A New Algorithm for Solving Job Shop Sequencing Problem

Srikant Gupta, Irfan Ali and A. Ahmed
Department of Statistics & Operations Research, Aligarh Muslim University, Aligarh (India)
Email: srikantgupta188@gmail.com, irfii.ali@gmail.com, aquilstat@gmail.com

Abstract
In this paper we have proposed SAI-Method for solving sequencing problems. The procedure adopted for solving the wide range of sequencing problems is easiest and involves the minimum numbers of iterations to obtain the sequence of jobs. Also, we have compared our proposed method with the Johnson's method.The steps used for obtaining the job sequence by SAI method in comparison to the Johnson's method are very simple and based on arithmetic reasoning. Also, we have solved several numerical examples to show that the solutions obtained by SAI method are consistent and efficient.

KEYWORDS: Sequencing Problem, Johnson's Method, SAI Method

I. Introduction

Sequencing problem is considered to be one of the classic and important applicationsof operations research. The main role of the classical sequencing problem is to find the optimal sequence of the jobs on machines so as to minimize the total amount of time required to complete the process of all the jobs. The simplest pure sequencing problem is one in which there is a single resource, or machine, and all processing times are deterministic.The goal of the sequencing problem consists of determining the order or sequence in which the machines will process the jobs so as to optimize some measure of performance (i.e. cost, time or mileage, weight etc.) to complete the process. The effectiveness of the sequencing problem can be measured in terms of minimized costs, maximized profits, minimized elapsed time and meeting due dates etc. In the past, because of its practical and significant use in production field many researchers have shown their interest in sequencing problems. One of the renowned work in the field of sequencing considered till date is by Johnson's, who gave the algorithm in 1954 for production scheduling in which he had minimized the total idle time of machines and the total production times of the jobs. Later in 1967 Smith and Dudek developed a general algorithm for the solution of the n- job on m- machine sequencing problem of the flow shop when no passing is allowed. Similarly, Maggu and Das (1977), Maggu (2002), Rao *et.al.* (2013) and many others gave the technique to minimize the total ideal time of machines or the total production time of the jobs on the two machines production scheduling problems. A heuristic algorithm for solving general sequencing or flow shop scheduling problem was given by Nawaz *et.al.* (1983), Johnny and Chang (1991), Koulamas (1998), Laha and Chakraborty (2009) for minimizing elapsed time in no-wait flow-shop scheduling. Cai *et.al.*(1997) in their work have concerned the problem of scheduling n jobs with a common due date on a single machine so as to minimize the total cost arising from earliness and tardiness. Baker (2002) considers complete enumeration, integer programming, branch and bound techniques to obtain the optimal sequences, but he does not provide efficient solutions for the large size problems. While Kalczynski and Kamburowski (2006) dealt with the classical problem of minimizing the makespan in a two-machine flow shop with deterministic job processing times, the optimal job sequence was determined by applying Johnson's rule. Recently, Ahmad and Khan (2015) gave an algorithm for the constrained flow-shop scheduling problem in which they considered the transportation time, weight of jobs and break down time with$m$-machines to obtained an optimal or near optimal solution.

In this paper, we proposed a procedure for solving the job scheduling problems named as SAI Method. This method is used to frame a sequence of jobs for processing the n jobson m machines in such a way that the total elapsed time is minimized. Several numerical examples are given for better understanding of the solution procedure of SAI method.

II. Preliminaries

In production scheduling, when we have m number of facility or machine and n number of jobs or tasks, then we have $(n!)^m$ of possible sequence but the most optimal sequence is one which minimizes the idle time or elapsed time (i.e. the time from the start of the 1st job to the completion of last job) by satisfying the order in which each job must be performed through m machines one at a time. Sequencing problems are concerned with an appropriate selection of a sequence of jobs to be given a finite number of service facilities.

Sequencing problem constitutes of various terminology:

I. Processing order means the order in which various machines are required for completing the job.
II. Processing time means the time each job on each machine.
III. Idle time on a machine is the time for which machine remains idle (not working) during the total elapsed time.
IV. Total elapsed time is the time between starting the first job and completing the last job. This also includes the idle time.
V. No passing rule means the passing is not allowed if each of the n job is to be processed through two machines say M_1 and M_2 in order M_1M_2 then the rule means that each job will go to machine M_1 first & then to machine M_2.

