

A New Heuristic Algorithm using Pascal's Triangle to determine more than one Sequence having Optimal/ near Optimal Make Span in Flow Shop Scheduling Problems

Baskar A
Research Scholar
VIT University
Vellore-632 014, India

Anthony Xavier M
Associate Professor
VIT University
Vellore-632 014, India

ABSTRACT

In this paper, an attempt is made to find a sequence having optimal or near optimal make span in a flow shop scheduling of 'n' jobs in 'm' machines using a newly proposed heuristic algorithm based on Pascal's Triangle (for nC_r). It is simple and can be easily coded in any high level language to run in a computer for effective and fast computation. Also, the effectiveness of the new Heuristic is analyzed using few case studies in comparison with some of the popular Heuristics like RA Heuristics, Palmer Heuristics, Gupta Heuristics, CDS Heuristics and Johnson's algorithm.

General Terms

Optimization of Make Span in General Flow Shop Scheduling Problems using Heuristics.

Keywords

Scheduling, Optimal sequence, Make Span, Heuristic, Pascal's Triangle.

1. INTRODUCTION

A flow shop scheduling problem (FSP) with m – machine permutation is considered in which, each job i , $i=1,2,\dots,n$ needs to be processed on each machine j , $j=1,2,\dots,m$, in that order during an uninterrupted processing time $t_{ij} \geq 0$. In such cases, minimizing the make span is considered as one of the most important performance measures that has to be optimized. Some common performance evaluation objective functions of a FSP are:

- Make Span – total time to completely process all jobs (Most Common)
- Average Time of jobs in shop
- Lateness
- Average Number of jobs in shop
- Utilization of machines
- Utilization of workers

Our objective is to find a processing order of the n jobs, the same for each machine, such that the make span is minimized. That is, the n jobs are finished as soon as possible. It is assumed that all jobs are available for processing at time zero. At any time, each machine can process at most one job and each job can be processed on at most one machine. The capacity of the queue for each machine is unlimited.

For n number of jobs, then, $n!$ Sequences are possible. The function grows exponentially with an increase in the problem

size. But, for a problem with 2 machines and n jobs, Johnson [1] had developed a polynomial algorithm to get an optimal sequence (more than one optimal solution may be available for the same problem), that is, in a definite time, one can get an optimal solution. Johnson's algorithm can be extended to a three machines and n jobs problems if any one of the following two conditions is satisfied:

$$\text{If } \min_i t_{i1} \geq \max_i t_{i2} \\ (\text{Or})$$

$$\text{If } \min_i t_{i3} \geq \max_i t_{i2}$$

If any one of the above conditions is satisfied, the 3 machines n jobs problem can be converted in to a 2 machine n jobs problem as shown in Table-1, and then the optimum make span is determined by using the data of the original problem.

Table – 1. Three machines converted to Two machines

Job	M/C I	M/C II
1	$t_{11} + t_{12}$	$t_{12} + t_{13}$
2	$t_{21} + t_{22}$	$t_{22} + t_{23}$
...
n	$t_{n1} + t_{n2}$	$t_{n2} + t_{n3}$

The job sequence is constructed in a forward direction while processing down the branching tree. For each node on the tree, a lower bound on the make span associated with the completion of the corresponding partial sequence σ is obtained by considering the work remaining on each machine. However, for large size problems, it would be difficult to get optimum solution in finite time, since the flow shop scheduling is a combinational problem. This means that the time complexity function for flow shop problem is exponential in nature. Hence, we have to use efficient Heuristics for large size problems. General notations and definitions pertaining to the Scheduling literature compiled by Conway et al. [2] are used through out this paper.

2. SOME CLASSICAL HEURISTICS

The basic consideration of any Heuristic shall be simple, optimum / near optimum solution in least possible time and minimum usage of any electronic processing equipment.

