and k+1 respectively, and σ_2 be a sequence of the jobs in which job x and y are arranged in apposition of k+1 and k respectively.

Let

$$f(\sigma_1)=\sum_{j=1}^{n}[[\sum_{i=1}^{j} A_{[1i]}]+A_{[2j]}+A_{[3j]}+\cdots+A_{[mj]}]^2 \quad (8)$$

And

$$f(\sigma_2)=\sum_{j=1}^{n}[[\sum_{i=1}^{j} A_{[1i]}]+A_{[2j]}+A_{[3j]}+\cdots+A_{[mj]}]^2 \quad (9)$$

Now, by expanding equation (8), we obtain

$$f(\sigma_1)=[A_{[11]}+A_{[21]}+A_{[31]}+\cdots+A_{[m1]}]^2+[A_{[11]}+A_{[12]}+A_{[22]}+A_{[32]}+\cdots+A_{[m2]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[23]}+A_{[33]}+\cdots+A_{[m3]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+A_{[24]}+A_{[34]}+\cdots+A_{[m4]}]^2+\ldots+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[1x]}+A_{[2x]}+A_{[3x]}+\cdots+A_{[mx]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[1x]}+A_{[1y]}+A_{[2y]}+A_{[3y]}+\cdots+A_{[my]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k+2)]}+A_{[2(k+2)]}+A_{[3(k+2)]}+\cdots+A_{[m(k+2)]}]^2+\ldots+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1n]}+A_{[2n]}+A_{[3n]}+\cdots+A_{[mn]}]^2 \quad (10)$$

$$f(\sigma_2)=[A_{[11]}+A_{[21]}+A_{[31]}+\cdots+A_{[m1]}]^2+[A_{[11]}+A_{[12]}+A_{[22]}+A_{[32]}+\cdots+A_{[m2]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[23]}+A_{[33]}+\cdots+A_{[m3]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+A_{[24]}+A_{[34]}+\cdots+A_{[m4]}]^2+\ldots+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[2(k-1)]}+A_{[3(k-1)]}+\cdots+A_{[m(k-1)]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[1y]}+A_{[2y]}+A_{[3y]}+\cdots+A_{[my]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots A_{[1(k-1)]}+A_{[1y]}+A_{[1x]}+A_{[2x]}+A_{[3x]}+\cdots+A_{[mx]}]^2+[A_{[11]}+A_{[12]}+A_{[13]}+\cdots+A_{[1(k+2)]}+A_{[2(k+2)]}+A_{[3(k+2)]}+\cdots+A_{[m(k+2)]}]^2+\ldots+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1n]}+A_{[2n]}+A_{[3n]}+\cdots+A_{[mn]}]^2 \quad (11)$$

$$\text{Now, } f(\sigma_1)-f(\sigma_2)=[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[1x]}+A_{[2x]}+A_{[3x]}+\cdots+A_{[mx]}]2+[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[1x]}+A_{[1y]}+A_{[2y]}+A_{[3y]}+\cdots+A_{[my]}]^2-[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[1y]}+A_{[2y]}+A_{[3y]}+\cdots+A_{[my]}]-[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots A_{[1(k-1)]}+A_{[1y]}+A_{[1x]}+A_{[2x]}+A_{[3x]}+\cdots+A_{[mx]}]^2 \quad (12)$$

Then by simplifying equation (12) we get

$$f(\sigma_1)-f(\sigma_2)$$

$$=A_{[1x]}[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}+A_{[1x]}+A_{[1y]}+A_{[2y]}+A_{[3y]}+\cdots+A_{[my]}]-A_{[1y]}[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots A_{[1(k-1)]}+A_{[1y]}+A_{[1x]}+A_{[2x]}+A_{[3x]}+\cdots+A_{[mx]}]$$