Following are the assumption related to sequencing problem:

I. Only one operation is carried out on a machine at a time.
II. Processing times are known and do not change.
III. The processing times of machines are independent of the order of processing the jobs.
IV. The time involved in moving jobs from one machine to another is negligible.
V. Each operation once started, must be completed.
VI. A job is processed as soon as possible but only in the order specified.

III. SAI - Method

The step-wise iterative procedure of SAI method for determining the optimum sequence for n jobs $(1,2,\ldots\ldots,n)$on m machines $(1,2,\ldots\ldots,m)$ is as follows:

Step 1: The processing time of n jobs $(1,2,\ldots\ldots,n)$on m machines $(1,2,\ldots\ldots,m)$ is given in table 1:

Table 1: Processing of n jobs in m machines

Machines \ Jobs	**1**	**2**	**3**	**…**	**k**	**…**	**n**
1	t_{11}	t_{12}	t_{13}	…	t_{1k}	…	t_{1n}
2	t_{21}	t_{22}	t_{23}	…	t_{2k}	…	t_{2n}
3	t_{31}	t_{32}	t_{33}	…	t_{3k}	…	t_{3n}
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
i	t_{i1}	t_{i2}	t_{i3}		t_{ik}		t_{in}
:	:	:	:		:	:	:
:	:	:	:		:	:	:
m	t_{m1}	t_{m2}	t_{m3}	…	t_{mk}	…	t_{mn}

Step 2: Examine the jobs and select the least job processing time among all n jobs $(k = 1,2,3\ldots\ldots,n)$ for each machine and then marked it with $(-)$ sign. Let the minimum processing time occurred at k^{th} job on i^{th} machine. Mathematically; we can say

$$\underset{k}{Min}\ \{t_{i1}, t_{i2}, \ldots, t_{ik}, \ldots, t_{in}\} = t_{ik}\ .$$

Step 3: Similarly, select the least processing time among all m machines $(i = 1,2,3\ldots\ldots,m)$ for each job and then marked it with $(+)$ sign. Let the minimum processing time is occurred at i^{th} machine for the k^{th} job. Mathematically; we can say

$$\underset{i}{Min}\ \{t_{1k}, t_{2k}, \ldots, t_{ik}, \ldots, t_{mk}\} = t_{ik}\ .$$

Step 4: Again examine the rows and columns of table, select the cell with $(\mp)$ sign. Let the $(\mp)$ has occurred at cell which corresponds to i^{th} machine and k^{th} job. The k^{th} job is excluded from the table and is placed in the optimal job sequence.

Step 5: Step 1to 4 are repeated until all the jobs are placed in the optimal job sequence.

There may be a situation where a tie has occurred -

i) If $(\mp)$ occurs at more than one place, then the job with least processing time is selected and is placed in the optimal job sequence.

ii) If $(\mp)$ occurs at more than one place and the processing time for the allocated jobs is same. Then the job which will process on the lower order positional machine is selected that is by ignoring the other higher order of machines.

Step 7: Lastly, we calculate the ideal time and total elapsed time of machines.

V. Numerical Examples

Example 1: There are 5 jobs, each of which must go through the two machines A and B in the order AB. Processing time are given below in table 2.

Table 2: Processing of 5 jobs in 2 machines

Job → Machine ↓	1	2	3	4	5
A	7	5	3	8	2
B	11	8	12	5	1

By applying SAI Method on table 2, we proceed as follows

Table 3: Processing of 5 jobs in 2 machines

Job → Machine ↓	1	2	3	4	5
A	7 +	5 +	3 +	8	2 −
B	11	8	12	5 +	1 $\mp$

The shortest Processing time is 1 for Job 5. So perform Job 5 in the beginning of the sequence

5				

After deletion of job 5 from table 3, the revised table 4 is given below.

Table 4: Processing of 4 jobs in 2 machines

Job → Machine ↓	1	2	3	4
A	7 +	5 +	3 $\mp$	8
B	11	8	12	5 $\mp$

Again the shortest processing time is 3 for Job 3. So perform the Job 3 after the Job 5 in the sequence

5	3			

Continuing in the same manner similarly table 4 reduces to table 5.