Many heuristics have been proposed to solve the flow shop problem. One can find the descriptions of these methods in the book by Baker. [3] Palmer [4] proposed the slope Index (SI) method. Gupta [5] proposed the functional algorithm. Both CDS and RA (rapid access procedure) heuristics proposed by Campbell et al. [6] and Dannenbring [7] respectively are based on Johnson's algorithm for the 2 machine problem. NEH algorithm proposed by Nawaz et al. [8] appears to be the best polynomial heuristics in practice. The heuristics RA or Palmer may also be useful when short computation times are required. That is, there has been a large amount of further researches directed at optimizing the make span of flow-shop instances of the permutation problem using constructive heuristics by researchers. The problem of minimizing sum of weighted mean flow time and intermediate storage cost in an open shop scheduling environment was addressed by Ho and Chang. [9] An improved iterated greedy algorithm (IGA) was proposed by Quan-Ke Pan et al. [10] to solve the no-wait flow shop scheduling problem with the objective to minimize the make span. A discrete firefly meta-heuristic with local search for make span minimization in permutation flow shop scheduling problems was presented by Mohammad Kazem Sayadia et al. [11] The main contribution of their work is to concern time-dependent weights which results in a more realistic insight for decision makers via considering time value of money in long scheduling problems.

Computational intelligence heuristics like simulated annealing , ant-colony optimization , differential evolution , particle swarm optimization , and fuzzy approaches have in recent years emerged as the most promising research directions for single-objective and multi-objective flow shop and job shop scheduling. An improved heuristic for permutation flow shop scheduling was proposed by Uday Kumar Chakraborty and Dipak Laha. [12] Dipak Laha and Uday Kumar Chakraborty proposed an efficient hybrid heuristic for make span minimization in permutation flow shop scheduling, and a new constructive heuristic, based on the principle of job insertion, for minimizing make span. [13,14] A constructive heuristic for the permutation flow shop scheduling problem with the objective of minimizing total completion time was presented by Dipak Laha and Arindam Chakravorty .[15]

While many procedures have been developed for optimizing a single performance criterion, a few researchers like Ravindran et al. , Allahverdi and Al-Anzi , Ming-Cheng Lo et al. have worked on Flow shop scheduling with multiple objective of minimizing more than one performance parameter like makes pan , total flow time, mean flow time, mean completion time.[16-18] Recently, Ant Colony Optimization technique for the sequence dependent flow shop Scheduling problem was analyzed by Mohammad Mirabi . [19]

As such, most of the Heuristics offer one optimal / near optimal solutions for general flow shop scheduling problems, CDS Heuristic offer a set of sequences which contain the optimal and a few more near optimum solutions with less effort. Except a few, most of the procedures are a bit complicated and requires some level of expertise to understand and implement. At the shop floor level, the requirement is a heuristic which is simple, fast, efficient, at the same time, should be able to give a sequence which is

having optimal or near optimal make span in most of the cases.

In this paper, we also propose a simple Heuristic based on the Pascal's Triangle for getting an optimal or near optimum sequence with less effort and analyzed with case studies presented by Panneerselvam.[20]

3. PASCAL'S TRIANGLE

It is the triangular representation of the combinational elements of nC_r , $r=0, 1, \dots, n$. Mathematically, $nC_r = n! / (r!(n-r)!)$. Typical triangle can be as shown below:

[illegible]

4. PROPOSED NEW HEURISTIC ALGORITHM BASED ON PASCAL'S TRIANGLE

Before stating the procedure, it will be appropriate to explain the situation which had resulted in this algorithm. While using the extended Johnson's algorithm, the 3 machines is reduced in to 2 machines. This is done by adding the processing times of first two machines of any job and assigning it to the first hypothetical machine and of last two machines to the second hypothetical machine as shown in Table-1 above.

When considering a problem with more than three machines, it can be shown that each Heuristic follows certain procedure. For example, RA Heuristic takes the sum of the products of the weights of a particular machine and the processing time of the corresponding machine of a particular job and assigning it to the first hypothetical machine. But, the weights will be in a decreasing order, that is, m for the first machine and 1 for the last machine. The weights will be in the increasing order for the second hypothetical machine, that is, 1 for the first machine and m for the last machine.