$$=A_{[1x]}A_{[11]}+A_{[1x]}A_{[12]}+A_{[1x]}A_{[13]}+\ldots+A_{[1x]}A_{[1(k-1)]}+A_{[1x]}A_{[1x]}+A_{[1x]}A_{[1y]}+A_{[1x]}A_{[2y]}+A_{[1x]}A_{[3y]}+\ldots+A_{[1x]}A_{[my]}-A_{[1y]}A_{[11]}-A_{[1y]}A_{[12]}-A_{[1y]}A_{[13]}-A_{[1y]}A_{[14]}-A_{[1y]}A_{[1(k-1)]}-A_{[1y]}A_{[1x]}-A_{[1y]}A_{[2x]}-A_{[1y]}A_{[3x]}-A_{[1y]}A_{[4x]}-\cdots-A_{[1y]}A_{[mx]}$$

$$=[A_{[1x]}]^2-[A_{[1y]}]^2+(A_{[1x]}-A_{[1y]})[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}]+[A_{[1x]}A_{[1y]}+A_{[1x]}A_{[2y]}+A_{[1x]}A_{[3y]}+A_{[1x]}A_{[4y]}+\cdots+A_{[1x]}A_{[my]}-A_{[1y]}A_{[1x]}-A_{[1y]}A_{[2x]}-A_{[1y]}A_{[3x]}-A_{[1y]}A_{[4x]}-\cdots-A_{[1y]}A_{[mx]}]$$

$$=[A_{[1x]}]^2-[A_{[1y]}]^2+(A_{[1x]}-A_{[1y]})[A_{[11]}+A_{[12]}+A_{[13]}+A_{[14]}+\cdots+A_{[1(k-1)]}]+[A_{[1x]}A_{[1y]}-A_{[1y]}A_{[1x]}]+[A_{[1x]}A_{[2y]}-A_{[1y]}A_{[2x]}]+[A_{[1x]}A_{[3y]}-A_{[1y]}A_{[3x]}]+[A_{[1x]}A_{[4y]}-A_{[1y]}A_{[4x]}]+\cdots+[A_{[1x]}A_{[my]}-A_{[1y]}A_{[mx]}]$$

Now, if $A_{1x} > A_{1y}$

$$A_{1x}\, A_{2y} \;\geq\; A_{1y}\;\, A_{2x}$$

$$A_{1x}\, A_{3y} \;\geq\; A_{1y}\;\, A_{3x}$$

$$A_{1x}\, A_{4y} \;\geq\; A_{1y}\;\, A_{4x}$$

$$A_{1x}\, A_{my} \;\geq\; A_{1y}\;\, A_{mx}\;,$$

Then, $f\ (\sigma_1) > f\ (\sigma_2)$.
Thus, the interchanging of job x and y reduces the value of L^2.