Table 5: Processing of 3 jobs in 2 machines

Job → Machine ↓	1	2	4
A	7 +	5 $\mp$	8
B	11	8	5 $\mp$

The shortest Processing time is 5 for Job 2 or Job 4. We will consider the job 2 for sequence

5	3	2		

After deletion of job 2 from table 5, the revised table 6 for the remaining jobs is given below.

Table 6: Processing of 2 jobs in 2 machines

Job → Machine ↓	1	4
A	7 $\mp$	8
B	11	5 $\mp$

The shortest processing time is 5 for job 4. Leading to the optimal sequence as

5	3	2	4	1

Flow of jobs through machines A and B using the optimal sequence

$$\mathbf{5 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1}$$

The minimum total elapsed time is calculated for the obtained sequence in table 7 as follows

Table 7: Computation of Total Elapsed Time for the job sequence

Sequence	Machine A		Machine B	
	Time In	**Time Out**	**Time In**	**Time Out**
5	0	2	2	3
3	2	5	5	13
2	5	10	13	21
4	10	18	21	26
1	18	25	26	37

Comparison between SAI and Johnson's Method

By using SAI Method, we have
Total elapsed time= 37 hours
Idle Time for Machine A= 12 hours
Idle Time for Machine B= 04 hours

By using Johnson's Algorithm, we have
Total elapsed time=40 hours
Idle Time for Machine A= 15 hours
Idle Time for Machine B= 03 hours

Example 2: There are 5 jobs, each of which must go through the three machines A, B and C in the order ABC. Processing time is given below.

Table 8: Processing of 5 jobs in 3 machines

Job → Machine ↓	1	2	3	4	5
A	3	8	7	5	2
B	3	4	2	1	5
C	5	8	10	7	6

By applying SAI Method on table 8, we obtain flow of jobs through machines A, B and C in the sequence

$$\mathbf{4 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 2}$$

Comparison between SAI and Johnson's Method

By using SAI Method, we have
Total elapsed time= 40 hours
Idle Time for Machine A= 15 hours
Idle Time for Machine B= 27 hours
Idle Time for Machine C= 06 hours

By using Johnson's Algorithm, we have
Total elapsed time= 42 hours
Idle Time for Machine A= 17 hours
Idle Time for Machine B= 27 hours
Idle Time for Machine C= 06 hours

Example 3: There are 5 jobs, each of which must go through the four machines A, B, C and D in the order ABCD. Processing time is given below

Table 9: Processing of 5 jobs in 4 machines

Machine → Job ↓	A	B	C	D
1	7	15	14	21
2	11	18	18	6
3	2	13	11	16
4	14	4	27	14
5	18	11	32	16

By applying SAI Method on table 9, we obtain flow of jobs through machines A, B, C and D in the sequence

$$3 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 5$$

Comparison between SAI and Johnson's Method

By using SAI Method, we have
Total elapsed time= 125 hours
Idle Time for Machine A= 73 hours
Idle Time for Machine B= 64 hours
Idle Time for Machine C= 46 hours
Idle Time for Machine D= 52 hours

By using Johnson's Algorithm, we have
Total elapsed time= 127 hours
Idle Time for Machine A= 65 hours
Idle Time for Machine B= 68 hours
Idle Time for Machine C= 46 hours
Idle Time for Machine D= 56 hours

Example 4: There are 4 jobs, each of which must go through the four machines A, B, C and Din the order ABCD. Processing time is given below

Table 10: Processing of 4 jobs in 4 machines

Machine → / Job ↓	A	B	C	D
1	20	10	9	20
2	17	7	15	17
3	21	8	10	21
4	25	5	9	25

By applying SAI Method on table 10, we obtain flow of jobs through machines A, B, C and D in the sequence

$$4 \rightarrow 2 \rightarrow 3 \rightarrow 1$$

Comparison between SAI and Johnson's Method

By using SAI Method, we have
Total elapsed time= 122 hours
Idle Time for Machine A= 39 hours
Idle Time for Machine B= 92 hours
Idle Time for Machine C= 79 hours
Idle Time for Machine D= 39 hours

By using Johnson's Algorithm, we have
Total elapsed time= 122 hours
Idle Time for Machine A= 65 hours
Idle Time for Machine B= 92 hours
Idle Time for Machine C= 79 hours
Idle Time for Machine D= 39 hours

