Vol. XIV & Issue No. 12 December - 2021
INDUSTRIAL ENGINEERING JOURNAL

MINIMIZING MAKESPAN FOR FLOW SHOP SCHEDULING PROBLEM

Dr. S. Jayakumar
S. Jayasankari

Abstract:

In a flow shop environment, scheduling n jobs on m machines is NP-hard, and it has a prominent position in the field of production scheduling. This paper considers the scheduling of n jobs on m machines in a permutation flow shop with a makespan minimization target. The proposed solution is demonstrated in detail by giving an suitable example. Finally, the result obtained using the proposed method are compared to those obtained using a previously published method. It was found that our algorithm outperforms the other algorithms.

Keywords: Flow shop, Heuristic, Permutation, Makespan.

1. INTRODUCTION

Scheduling is the method of allocating resources (such as machines) to tasks (such as jobs) in order to ensure that these tasks are done in a reasonable period of time. Some strategies are used to pick the most suitable work. The main task is to determine the best approach for running jobs on various machines or processors. The number of processors or jobs can vary from one to the next. Scheduling specifies the best order for n jobs to be processed on m machines in the same order, i.e. each job must be processed in the same order on machines 1,2,...,m. Scheduling can be broadly classified into three categories namely, flow shop scheduling problem, job shop scheduling problem and open shop scheduling problem. In flow shop scheduling problem, which is a pure sequencing problem. Here all the jobs must be processed using the same sequence and the objective must be makespan minimization or tardiness or lateness or weighted tardiness, weighted lateness and due date related criteria has been chosen. In the area of job shop scheduling problem jobs can be processed through routing and scheduling. Here the first job may follow some route, second job may follow some route, etc., Here the objective is to find appropriate route-based scheduling algorithm must be developed. It must be either heuristic based or branch and bound based algorithm is developed for processing. In the area of open shop scheduling jobs can be processed in any conceivable order. Here there is no predetermined routing or sequencing. In this type of problems, the objective must be minimizing the makespan, minimizing the total completion time, etc., Basically thes type of problems are NP-hard in nature. So complete enumeration is not possible. Therefore, we choose heuristic approach to solve this type of problems. Finding the objective for a particular problem is based on the nature of the problem and the need of the problem. In this paper, we have developed an algorithm for finding the minimum makespan in the flow shop scheduling problems. The flow shop scheduling problem is a production problem in which a number of n jobs must be processed on m machines in the same way. The permutation flow shop sequencing output environment exists when the sequence of jobs processing on all machines is the same. At time zero, a flow shop has M machines in series and N separate jobs available for processing.

2. LITERATURE REVIEW

Johnson began by presenting an algorithm for determining the best sequence for an n-job, two-machine problem. Palmer achieved the necessary result of minimising the make span using a single iteration process. The jobs were sequenced based on the estimation of the slope index, which was then sorted in decreasing order to obtain the best sequence using Palmer's process. This method can be used to solve scheduling issues in m-machine and n-job flow shops. Campbell et.al. later generalised the basic Johnson's algorithm. It uses a number of iterations before arriving at the final result, unlike the updated Johnson's algorithms. This approach is a heuristic that is generally referred to as the Campbell, Dudek, and Smith heuristics (CDS). There are several different sequences proposed here, with the ideal sequence having the shortest makespan. And Johnson's law was followed in every iteration. Gupta proposed a different algorithm for the procedure of minimising the sequence's completion time. He proposes a new method for calculating the slope index, which is then used to sequence jobs in the flow shop environment. Nawaz et.al. created a model that works by measuring the overall processing time of individual workers. The idea that the worker with the longest processing time has the greatest lead over the others is used to prioritise the jobs in the schedule. He took this idea and proposed the NEH heuristics algorithm, which is a heuristic for minimising makespan. Jayakumar, Shanthi, and Meganathan used a heuristic approach to solve the permutation flow shop scheduling problem. They also looked at the same n work m machine case and compared their algorithm to a previously published algorithm. They came to the conclusion that their algorithm outperformed the current algorithm in the literature. Jayakumar, Meganathan, and Shanthi used a heuristic approach to solve the two-machine n job flow shop problem. The results were compared to Johnson's method, and it was discovered that their algorithm is superior to Johnson's algorithm. Jayakumar and Vasudevan developed an algorithm for permutation flow shop scheduling with the aim of minimising the makespan.

They concluded that their modified algorithm not only minimises the makespan but also decreases resource idle time.

3. ASSUMPTION FOR SEQUENCING PROBLEMS

No machine can perform more than one operation at a time.

Once the operation is started it must be finished without any interruption (i.e.) pre-emption is not allowed.

Time intervals for processing jobs on various machines are unaffected by the order in which they are to be finished.