Gupta's Heuristic just adds the processing times of all machines except the last one and assigns it to the first hypothetical machine and adds the processing times of all machines except the first one and assigns it to the second hypothetical machine, for a particular job. CDS Heuristic corresponds to multistage use of Johnson's rule applied to a new problem formed from the original problem.

At stage 1,
 $t_{ii}^1 = t_{ij}$ and $t_{ij}^1 = t_{im}$

In other words, Johnson's rule is applied to the first and m th operations and intermediate operations are ignored.

At stage 2,

$$t_{j1}^2 = t_{j1} + t_{j2}, t_{j2}^2 = t_{jm} + t_{j,m-1}$$

That is, Johnson's rule is applied to the sum of the first two and the last two operation processing times. In general, at stage I,

$$t_{j1}^i = \sum_{k=1}^i t_{j,k} \text{ and } t_{j2}^i = \sum_{k=1}^i t_{j,m-k+1}$$

For each stage, $i (i=1, 2, \dots, m-1)$, the job order obtained is used to calculate a make span for the original problem. After $m-1$ stages, the best make span among the $m-1$ schedules is identified (some of the $m-1$ sequences may be identical).

When these Heuristics were studied, once a trial was made to reduce the m machine problem in to a two machine problem in a similar way which may be clearly explained as follows:

Let us consider, for simplicity, one job to be processed in 8 machines, from A to H with processing times a to h respectively.

	<u>Machine</u>							
	A	B	C	D	E	F	G	H
Job-I	a	b	c	d	e	f	g	h

Now, the machines are reduced to 7 with processing times as shown:

$$\begin{array}{ccccccc} a+b & b+c & c+d & d+e & e+f & f+g & g+h \\ & & & \text{and to 6 machines,} & & & \\ a+2b+c & b+2c+d & c+2d+e & d+2e+f & e+2f+g & f+2g+h & \\ & & & \text{and to 5 machines,} & & & \\ a+3b+3c+d & b+3c+3d+e & c+3d+3e+f & d+3e+3f+g & e+3f+3g+h & & \\ & & & \text{and to 4 machines,} & & & \\ a+4b+6c+4d+e & ; & b+4c+6d+4e+f & ; & c+4d+6e+4f+g & ; & d+4e+6f+4g+h \\ & & & \text{and to 3 machines,} & & & \\ a+5b+10c+10d+5e+f & ; & b+5c+10d+10e+5f+g & ; & c+5d+10e+10f+5g+h & & \\ & & & \text{and finally to 2 machines with processing times,} & & & \\ a+6b+15c+20d+15e+6f+g & \text{ and } & b+6c+15d+20e+15f+6g+h. & & & & \end{array}$$

That is, the two adjacent processing times are added together to get an other time and in this process, one machine gets reduced at each level.

In a similar way, the process is continued for all the jobs and the problem will be reduced to an n jobs and 2 machines problem which can conveniently be solved using Johnson's Algorithm. In the above procedure, at each stage, it may be noted that the coefficients of the processing times in each element are nothing but the members of a Pascal's Triangle for $nC_r, r=0, 1, \dots, n$.

That is, for m number of machines , the coefficients are the members of $(m-2)C_r$ in the final stage.

The algorithm can be presented as follows:

Step 1.

Let ' n ' number of jobs to be machined in ' m ' machines. It is assumed that all jobs are available for processing at time zero. At any time, each machine can process at most one job and each job can be processed on at most one machine

Step 2.

From the Pascal's Triangle, select the elements pertaining to $(m-2)C_r$.

Step 3.

For each Job, Multiply the machining times with the corresponding Pascal's Triangle elements and add them together ; for T_{i1} , the last machine is left out and for T_{i2} , the first machine is left out, $i=1,2,\dots,n$. The processing times of the two hypothetical machines can also be represented in Matrix form:

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \\ \dots & \dots \\ T_{n1} & T_{n2} \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2m} \\ \dots & \dots & \dots & \dots \\ t_{n1} & t_{n2} & \dots & t_{nm} \end{bmatrix} \times$$

$$\begin{bmatrix} (m-2)C_0 & 0 \\ (m-2)C_1 & (m-2)C_0 \\ \dots & \dots \\ (m-2)C_{(m-2)} & (m-2)C_{(m-3)} \\ 0 & (m-2)C_{(m-2)} \end{bmatrix}$$

Step 4.