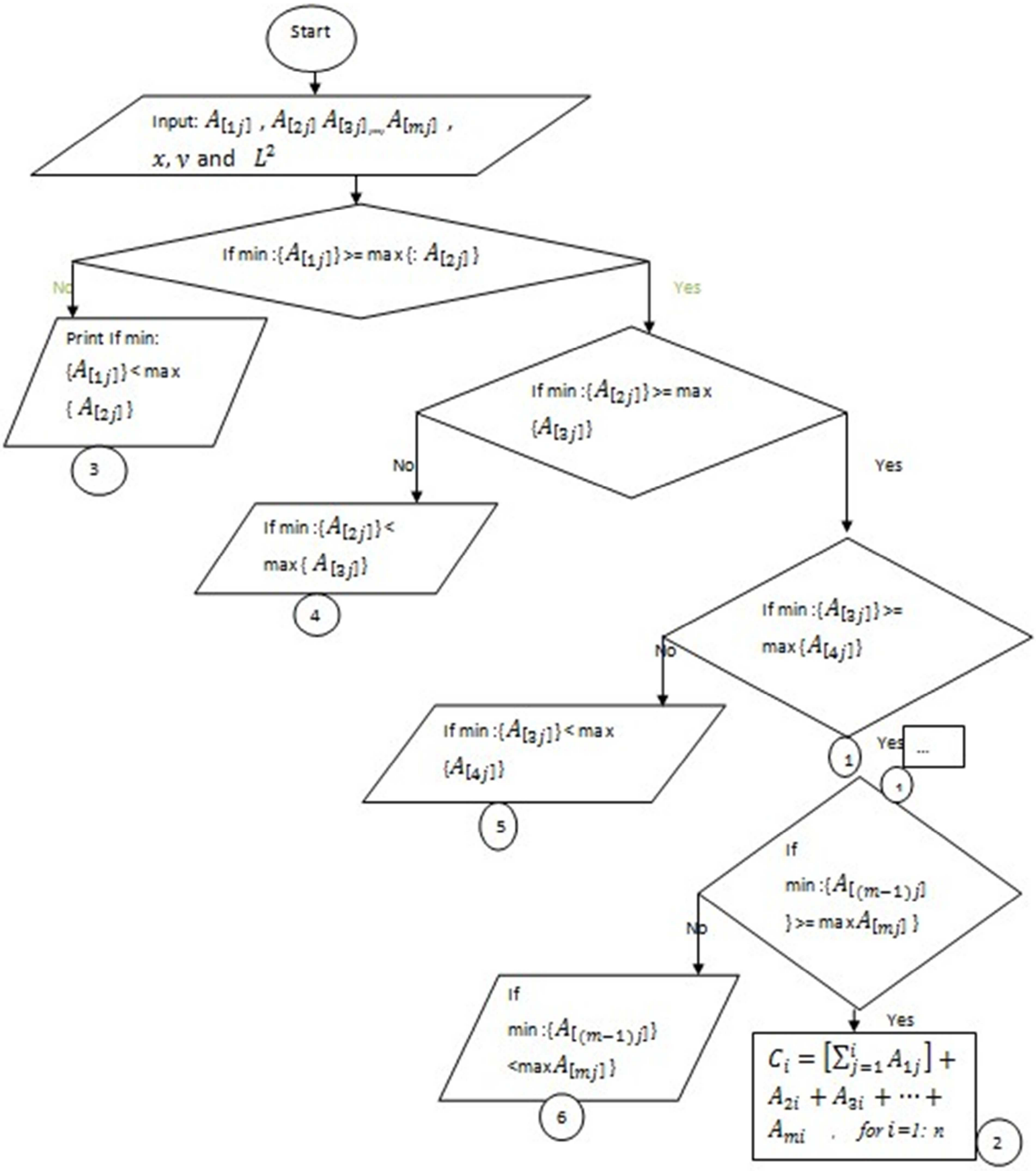
Hence, job y should be done before job x.

Therefore, the conditions stated in the above theorem are satisfied.

Now, by repeatedly applying the above rule, L^2 can be minimized by arranging jobs depending on their processing time on the first machine as S.P.T rule.

In this study, we developed the following flowchart for the above theorem and show how L^2 is minimized.