There should be a one-to-one communication between the tasks to be completed and the resources available at any given time.

Before the time span under review starts, all jobs are identified and ready to be processed.

Each jobs processing time on each machine is known and does not change.

The time spent switching jobs between machines is zero.

It is not permitted to transfer jobs from one machine to another (i.e., jobs are processed in the same order on each machine.) If each job is managed separately on each machine. If each of the n jobs is to be processed by two machines A and B in the order A-B, then each job will be sent to machine A first, followed by machine B. The No Passing Rule is the name for this rule.

There is no parallel processing.

Each job must follow the same sequence of processing on each machine.

Since it is pure sequencing routing is not possible for processing.

4. PROCESSING N JOB THROUGH 2 MACHINES

4.1 Algorithm

Step 4.1.1: List the jobs along with their processing times on each machine in a table as shown below

Processing time	J_1	J_2	J_3	…	J_n
M_1	P_{11}	P_{12}	P_{13}	…	P_{1n}
M_2	P_{21}	P_{22}	P_{23}	…	P_{2n}

Step 4.1.2: Examine the processing times on machines M_1 and M_2 in each column in the table and find the minimum processing time and subtract it from all the processing time. Now we get processing time 'zero' in either M_1 or M_2.

Step 4.1.3: If the processing time 'zero' is one of M_1 then place the job in the first available position in the sequence. If the processing time 'zero' is on machine M_2, then place it at the last available position in the sequence.

Step 4.1.4: Suppose, if we get more than one zero, then the following three situations may arise,

If we get zero on both the machines, i.e.) $P_{1k} = P_{2r} = 0$, then process the k^{th} job first in the sequence and the r^{th} job last in the available sequence.

Suppose, if we get more than one zero among M_1, then select any one of these jobs randomly and put it first and then put another job second in the sequence.

Suppose, if we get more than one zero among M_2, then select any one of these jobs randomly and put it last and then put another job second from the last in the sequence.

Step 4.1.5: Cross off the assigned jobs from the table. If no job remains to be assigned, then stop the procedure and go to 4.1.6 otherwise go to step 4.1.2.

Step 4.1.6: Then using this sequence, we find the total elapsed time by using Johnson's method.

4.2. Example For 'N' Job '2' Machine Problem

Table 4.2.1

Job / Machine	1	2	3	4	5
A	8	6	9	7	10
B	4	9	8	6	5

Choose the minimum processing time in the given table. In the above table the minimum processing time is 4. So, subtract 4 from all the processing time and form the new table.

Table 4.2.2

Job / Machine	1	2	3	4	5
A	4	2	5	3	6
B	0	5	4	2	1

Here we get zero in the first job in last machine. Hence, we put the first job last in the sequence.

				1

Then remove the first job in the given table and choose the minimum number in the reduced table and subtract all the other processing time from it and form the new table.

Table 4.2.3

Job / Machine	2	3	4	5
A	2	5	3	6
B	5	4	2	1

Here, the minimum processing time is 1. So, subtract 1 from all the processing time and form the new table.

Table 4.2.4

Job / Machine	2	3	4	5
A	1	4	2	5
B	4	3	1	0

In the above table, we get processing time 'zero' in the fifth job in the last machine and so, put the fifth job from the last in the sequence.

			5	1

Now remove the fifth job in the above table and form the new table.

Table 4.2.5

Job / Machine	2	3	4
A	1	4	2
B	4	3	1

In the reduced table, the minimum processing time is 1 and so subtract one from all the processing time and form the new table.

Table 4.2.6

Job / Machine	2	3	4
A	0	3	1
B	3	2	0

In the above table we get processing time zero in two jobs. One is on the second job in the first machine and on the fourth job in the last machine. So, put the second job first in the sequence and fourth job from the last on the sequence.

2		4	5	1

Now put the left-out job 3 at the remaining place in the sequence. Thus, the optimal sequence is

2	3	4	5	1

Now calculate the total elapsed time

Table 4.2.7

Machines / Jobs	M_1		M_2	
	In	Out	In	Out
2	0	0+6=6	6	6+9=15
3	6	6+9=15	15	15+8=23
4	15	15+7=22	23	23+6=29
5	22	22+10=32	32	32+5=37
1	32	32+8=40	40	40+4=44

Idle time for M_1=4 hours

Idle time for M_2 = 12hours

5. PROCESSING N JOB THROUGH THREE MACHINES

5.1 Algorithm

Step 5.1.1: List the jobs along with their processing times on each machine in a table as shown below

Processing time	J_1	J_2	J_3	…	J_n
M_1	P_{11}	P_{12}	P_{13}	…	P_{1n}
M_2	P_{21}	P_{22}	P_{23}	…	P_{2n}
M_3	P_{31}	P_{32}	P_{33}	…	P_{3n}

Step 5.1.2: Examine the processing times on machines M_1, M_2 and M_3 in each column in the table and find the minimum processing time and subtract it from all the processing time. Now we get processing time 'zero' in at least one of the machines.