Now, the problem is reduced to n jobs, 2 machines and then the optimum / near optimum make span is determined by using the data of the original problem using Johnson's Algorithm.

Note: If $m=2$, the procedure is reduced to Johnson's algorithm and If $m=3$, the procedure is reduced to extended Johnson's algorithm straight away.

5. PERFORMANCE OF THE NEW HEURISTIC ALGORITHM

To demonstrate the new Heuristics, an example described by Panneerselvam, as mentioned in Table-2 is considered. There are 4 jobs to be processed in 4 machines with corresponding processing times as shown in the table 2.

Table – 2, Example to illustrate the new Heuristic Algorithm

Processing Times in M/C				
Job	I	II	III	IV
1	4	3	7	8
2	3	7	2	5
3	1	2	4	7
4	3	4	3	2

Step 1

$n=4$ and $m=4$.

Step 2

$(m-2)C_r$ row has to be selected from the Pascal's Triangle. That is, $2C_r$ is selected and the elements are : **1 2 1**

Step 3

For Job no. 1: $t_{11} = 1(4)+2(3)+1(7) = 17$ and

$$t_{12} = 1(3)+2(7)+1(8) = 25$$

ie , the machining times are multiplied with the corresponding Pascal's Triangle elements and added together ; for t_{11} , the last machining is left out and for t_{12} , the first machining time is left out. (It may be noted that 3 elements are there and there are 4 machines)

For Job no. 2: $t_{21} = 1(3)+2(7)+1(2) = 19$ and

$$t_{22} = 1(7)+2(2)+1(5) = 16$$

For Job no. 3: $t_{31} = 1(1)+2(2)+1(4) = 09$ and

$$t_{32} = 1(2)+2(4)+1(7) = 17$$

For Job no. 4: $t_{41} = 1(3)+2(4)+1(3) = 14$ and

$$t_{42} = 1(4)+2(3)+1(2) = 12.$$

Step 4

Now, the problem has been reduced to a 2 machine problem as shown in Table – 3 and using the Johnson's algorithm the optimal sequence is computed.

Table – 3. Four machine problem reduced to two machine problem

Processing Times		
Job	M/C I	M/C II
1	17	25
2	19	16
3	9	17
4	14	12

And, the sequence obtained is: 3-1-2-4 with a make span of 30 units.

If Palmer's slope index method is applied, a sequence of 3-1-2-4 is found to be the optimal one.

If Gupta's Heuristics is used, a sequence of 3-2-1-4 with a make span of 31 units is obtained which is not optimal.

If RA Heuristics is used, a sequence of 3-2-1-4 with a make span of 31 units is obtained which is not optimal.

If CDS Heuristics is used, a sequence of 3-1-2-4 with a make span of 30 units is obtained which is optimal.

This example shows that the new proposed Heuristics gives the optimal sequence along with Palmer's method and CDS Heuristics for the given problem.

An other classical example as in Table 4, described in "A Heuristic Algorithm for the n Job m Machine Sequencing Problem" by Campbell et al. is being discussed in the following section. The total possible number of sequences is $8! = 40,320$.

Table-4, Example of an 8 Job, 7 Machine problem

Job	A	B	C	D	E	F	G
1	13	79	23	71	60	27	2
2	31	13	14	94	60	61	57
3	17	1	€	23	36	8	86
4	19	28	10	4	58	73	40
5	94	75	€	58	€	68	46
6	8	24	3	32	4	94	89
7	10	57	13	1	92	75	29
8	80	17	38	40	66	25	88

Note: The elements of value € require zero processing time but must pass the machine

