Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 9s (2025)

Minimizing Makespan in Three Phase Flowshop Scheduling Problems with Multiple Processor at Level First

Sonia[1], Pooja kaushik[2] Deepak Gupta[3], Sonia Goel[*4]

[1,2,3,4]Department of Mathematics, Maharishi Markandeshwar Engineering College, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala

[*]*Corresponding author* [*]*Email: Sonia.mangla14@gmail.com*

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:

This paper indicates the investigation of most efficient use of limited resources in production industries. Here the study provides a sequence of steps for solving the problems with three phase flowshop scheduling with equi-potential machines with the processing time accepted as Triangular Fuzzy numbers, with each machine having different operating cost.

Objectives:

The motivation behind this study is to find ideal or near ideal solution for solving the three-stage scheduling problem in fuzzy environment and making the best or most effective use of time and money.

Methods:

Here the methodology Branch and Bound is stacked up with the Genetic algorithm and Gupta's heuristic method. In our study we evaluated the technique of finding step by step process for solving the three-stage scheduling problem with processing time accepted as triangular fuzzy number at each stage . Various parameters like transportation time in between machines, no-idle time, job block etc. can be extended work for this study.

Conclusion:

In this study the point that should be observed that the Branch and Bound gave the makespan 139 but on the other hand by genetic algorithm the make span is different according to different iteration . It may happen that by chance, at any one iteration genetic algorithm gives more optimum solution ,but this will not possible always. Also, Gupta's Heuristic does not guarantee the optimum solution but prioritizes jobs based on the balanced between stages .This is useful when there is no need to minimum makespan but ensuring a smooth flow across the multiple stages.

Keywords: flow shop scheduling, triangular fuzzy numbers, Branch and Bound

I. Introduction

Success in today's modern industry requires a lot of advanced knowledge. Scheduling is one of them. Scheduling is step by step process of making the best or most effective use of time and money. Since we can't add hours to the day, so with the help of scheduling we do the right job at right time with better, faster effect and increase production capacity. Scheduling has major impact on workloads and productivity of certain assignment in any industry. The main concern here is to keep due dates of customers, minimize the total elapse time and cost of production. There are different types of scheduling like flow shop, job shop, and open shop etc. Our main concern here is on flowshop scheduling, which involves sequence of jobs with processing time accepted as triangular fuzzy number. The flow shop scheduling problems have different modern and financial applications so studied broadly by a number of researchers and scientists.

The basic concepts of flowshop production scheduling were given by Johnson's (1954) [1]. He found the optimal solution for n-job ,2-machine problem. Later on, Palmer (1965) [2] used the slop index and single iteration method to minimizing the make span. Campbell (1970) [3] worked on exact and approximation method for solving any size of n job, m-machines problems. Ashor (1974) [4] gave the development and computational results for the nonprimitive scheduling problem. JND Gupta (1971) [5] describes functional heuristic algorithm for finding an approximate solution for flowshop scheduling problems. Sujit K. Dutta et al. (1975) [6] determine the dynamic programming procedure for the flowshop sequencing problem. The approximation method is shown to provide an alternative means for large problems. M Nawaz (1983) [7] presented an algorithm on heuristic for different machines, where the priority given to machine with higher total processing time. A Nagar (1996) [8] studied the application of meta-heuristic procedures for ideal and near ideal solution. Neppalli et al. (1996) [9] approaches two-stage bicriteria flowshop scheduling problems and compare the study with heuristic approach. J Jungwattanakit et al. (2005) [10] evaluated heuristic approaches for scheduling problems to increase the quality of the constructive solution. T Erne et al. (2008) [11] gave bicriteria for level two flowshop problems for minimising the make span. C. Tseng (2008) [12] consider n job, m-machine problems to minimize the total earliness and tardiness. TCE Cheng (2009) [13] gives the concept of composite project with two machine flow shop scheduling problems. K R barker (2013) [14] provides accessible the core topics of book 'The Principle of Sequencing and Scheduling'. Gupta et al. (2013) [15] studied 3-phase flowshop scheduling problem including the transportation time and approached B&B method. Yenisey (2014) [16] worked on a number of methods to deal with multi-objective permutation flowshop scheduling issues. Bahman et al. (2014) [17] worked on no-idle open shops scheduling problems to reduce the make-span of jobs. They used modelling on meta-heuristic approaches and the concepts of mixed integer linear programming. Pan Q (2014) [18] deal with iterated greedy approach for no-idle permutation flowshop problems. Arun Gupta et al. (2015) [19] proposed heuristic approach to develop hybrid heuristic and meta heuristic approach for flowshop environment to minimize make span. Han Yuyan et al. (2016) [20] worked on a novel hybrid algorithm and the tabu search methods. Sonia et al. (2018) [21] worked with scheduling on three processor with parallel machines on first stage. M K Marichelvem et al. (2019) [22] [23] solved the problem of scheduling with hybrid algorithm. They worked on the firefly technique and the variable neighbourhood search technique to solve the hybride problems. S Goel et al. (2020) [24] found the solution procedure to solve the flowshop problems for two stages with parallel machines. They provide the step-by-step solution with modified distribution method. N Brauner (2021) [25] gave a number of properties of the problems of no-idle parallel machines scheduling. K Malhotra et al. (2022) [26] solved the problem of optimization of two stage parallel machines with triangular fuzzy number as transportation time. Gupta et al. (2022) [27] worked on branch and bound technique with equipotential parallel machines at every stage. K Malhotra et al. (2023) [28] extended the work by compared the branch and bound method by heuristic approach inspired by genetic algorithm. This study is a continuation of the work done by S Goel et al. [29] [26].