2.4. Flow Chart



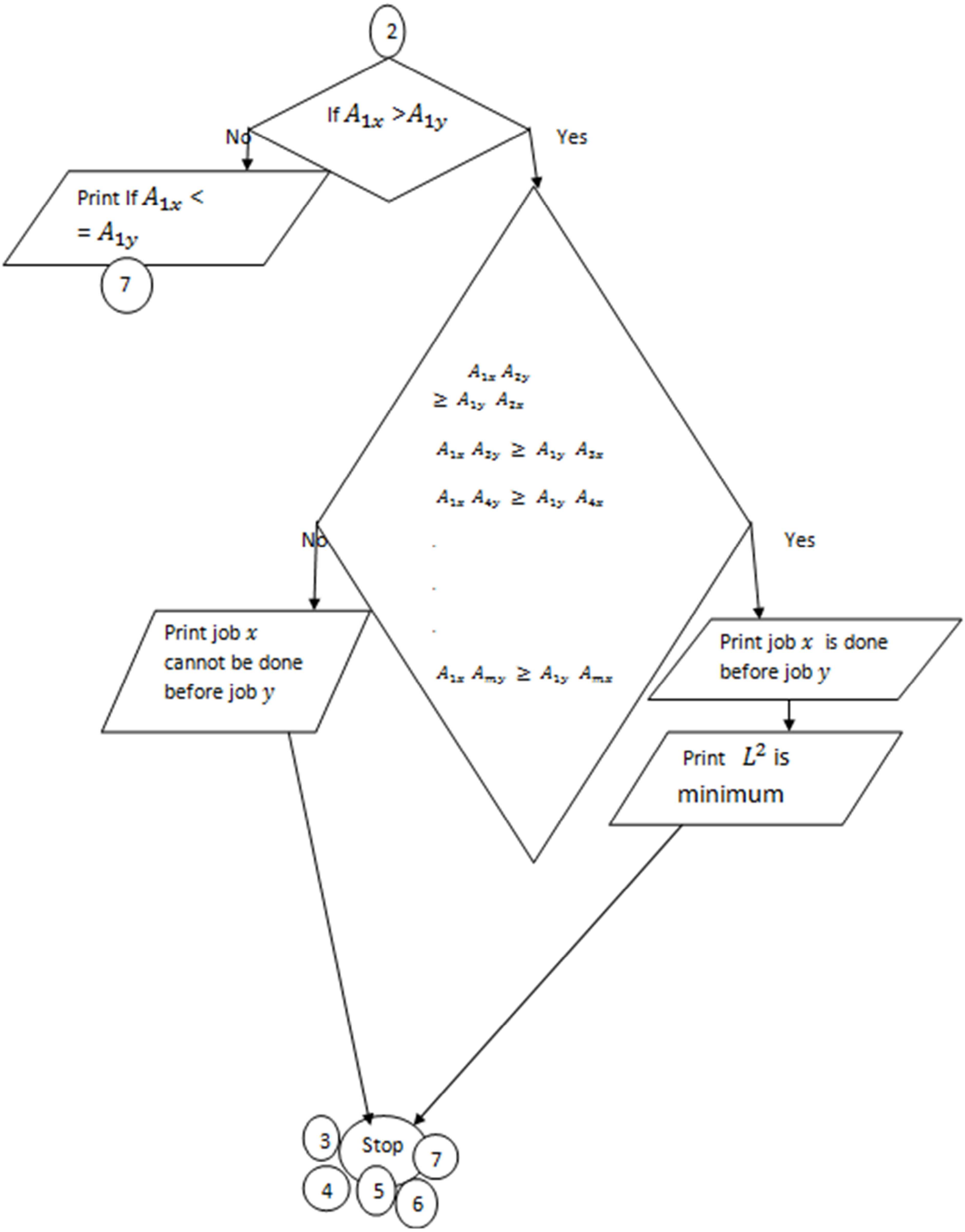


Figure 1. *Flow chart of how to Get Optimal Sequence.*

3. Algorithms to Get Optimal Sequence

Step 1*:* Verify that the conditions given from (1.0)-(1.m) and (2.0) – (2.m). If all of these conditions holds true proceed to the next step. Else stop.

Step 2*:* Determine the values of k^* using by the formula (5).

Step 3*:* By using shortest processing time rule on the first machine A_1 determine the optimal sequence of jobs.

Step 4*:* Finally, find L^2 for the obtained optimal sequences of jobs.

Example

For the following 3-jobs 3-machine flow shop scheduling problem, find the optimal sequence of jobs such that L^2 is minimum.

Table 3. *3-jobs 3-machine flow shop scheduling problem.*

Jobs	Machines		
	A_1	A_2	A_3
1	$A_{11}=12$	$A_{21}=6$	$A_{31}=3$
2	$A_{12}=10$	$A_{22}=8$	$A_{32}=5$
3	$A_{13}=11$	$A_{23=7}$	$A_{33}=4$

Solution:

Here, what we have to do is that, according to the above algorithm, we have to find:-

- k^*
- Due-dates of each job.
- An optimal sequence.

Step 1:

Clearly, $min\{12,10,11\}=10>max\{6,8,7\}=8$

$min\{6,7,8\}=6>max\{3,5,4\}=5$

And also, let say job 1 is x and job 2 is y.