In this case, number of jobs, $n=8$ and number of machines, $m=7$. Hence, we select $(m-2) C_r = 5C_5$ and the corresponding Pascal's elements are **1 5 10 10 5 1**. If the procedure stated in section 3 is followed, the problem can be reduced to 8 jobs and 2 machines as shown in Table-5

Table – 5. Seven machine problem reduced to two machine problem

Processing Times		
Job	M/C I	M/C II
1	1675	1641
2	1537	1985
3	440	717
4	662	1103
5	1117	1041
6	592	958
7	970	1456
8	1300	1480

Now, using the Johnson's Algorithm for n jobs, 2 machine problem the sequence can be found as 3-6-4-7-8-2-1-5. The make span is determined by using the data of the original problem and found to be 595 units.

In CDS algorithm, out of the 6 sequences obtained, the best one is 3-6-4-7-8-2-1-5 with a make span of 584 units which is the same as above.

Using the Smith-Dudek algorithm, it was determined that there is one optimal sequence to this problem with a total time of 584 units; it is 3-6-4-7-2-8-1-5.

The error, if taken as the deviation of the make span from the optimum solution time, is:

$$\frac{(595-584) \times 100}{584} = 1.9 \%$$

Using Gupta's Heuristic, the sequence obtained is 3-6-4-7-8-2-1-5 with a make span of 595 units and with RA Algorithm, the sequence is 3-6-4-7-2-8-5-1 with a make span of 596 units.

Finally, let us consider a random problem with 10 jobs to be processed in 10 different machines as shown in Table 6. The total possible number of sequences is $10! = 36, 28, 800$.

Table-6, Example of 10 Job, 10 Machine problem

Job	M/C1	M/C 2	M/C 3	M/C 4	M/C 5	M/C 6	M/C 7	M/C 8	M/C 9	M/C 10
1	5	2	3	5	7	9	7	8	2	7
2	2	6	4	2	6	2	5	2	6	1
3	1	2	2	1	3	7	2	5	4	4
4	7	5	6	3	2	3	2	4	2	2
5	6	6	1	8	6	4	3	9	6	4
6	3	7	5	2	2	1	5	3	2	6
7	7	2	4	6	5	5	1	2	5	2
8	5	1	7	1	7	3	6	6	2	2
9	7	8	6	9	1	8	2	1	6	6
10	4	3	5	8	3	1	3	8	3	7

In this case, number of jobs, $n=10$ and number of machines, $m =10$. Hence, we select $(m-2) C_r = 8C_8$ and the corresponding Pascal's elements are 1 **8 28 56 70 56 28 8 1**. If the procedure stated in section 3 is followed, the problem can be reduced to 10 jobs and 2 machines as shown in Table-7.

Table – 7. Ten machine problem reduced to two machine problem

Processing Times		
Job	M/C I	M/C II
1	1643	1828
2	968	955
3	831	992
4	781	701
5	1336	1326
6	673	671
7	1150	986
8	1141	1199
9	1331	1117
10	1069	952

Now, using the well known Johnson's Algorithm for n jobs, 2 machine problem the sequence can be found as 3-8-1-5-9-7-2-

10-4-6. The make span is determined by using the data of the original problem and found to be 103 units.

If Palmer's slope index method is applied, a sequence of 3-1-10-5-6-2-8-7-9-4 is obtained with a make span of 99 units.

If Gupta's Heuristics is used, a sequence of 3-6-10-1-5-9-8-2-7-4 with a make span of 103 units is obtained.

If RA Heuristics is used, a sequence of 3-10-1-5-9-8-6-2-7-4 with a make span of 97 units is obtained.

If CDS Heuristics is used, a sequence of 3-6-10-1-9-5-8-2-7-4 with a make span of 102 units is obtained.

It can be seen that the make span obtained using the proposed Heuristic Algorithm matches with the make span obtained from Gupta's Heuristic.