Example 5: There are 4 jobs, each of which must go through the five machines A, B, C, D and E in the order ABCDE. Processing time is given below

Table 11: Processing of 4 jobs in 5 machines

Job → / Machine ↓	1	2	3	4
A	6	5	4	7
B	4	5	3	2
C	1	3	4	2
D	2	4	5	1
E	8	9	7	5

By applying SAI Method on table 11, we obtain flow of jobs through machines A, B, C, D and E in the sequence

$$1 \rightarrow 4 \rightarrow 3 \rightarrow 2$$

Comparison between SAI and Johnson's Method

By using SAI Method, we have
Total elapsed time= 43 hours
Idle Time for Machine A= 21 hours
Idle Time for Machine B= 29 hours
Idle Time for Machine C= 30 hours
Idle Time for Machine D= 31 hours
Idle Time for Machine E= 22 hours

By using Johnson's Algorithm, we have
Total elapsed time= 43 hours
Idle Time for Machine A= 21 hours
Idle Time for Machine B= 29 hours
Idle Time for Machine C= 33 hours
Idle Time for Machine D= 31 hours
Idle Time for Machine E= 18 hours

Example 6: There are 4 jobs, each of which must go through the six machines A, B, C, D, E and F in the order ABCDEF. Processing time is given below

Table 12: Processing of 6 jobs in 4 machines

Machine → Job ↓	A	B	C	D	E	F
1	15	8	6	14	6	26
2	17	7	9	10	15	22
3	21	7	12	9	11	19
4	18	6	11	12	14	17

By applying SAI Method on table 12, we obtain flow of jobs through machines A, B, C, D, E and F in the sequence

1 → 4 → 2 → 3

Comparison between SAI and Johnson's Method

By using SAI Method, we have
Total elapsed time= 133 hours
Idle Time for Machine A= 62 hours
Idle Time for Machine B= 106 hours
Idle Time for Machine C= 95 hours
Idle Time for Machine D= 88 hours
Idle Time for Machine E= 87 hours
Idle Time for Machine F= 50 hours

By using Johnson's Algorithm, we have
Total elapsed time= 133 hours
Idle Time for Machine A= 62 hours
Idle Time for Machine B= 105 hours
Idle Time for Machine C= 95 hours
Idle Time for Machine D= 88 hours
Idle Time for Machine E= 87 hours
Idle Time for Machine F= 49 hours

Example 7:

Table 13: Processing of 5 jobs in 2 machines

Job	1	2	3	4	5
Machine A	10	2	18	6	20
Machine B	4	12	14	16	8

By applying SAI method on table 13, we compare the result with Johnson's Method

By using SAI Method, we have
Sequence: 2 → 1 → 4 → 5 → 3
Elapsed time: 60 hours

By using Johnson's Algorithm, we have
Sequence: 2 → 4 → 3 → 5 → 1
Elapsed time: 60 hours

Example 8:

Table 14: Processing of 5 jobs in 3 machines

Job	1	2	3	4	5
Machine A	1	2	3	4	5
Machine B	8	12	5	7	11
Machine C	15	12	12	14	17

By applying SAI method on table 14, we compare the result with Johnson's Method

By using SAI Method, we have
Sequence: 1 → 2 → 3 → 4 → 5
Elapsed time: 78 hours

By using Johnson's Algorithm, we have
Sequence: 3 → 1 → 4 → 2 → 5
Elapsed time: 78 hours

Example 9:

Table 15: Processing of 5 jobs in 3 machines

Job	1	2	3	4	5
Machine A	10	12	8	9	13
Machine B	7	8	4	5	6
Machine C	6	11	10	8	7

By applying SAI method on table 15, we compare the result with Johnson's Method

By using SAI Method, we have
Sequence: 3 → 4 → 5 → 1 → 2
Elapsed time: 65 hours

By using Johnson's Algorithm, we have
Sequence: 3 → 2 → 4 → 1 → 5
Elapsed time: 65 hours

Example 10:

Table 16: Processing of 5 jobs in 3 machines

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

By applying SAI method on table 16, we compare the result with Johnson's Method

By using SAI Method, we have
Sequence: 4 → 3 → 5 → 1 → 2
Elapsed time: 42 hours

By using Johnson's Algorithm, we have
Sequence: 1 → 4 → 5 → 3 → 2
Elapsed time: 42 hours