Then, $A_{1x}A_{2y}$= (12) (8) =96

$A_{1x}A_{3y}$= (12) (5) =60

$A_{1y}A_{2x}$= (10) (7) =70

$A_{1y}A_{3x}$= (10) (3) =30

This implies that

$$\begin{cases} A_{1x}A_{2y}=\ (12)(8)=96>A_{1y}A_{2x}=70 \\ A_{1x}A_{3y}=\ (12)(5)=60>\ A_{1y}A_{3x}=30 \end{cases}$$

Therefore, all conditions for step 1 are satisfied.

Step 2:

Since,

$$k^*=\frac{\sum_{j=1}^{n}A_{[1j]}\sum_{i=1}^{j}A_{[1i]}+\sum_{j=1}^{n}A_{[1j]}A_{2j}+\sum_{j=1}^{n}A_{[1j]}A_{[3j]}+\sum_{j=1}^{n}A_{[1j]}A_{[4j]}+\ldots+\ \sum_{j=1}^{n}A_{[1j]}A_{[mj]}}{\sum_{j=1}^{n}[A_{[1j]}]^2} \quad (5)$$

Then, because of our problem is for n=3=m, k* becomes,

$$k^*=\frac{\sum_{j=1}^{3}A_{[1j]}\sum_{i=1}^{j}A_{[1i]}+\sum_{j=1}^{3}A_{[1j]}A_{2j}+\sum_{j=1}^{3}A_{[1j]}A_{[3j]}}{\sum_{j=1}^{3}[A_{[1j]}]^2}$$

$$=\frac{A_{11}(A_{11})+(A_{11}+A_{12})(A_{11}+A_{12})+(A_{11}+A_{12}\ A_{13})(A_{11}+A_{12}\ A_{13})+A_{11}A_{21}+A_{12}A_{22}+A_{13}A_{23}+A_{11}A_{31}+A_{12}A_{32}+A_{11}A_{33}}{{A_{11}}^2+{A_{12}}^2+{A_{13}}^2}$$

$$=\frac{(12)(12)+(12+10)(12+10)+(12+10+11)(12+10+11)+(12)(6)+(10)(8)+(11)(7)+(12)(3)+(10)(5)+\ (12)(4)}{(12)^2+(10)^2+(11)^2}$$

$$=\frac{2080}{365}$$

=5.70

Now, by using this values of k* we can assign the due dates of each job as follows:

Table 4. *Assign the due dates of each job by using this values of* k^*.

Jobs	Machine A_1	Due-date ($d_{[j]}$= $k^*A_{[1j]}$)
1	12	68.40
2	10	57.00
3	11	62.70

Step 3:

As indicated in the above algorithm, we arrange jobs as per shortest processing time rule on machine 1(A_1). And also, the same result we obtain, if we arrange jobs by earliest due date rule.

Therefore, by using both of them (one of them is enough), we obtain the optimal sequence of jobs 2-3-1.

Step 4:

Determination of L^2. Here, L^2 is listed in the following table for all possible sequences of jobs we have.

Table 5. *L^2 for all possible sequences of jobs.*

Sequences of jobs (σ)	Processing time multiple k*	Squared value of lateness L^2
1-2-3	5.70	3080.45
1-3-2	5.70	3191.45
3-1-2	5.70	3102.45
3-2-1	5.70	2882.45
2-3-1	5.70	2795.45
2-1-3	5.70	2904.45

Note that, L^2 is calculated by

$L^2 = \sum_{j=1}^{n}\left[\sum_{i=1}^{j}A_{[1i]}\right]+A_{[2j]}+A_{[3j]}+\cdots+A_{[mj]}-k^*A_{[1j]}]^2$, because of n=m=3 for our problem, then it becomes

$L^2=\sum_{j=1}^{3}\left[\sum_{i=1}^{j}A_{[1i]}\right]+A_{[2j]}+A_{[3j]}-\ k^*A_{[1j]}]^2$