Step 5.1.3: Now check whether which machine contains the zero-processing time. Suppose, if the processing time zero is in the first machine, then put the corresponding job first in the sequence. Suppose, the zero-processing time in the last machine, then put the corresponding job last in the sequence and suppose if it is on the centre machine, cross off that machine without making any changes.

Step 5.1.4: Now, the problem is converted into two machine problem, so proceed the procedure given in the two-machine problem and find the optimal sequence.

Step 5.1.5: By using the obtained sequence, calculate the total elapsed time.

5.2 Example For 'N' Job '3' Machine Problem

Table 5.2.1

Job / Machine	1	2	3	4	5
A	5	7	6	9	5
B	2	1	4	5	3
C	3	7	5	6	7

Here the minimum processing time is 1. So, subtract one from all the processing time given in the table.

Table 5.2.2

Job / Machine	1	2	3	4	5
A	4	6	5	8	4
B	1	0	3	4	2
C	2	6	4	5	6

In the above table, the processing time 'zero' is on the centre machine. Hence cancel the second machine without making any changes.

Table 5.2.3

Job / Machine	1	2	3	4	5
A	4	6	5	8	4
C	2	6	4	5	6

Here the minimum processing time is 2. So, subtract 2 from all the processing time and form the new table.

Table 5.2.4

Job / Machine	1	2	3	4	5
A	2	4	3	6	2
C	0	4	2	3	4

Now, we get zero on the first job in last machine. So put the job 1 is last in the sequence.

				1

Now cancel the first job in the table. Choose the minimum processing time in the remaining job and repeat the same procedure.

Table 5.2.5

Job / Machine	2	3	4	5
A	2	1	4	0
C	2	0	1	2

Here we get two zeros. One is on the third job in the last machine and the another one is one the fifth job in first machine. So, put the third job from last in the sequence and put fifth job first in the sequence.

5			3	1

Now cancel the third and fifth job and in the reduced table choose the minimum processing time and repeat the same procedure.

Table 5.2.6

Job / Machine	2	4
A	1	3
C	1	0

Here we get zero on the fourth machine in the last job. So put the job 4 from last in the sequence.

5		4	3	1

Now put the left-out job 2 at the remaining place in the sequence. Thus, the optimal sequence is

5	2	4	3	1

Now calculate the total elapsed time

Table 5.2.7

Jobs	M_1		M_2		M_3	
	In	Out	In	Out	In	Out
5	0	0+5=5	5	5+3=8	8	8+7=15
2	5	5+7=12	12	12+1=13	15	15+7=22
4	12	12+9=21	21	21+5=26	26	26+6=32
3	21	21+6=27	27	27+4=31	32	32+5=37
1	27	27+5=32	32	32+2=34	37	37+3=40

Idle time for M_1=8 hours

Idle time for M_2 = 25 hours

Idle time for M_3 = 12 hours

6. PROCESSING N JOB THROUGH M MACHINES

6.1. Algorithm:

Step 6.1.1: Select the minimum processing time in the given table.

Step 6.1.2: Subtract the minimum processing time from all the processing time given in the table. Now we get at least one of the processing time is zero.

Step 6.1.3: Now check whether which machine contains the zero-processing time. Suppose, if the processing time zero is in the first machine, then put the corresponding job first in the sequence. Suppose, the zero-processing time in the last machine, then put the corresponding job last in the sequence and suppose if it is not on the first or last machine, cross off that corresponding machine without making any changes.

Step 6.1.4: Now, cancel the job, which is placed in the sequence. Then form the reduced table with the remaining processing time. Then go to 6.1.1 and repeat the same procedure until all the jobs get sequenced.

Step 6.1.5: By using the obtained sequence, calculate the total elapsed time.

6.2 Example For 'N' Job 'M' Machine Problem

Table 6.2.1

Jobs / Machines	J_1	J_2	J_3	J_4	J_5
M_1	7	9	11	8	10
M_2	6	5	4	5	6
M_3	5	4	6	8	4
M_4	9	8	12	10	7

In the above table, choose the minimum processing time and subtract the minimum processing time from all the processing time. Here the minimum processing time is 4. So, subtract 4 from all the processing time given in the table and form the new table.

Table 6.2.2

Jobs / Machine	J_1	J_2	J_3	J_4	J_5
M_1	3	5	7	4	6
M_2	2	1	0	1	2
M_3	1	0	2	4	0
M_4	5	4	8	6	3

Here we get zero in the second and third machine. So, cancel the second and third machine without make any changes and form the new table.