6. CONCLUSION

In this paper, one new heuristic algorithm based on Pascal's Triangle (for nC_r) is proposed for the flow shop scheduling of ' n ' jobs in ' m ' machines problem and illustrated using a few case studies. Also, the results are compared with a few other popular Heuristics and found to be giving reasonably good results. The authors are still working with quite a large number of problems varying from 3 jobs, 3 machines to larger ones. The Heuristic Algorithm described provides a practical solution to large sequencing problems that cannot be solved by exact procedures. Solutions produced by this algorithm are optimal or near-optimal and are easily and quickly produced. It is simple and can be easily coded in any high level language to run in a computer for effective and fast computation

7. REFERENCES

- [1] Johnson, S. M., "Optimal two and three machine production scheduling with set up times included", Naval Research, Log 1, No. 1 (1954).
- [2] Conway, R. W., Maxwell, W. L. and Miller, L. W. 1967 Theory Of Scheduling Dover Publications INC.
- [3] Baker, K. R. 1974 Introduction to sequencing and scheduling Wiley.
- [4] 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 (1965), 101-107.
- [5] Gupta, J. N. D., "A Functional Heuristic Algorithm for the Flow Shop Scheduling Problems", Operations Research 22 (1971), 39-47.
- [6] Campbell, H. G., Dudek, R. A. and Smith, M. L., "A Heuristic Algorithm for the n Job m Machine Sequencing Problem", Management Science 16 (1970), B630-637.
- [7] Dannenbring, D. G., "An Evaluation of Flow-Shop Sequencing Heuristics", Management Science 23 (1977), 1174-1182.
- [8] Nawaz, M., Ensore 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 (1983), 91-95.
- [9] Ho, J. C. and Chang, Y. I., "A new heuristic for the n-job, m-machine flow shop Problem", European Journal Of Operations Research 52 (1990), 194-206.
- [10] Quan-Ke-Pan, Ling Wang and Bao-Hua Zhao, "An improved iterated greedy algorithm for the no-wait flow shop scheduling problem with make span criterion", International Journal of Advanced Manufacturing Technology 38 (2008), 778-786.
- [11] Mohammad Kazem Sayadia, Reza Ramezani and Nader Ghaffari-Nazab, "A discrete firefly meta-heuristic with local search for make span minimization in permutation flow shop scheduling problems", International Journal of Industrial Engineering Computations 1 (2010), 1-10.
- [12] Uday Kumar Chakraborty and Dipak Laha, "An improved Heuristic for permutation flow shop scheduling", International Journal of Information and communication technology 1 No.1 (2007).
- [13] Dipak Laha and Uday Kumar Chakraborty, "An efficient hybrid heuristic for make span minimization in permutation flow shop scheduling", International Journal of Advanced Manufacturing Technology 44 (2009), 559-569.
- [14] Dipak Laha and Uday Kumar Chakraborty, "A constructive heuristic for minimizing make span in no-wait flow shop scheduling", International Journal of Advanced Manufacturing Technology 41 (2009), 97-109.
- [15] Dipak Laha and Arindam Chakravorty, "A new heuristic for minimizing total completion time objective in permutation flow shop scheduling", International Journal of Advanced Manufacturing Technology DOI 10.1007/s00170-010-2895-9.
- [16] Ravindran, D, Noorul Hag, A, Selva Kumar, S. J. and Siva Raman, R., "Flow shop scheduling with multi objective of minimizing make span and total flow time", International Journal of Advanced Manufacturing Technology 25 (2005), 1007-1012.
- [17] Allaverdi, A and Al-Anzi, F. S., "The two stage assembly flow shop scheduling problem with bi-criteria of make span and mean completion time", International Journal of Advanced Manufacturing Technology 37 (2008), 166-177.
- [18] Ming-Cheng Lo, Jen-Shing Chen and Yong-Fo Chang, "Minimizing make span and mean flow time in Two-Versatile machine Flow Shop with alternative operations", Information Technology Journal 9(2) (2010), 257-265.
- [19] Mohammad Mirabi, "Ant colony optimization technique for the sequence-dependent for the flow shop scheduling problem", International Journal of Advanced Manufacturing Technology DOI 10.1007/s00170-010-3037-0.
- [20] Panneer Selvam, R. 2005 Production and Operations Management Prentice Hall of India.