$=(A_{11}+A_{21+}A_{31}-\ k^*A_{11})^2+(A_{11}+A_{12+}A_{22+}A_{32}-k^*A_{12})^2+(A_{11}+\ A_{12+}A_{13+}A_{23}+A_{33}-\ k^*A_{13})^2$

For instance, for the sequence of jobs2-3-1:-

Table 6. *The sequence of jobs2-3-1.*

Jobs	Machines		
	A_1	A_2	A_3
2	$A_{11}=10$	$A_{21}=8$	$A_{31}=5$
3	$A_{12}=11$	$A_{22}=7$	$A_{32}=4$
1	$A_{13}=12$	$A_{23=6}$	$A_{33}=3$

L^2= $(10+8+5-(5.70)10)^2+(10+11+7+4-(5.70)11)^2+(10+11+\ 12+6+3-(5.70)12)^2$

= $(23-57.00)^2+(32-62.70)^2+(42-68.40)^2$

= $(-34)^2+(-30.70)^2+(-26.40)^2$

=1156+ 942.49+696.96

=2795.45

Similarly, for the sequence of jobs1-3-2 we have,

Table 7. *The sequence of jobs1-3-2.*

Jobs	Machines		
	A_1	A_2	A_3
1	$A_{11} = 12$	$A_{21} = 6$	$A_{31} = 3$
3	$A_{12} = 11$	$A_{22} = 7$	$A_{32} = 4$
2	$A_{13} = 10$	$A_{23=8}$	$A_{33} = 5$

L^2=$(12+6+3-(5.70)12)^2$+$(12+11+7+4-(5.70)11)^2+(12+11+\ 10+8+5-(5.70)10)^2$

=$(21-68.40)^2$+$(34-62.70)^2$+$(46-57.00)^2$

= $(-47.4)^2+(-28.7)^2$+$(-11.00)^2$

=2246.76+823.69+121

=3191.45

For the sequence of jobs1-2-3 we have,

Table 8. *The sequence of jobs1-2-3.*

Jobs	Machines		
	A_1	A_2	A_3
1	$A_{11} = 12$	$A_{21} = 6$	$A_{31} = 3$
2	$A_{12} = 10$	$A_{22} = 8$	$A_{32} = 5$
3	$A_{13} = 11$	$A_{23=7}$	$A_{33} = 4$

L^2=$(12+6+3-(5.70)12)^2$+$(12+10+8+5-(5.70)10)^2+(12+10+\ 11+7+4-(5.70)11)^2$

=$(21-68.40)^2$+$(35-57.00)^2$+$(44-62.70)^2$

=$(-47.4)^2+(-22)^2$+$(-18.7)^2$

=2246.76+484+349.69

=3080.45

For the sequence of jobs 2-1-3 we have,

Table 9. *The sequence of jobs 2-1-3.*

Jobs	Machines		
	A_1	A_2	A_3
2	$A_{11} = 10$	$A_{21} = 8$	$A_{31} = 5$
1	$A_{12} = 12$	$A_{22} = 6$	$A_{32} = 3$
3	$A_{13} = 11$	$A_{23=7}$	$A_{33} = 4$

L^2=$(10+8+5-(5.70)10)^2$+$(10+12+6+3-(5.70)12)^2+(10+12+\ 11+7+4-(5.70)11)^2$

= $(23-57.00)^2$+$(31-68.40)^2$+$(44-62.70)^2$

= $(-34)^2+(-37.4)^2$+$(-18.7)^2$

=1156+1398.76+349.69

=2904.45

For the sequence of jobs3-1-2 we have,

Table 10. *The sequence of jobs3-1-2.*

Jobs	Machines		
	A_1	A_2	A_3
3	$A_{11} = 11$	$A_{21} = 7$	$A_{31} = 4$
1	$A_{12} = 12$	$A_{22} = 6$	$A_{32} = 3$
2	$A_{13} = 10$	$A_{23=8}$	$A_{33} = 5$