Table 6.2.3

Jobs / Machines	J_1	J_2	J_3	J_4	J_5
M_1	3	5	7	4	6
M4	5	4	8	6	3

In the above table the minimum processing time is 3. So, subtract 3 from all the processing time and form the new table.

Table 6.2.4

Jobs / Machines	J_1	J_2	J_3	J_4	J_5
M_1	0	2	4	1	3
M4	2	1	5	3	0

Here we get zero on job 1 in the first machine and also on job 5 in the last machine. So, put J_1 first in the sequence and J_5 is last in the sequence. Therefore, the sequence is

J_1				J_5

Now form the new table, by cancelling first and fifth job. Hence, we get

Table 6.2.5

Jobs / Machines	J_2	J_3	J_4
M_1	2	4	1
M4	1	5	3

Here, in this table the minimum processing time is 1, so subtract 1 from all the processing time and form the new table.

Table 6.2.6

Jobs Machines	J_2	J_3	J_4
M_1	1	3	0
M4	0	4	2

Here we get zero on second job in the last machine and fourth job in the first machine. Therefore, put J_2 from the last in the sequence and J_4 from first in the sequence.

Hence, the sequence is

J_1	J_4		J_2	J_5

Now put the left-out job J_3 at the remaining place in the sequence. Thus, the optimal sequence is

J_1	J_4	J_3	J_2	J_5

Now calculate the total elapsed time

Table 6.2.7

Jobs	M_1		M_2		M_3		M_4	
	In	Out	In	Out	In	Out	In	Out
J_1	0	0+7=7	7	7+6=13	13	13+5=18	18	18+9=27
J_4	7	7+8=15	15	15+5=20	20	20+8=28	28	28+10=38
J_3	15	15+11=26	26	26+4=30	30	30+6=36	38	38+12=50
J_2	26	26+9=35	35	35+5=40	40	40+4=44	50	50+8=58
J_5	35	35+10=45	45	45+6=51	51	51+4=55	58	58+7=65

Idle time for M_1 = 20 hours

Idle time for M_2 = 39 hours

Idle time for M_3 = 38 hours

Idle time for M_4 = 19 hours

6.3 Result:

Table 6.3.1

No. of Observation	Technique	Total elapsed time
1.	Our algorithm	65
2.	Palmer method	67
3.	CDS method	65
4.	NEH method	65
5.	Gupta method	67
6.	JV algorithm	67

7. CONCLUSION

In this paper, we developed a procedure to obtain the minimum total elapsed time. By the obtained result, we conclude that our algorithm gives the better result when compared with Palmer method, JV method and Gupta method and also our algorithm gives the same result when compared with CDS and NEH methods even though the procedure are lengthy when compare to our algorithm which is most efficient and simple.

REFERENCES

1. *Campell H G, Dudek R.A. and Smith M.L., "A heuristic algorithm for n-jobs m-machines sequencing problems", Management Science, Vol. 16B, (1970), pp.166-177.*
2. *Gupta, J. N. D., "A functional heuristic algorithm for the flow shop scheduling problem", 22, (1971), pp.39-47.*
3. *Jayakumar S, Meganathan R and Sathiya Shanthi R, "An Heuristic Approach for Solving 2-machine n-jobs Flow Shop Scheduling Problem with makespan objective", IOSR Journal of Mathematics, 12(3), (2016), pp.23-26.*
4. *Jayakumar S, Sathiya Shanthi R and Meganathan R, "An Heuristic Approach for Solving Permutation Flow Shop Scheduling Problem", International Journal of Emerging Technology and Advanced Engineering, 6(4), (2016), pp. 144-146.*
5. *Jayakumar and Vasudevan, "Heuristic algorithm for permutation flow shop scheduling", International Journal of Technology, 9(1), (2019), pp.5-7.*
6. *Johnson, S. M., "Optimal two- and three-stage production schedules with setup times included". Naval Research Logistics, 1, (1954), pp. 61–68.*
7. *Nawaz M, Enscore Jr.E. and Ham I, "A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problems", OMEGA, The International Journal of Management Science 11, no.1, (2009), pp. 1-10.*
8. *Palmer D S "Sequencing jobs through a multi stage process in the minimum total time – A quick method for obtaining a near optimum", Operations Research, 16 (1956), pp. 101-107.*

AUTHORS

Dr. S. Jayakumar, HOD, PG and Department of Mathematics, AAGA College, Arcot Road, Cheyyar – 604 407, (Tamil Nadu) India.
Email: sundarajayakumar@gmail.com

S. Jayasankari. Research Scholar, PG and Dept. of Mathematics, AAGA College, Arcot Rd, Cheyyar – 604 407, (Tamil Nadu) India.
Email: jayasankarikumari03@gmail.com