2. Statistical Development

Due to increasing demand of customer and other factors like time factor, breakdown of machines and other imposed policies, industries use multiple machines at different stages instead of using single machine. This type of model is very useful in various production industries producing a number of products in large scale in minimum time. The concept of triangular fuzzy numbers accepted as time gave an alternative way to solve the problems. Mathematical problem description is as follow.

Suppose there are n jobs (i= 1,2…., n) are to be performed on three processors *A*, *B* & *C* . which are called at first, second and third stage respectively. There is m multiple machines of types A_j (j= 1,2,3, 4….., m) at level one and single machine at level two and three. The utilization time is given as triangular fuzzy numbers at each stage. The unit operational cost of each job is also given on each equipotential machines at first stage. Processing time as triangular fuzzy numbers of processor A, B and C are taken of the form (x_{i1}, x_{i2}, x_{i3}) , (y_{i1}, y_{i2}, y_{i3}) and (z_{i1}, z_{i2}, z_{i3}) for i= 1,2,3,4, ……… ,n. and t_j (j=1,2,3, …… , m) is available time for all the equipotential machine in stage one. Table 1 show the mathematical model.

Table 1*: Structure of the model*

jobs	Processor A		Processor B	Processor C
(i)	A_1 A_2 A_3 …………… A_m	Processing time of A $\alpha_i=(x_{i1},x_{i2},x_{i3})$	Processing time of B $\beta_i=(y_{i1},y_{i2},y_{i3})$	Processing time of C $\delta_i=(z_{i1},z_{i2},z_{i3})$
1	A_{11} A_{12} A_{13} ……………. A_{1m}	(x_{11}, x_{12}, x_{13})	(y_{11},y_{12},y_{13})	(z_{11},z_{12},z_{13})
2	A_{21} A_{22} A_{23} ……………. A_{2m}	(x_{21}, x_{22}, x_{23})	(y_{21},y_{22},y_{23})	(z_{21},z_{22},z_{23})
3	A_{31} A_{32} A_{33} ……………. A_{3m}	(x_{31}, x_{32}, x_{33})	(y_{31},y_{32},y_{33})	(z_{31},z_{32},z_{33})
4	A_{41} A_{42} A_{43} ……………. A_{4m}	(x_{41}, x_{42}, x_{43})	(y_{41},y_{42},y_{43})	(z_{41},z_{42},z_{43})
5	A_{51} A_{52} A_{53} ……………. A_{5m}	(x_{51}, x_{52}, x_{53})	(y_{51},y_{52},y_{53})	(z_{51},z_{52},z_{53})
. . n	. . A_{n1} A_{n2} A_{n3} ……………. A_{nm}	. . (x_{n1}, x_{n2}, x_{n3})	. . (y_{n1},y_{n2},y_{n3})	. . (z_{n1},z_{n2},z_{n3})
(t_j)	t_1 t_2 t_3………………. t_m			

2.1 Assumptions

1. Jobs are independent to each other.

2. Setup times are insignificant.

3. All jobs need not be processed on all parallel machines.

4. There is different operating cost for every equipotential parallel machine.

5. Initial time may be same for all the equipotential parallel machines.

6. Machines are available throughout the scheduling period.

2.2 Problem solving Approach

Step (1): first of all, we will apply Average High-Ranking Formula of fuzzy dispensation time of different jobs by using the formula crisp(Ă) = h(Ă)= $\frac{3b+c-a}{3}$, where (a, b, c) denotes the fuzzy processing time of i^{th} job of corresponding processor. This process will convert the time taken as triangular fuzzy number to a single number.

Step (2): Check the condition $\sum_{r=1}^{m} t_r = \sum_{i=1}^{n} \alpha_i$

if the condition is satisfied then the optimal allotment of utilization time to every equipotential job can be found by using Modified Distribution Method.

The unbalanced case can be balanced with the constriction of dummy job or machines.

Step (3): Apply Branch and Bound method to compute the optimal schedule. The following divided parts sums up the whole method.