L^2=$(11+7+4-(5.70)11)^2$+$(11+12+6+3-(5.70)12)^2+(11+12+\ 10+8+5-(5.70)10)^2$

= $(22-62.7)^2$+$(32-68.40)^2$+$(46-57.00)^2$

= $(-40.7)^2+(-36.4)^2$+$(-11)^2$

=1656.49+1324.96+121

=3102.45

For the sequence of jobs3-2-1 we have,

Table 11. *The sequence of jobs3-2-1.*

Jobs	Machines		
	A_1	A_2	A_3
3	$A_{11} = 11$	$A_{21} = 7$	$A_{31} = 4$
2	$A_{12} = 10$	$A_{22} = 8$	$A_{32} = 5$
1	$A_{13} = 12$	$A_{23=6}$	$A_{33} = 3$

L^2=$(11+7+4-(5.70)11)^2$+$(11+10+8+5-(5.70)10)^2+(11+10+\ 12+6+3-(5.70)12)^2$

= $(22-62.7)^2$+$(34-57.00)^2$+$(42-68.40)^2$

=$(-40.7)^2+(-23)^2$+$(-26.4)^2$

=1656.49+529+696.96

=2882.45

Now, from the above table the optimal sequence of the given jobs is 2-3-1 with L^2 = 2795.45.

This completes our example.

Recommendation

Here what I have as recommendation is that one can do one or both of the following:

I. One can develop a program for this algorithm accordingly, in order to handle the problem for large values of n and m.

II. One can extend the condition I have used in my work for in order to handle large size of problem as much as possible.

References

[1] Chenge, T. C. E., "Optimal due date determination and scheduling of n- jobs on a single machine", Journal of the Operation Research society 35, pp. 433-437, 1984.

[2] Ikram, M., "A note on minimization of lateness Cost Function and Determination of Optimal Due – date in two machine problem," Pure Applied Mathematika sciences, Vol. XXIII, 1-2, March 1986.

[3] Johnson, S. M. "Optimal two and three stage production schedules with setup times included." Naval Research Logistics Quarterly, Vol I, 61-68, 1954.

[4] Jackson, J. R., "An extension of Johnson's results on lot scheduling", Naval Research Logistics Quarterly, Vol 3, pp.201-2003, 1956.

[5] Ibrahim, M.A., “Algorithms for Sequencing and Scheduling,” Industrial Engineering Department, College of Engineering, King saud University, Riyadh, Saudi Arabia.

[6] Osman, M. R., Ismail, N., Zairian, M. R. M, Yusuf, R. M., Sapuan, S. M. “sheet metal Fabrication Scheduling Using Selective Performance Measure and priority Dispatching Rule,” International Journal of Engineering and Technology, Vol. 1 (1), pp. 74-83, 2004.

[7] Taylor, F. W., “Shop Management”, Harper And Bros, New York, 1911.

[8] Herrmann, Jeffrey. W “A history of production Scheduling” in Handbook of Production Scheduling, Springer, New York, 2006.

[9] Dudek, R. A., S. S. Panwalkar, and M. L. Smith, “The lessons for flow shop scheduling Research,” Operation Research, Vol 40(1), pp. 7-13, 1993.

[10] Conway, R. W., Maxwell, W. L., and Miller, L. W. “Theory of Scheduling,” Addison-Wesley Publishing Company, Massachusetts, 1967.

[11] Miloš Šeda “Mathematical Models of Flow Shop and Job Shop Scheduling Problems”, International Journal of Applied Mathematics and Computer Sciences Volume 4 Number 4.

[12] Kerem Bu¨lbu¨l, Philip Kaminsky, Candace Yano,” Flow Shop Scheduling with Earliness, Tardiness, and Intermediate Inventory Holding Costs”, Industrial Engineering and Operations Research, University of California, Berkeley, California 94720-1777.

[13] Mircea ANCĂU, “on solving flow shop scheduling problems” THE PUBLISHING HOUSE PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A, OF THE ROMANIAN ACADEMY Volume 13, Number 1/2012, pp. 71–79.