VI. Conclusion

In this paper, we have proposed an algorithm called SAI Method for providing job sequence of *n* jobs on *m* machines in minimum number of iterations. This method has the following advantages:

- The solution obtained by this method is either optimal or nearest to the optimal solution.
- The computational steps performed to obtain the optimal solution are very easy.
- The final optimal solution is obtained in a short time.
- This method can be easily applied on small number of jobs as well as complex jobs scheduling problems.
- This method reduces the complexity faced at the time of solving the job sequencing problem.
- It is simple to use and thus anyone can adopt it easily.
- In case of *n* jobs and *m* machines, Johnson's rue is not applicable unless the problem is converted into 2 jobs and *m* machines where as the proposed method can be easily applied directly on *n* jobs and *m* machines.

Thus it can be concluded that the SAI method will be helpful for a layman who is dealing with production job scheduling problems.

Acknowledgement

The second author is thankful to University Grant Commission (UGC) to provide the financial assistance under the UGC start-up grant No. **F.30-90/2015 (BSR)**, Delhi India to carry out this research work.

Conflict of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] Johnson, S. M. (1954): "Optimal Two Stage and Three Stage Production Schedules with Setup Times Included", Naval Logistics Quarterly, vol. 1, no. 1, pp. 61–68.

[2] Smith, R. D., & Dudek, R. A.(1967): "A General Algorithm for Solution of the n-Job, M-Machine Sequencing Problem of the Flow Shop",Operations Research, Vol. 15, pp 71-82.

[3] Maggu & Das (1977): "Equivalent jobs for job blocks in job sequencing", PAMS, Vol. XIV, No. 4, pp. 271-281.

[4] Nawaz M. *et.al.* (1983): "A heuristic algorithm for the n-job, m-machine flowshop sequencing problem", Omega, 11, 91-95.

[5] Ho J. C. and Chang Y. L.(1991): "A new heuristic for the n-job, m-machine flow-shop problem", European Journal of Operational Research, 52, 194-202.

[6] Cai. X., *et.al.* (1997): "Scheduling about a common due date with job dependent asymmetric earliness and tardiness penalties", European Journal of Operational Research, Vol. 98, pp. 154 – 168.

[7] Koulamas, C. (1998):"A new constructive heuristic for the flowshop scheduling problem", European Journal of Operational Research, 105, 66-71.

[8] Pinedo, M., and Chao, X. (1998): "Operations Scheduling with Applications in Manufacturing and Services", Boston: McGraw Hill/Irwin.

[9] Baker, K. R. (2002): "Elements of Sequencing and Scheduling. Hanover", NH: Baker Press**.**

[10] Pinedo, M.(2002): "Scheduling: Theory, Algorithms, and Systems", 2nd ed. Upper Saddle River, NJ: Prentice Hall.

[11] Maggu, A.(2002):"On two machines production scheduling problem to minimize total expected time of job's or to minimize total expected idle times of machines", PAMS, Vol. LV, No.1-2, pp. 65-67.

[12] Srinivas. J., *et.al.* (2004): "An optimal sequencing approach for job – shop production", Journal of Scientific and Industrial Research, Vol.63, pp. 458 – 461.

[13] Kalczynski.P.J., Kamburowski. J. (2006): "A heuristic for minimizing the expected makespan in two machine flow shops with consistent co-efficient of variation", European Journal of Operational Research, Vol. 169, pp. 742 – 750.

[14] Laha, D., & Chakraborty, U. K. (2009): "A constructive heuristic for minimizing makespan in no-wait flow-shop scheduling", International Journal of Advanced Manufacturing Technology, Vol. 41, pp. 97-109.

[15] Rao, Rajuand, Ramesh (2013): "Modified heuristic time deviation technique for job sequencing and computation of minimum total elapsed time", International Journal of Computer Science & Information Technology (IJCSIT),Vol. 5, No 3.

[16] Ahmad, Q.S., and Khan, M.H.(2015): "Constrained Flow-Shop Scheduling Problem with m Machines".Journal of Multidisciplinary Engineering Science and Technology, Vol. 2, No.2.