- Apply the formula

g'= max $(\sum_{i=1}^{n} A_{ij})$+ $\min_{i \epsilon j_{r'}}(\beta_i + \delta_i)$

g'' = $\max_{i \epsilon j_r}(A_{ij} + \beta_i)$+ $\sum_{i \epsilon j_{r'}} \beta_i$+ $\min_{i \epsilon j_{r'}} \delta_i$

g'''= $\max_{i \epsilon j_r}(A_{ij} + \beta_i + \delta_i)$ + $\sum_{i \epsilon j_{r'}} \delta_i$

- After first step, calculate l= max (g', g'', g''')
- Find out l for all the corresponding jobs.
- Find out the minimum of all the l's obtained in step (c) for all the corresponding jobs or tasks.
- The job corresponding to the minimum value calculated in above step will be at the first position in the optimal schedule.
- Now repeat the above process for (n-1) subsequence, then for (n-2) and so on. Finally, the process gives the optimal or near optimal schedule.
- Now for processing most advantageous schedule of jobs formulate in/out table for above step.

3. Explanation of algorithm by a numerical problem

we will explain the algorithm by giving a numerical problem for performing 4 tasks on processor A, B and C in the same order. Here we have taken three like parallel machines of type A. The processing time is supposed to be triangular fuzzy numbers for all the processor.

Table 2 shows the related numerical.

Table 2: ***Numerical problem:***

Processor A					Processor B	Processor C
Jobs (i)	A_1	A_2	A_3	Processing time of A (x_{i1},x_{i2},x_{i3})	Processing time of B (y_{i1},y_{i2},y_{i3})	Processing time of C (z_{i1} ,z_{i2} ,z_{i3})
1	2	7	4	(10 ,20, 40)	(03,08,09)	(15,20,30)
2	3	3	1	(05,15 ,20)	(05,15,20)	(02,14,20)
3	5	4	7	(02, 04,08)	(20,30,50)	(05,15,20)
4	1	6	2	(20 ,40 ,80)	(08,10,20)	(08,10,20)
(t_i)	50	30	36			

Solution:

Step1: According to step one of algorithm mention in problem solving approach, using Average High-Ranking formula to the processing time of A, B and C.

Table 3 shows reduced results.

Table 3:

Jobs	A				B	C
(i)	A_1	A_2	A_3	Processing time of A (α_i)	Processing time of B (β_i)	Processing time of C (δ_i)
1	2	7	4	30	10	25
2	3	3	1	20	20	20
3	5	4	7	06	40	20
4	1	6	2	60	14	14
(t_j)	50	30	36			

Step 2: As per second step of algorithm checks the mention condition there. Since we have

50+30+36=116 and also 30+20+06+60=116 so the condition is satisfied.

Now we will find the optimal allotment of utilization time to every equipotential machine by using Modified Distribution Method. Table 4 give the result;

Table 4: ***Optimal allocations of processing time***

Jobs	A			B	C
(i)	A_1	A_2	A_3	(β_i)	(δ_i)
1	30	0	0	10	25
2	0	20	0	20	20
3	0	06	0	40	20
4	20	04	36	14	14

Step3: Now we will compute the optimal sequence by the formula given in step 3.

LB(1)= max{78,128,119} =128

LB(2)= max{78,118,119}=119

LB(3)=max{78,104,125}=125

LB(4)=max{85,140,129}=140

Here minimum of these obtained lower bound for all the jobs separately is 119 i.e.

Min {128,119,125,140}=119 ,which corresponds to job 2.

Hence job 2 will be fix at first position in the optimal schedule. Now, repeat the above process for three subsequences ,which is (21) ,(23) and (24).

LB(21)= max{ 78,128,119 } =128

LB(23)= max{ 78,118,139 } =139

LB(24)= max{ 85,124,119 } =124

Here minimum { 128,139,124 }=124 ,which associated to subsequence (24). Therefore task 4 will be at second position in the optimal schedule. Now move to the next step.

LB(241)=max{ 110,140,125} =140

LB(243)=max{ 85,129,139} =139

Now min{140,139}=139 which associated with the subsequence (243). Therefore task 3 will be at third position of optimal schedule. And job 1 will be at 4th position.

Hence the required position of jobs in the optimal schedule will be { 2,4,3,1}

The processing of jobs for the best possible schedule{ 2,4,3,1 }will be show in table 5 in the form of in\out table

Table 5: ***Makespan for the best optimal sequence***

jobs	**A_1**		**A_2**		**A_3**		**B**		**C**	
(i)	**In**	**Out**	**In**	**Out**	**In**	**Out**	**In**	**Out**	**In**	**Out**
2	-	-	0	20	-	-	20	40	40	60
4	0	20	20	24	0	36	40	54	60	74
3	-	-	24	30	-	-	54	94	94	114
1	20	50	-	-	-	-	94	104	114	139

Hence the minimum make- span is 139.

4. Genetic Algorithm Approach

One of the most widely utilized stochastic optimization techniques for resolving intricate, large-scale optimization issues in a variety of domains is the genetic algorithm (GA). General concept of GA is inspired by the rule that good parents produce better generation .

Process to find out solution by genetic algorithm:

Step 1 Random initialize the solution say (Encoding)

Step 2 Ascertain fitness function

Step 3 Pick out parents from population

Step 4 Crossover and generate new solution called child

Step 5 Put in an application of mutation on new solution

Step 6 Calculate fitness function for new solution

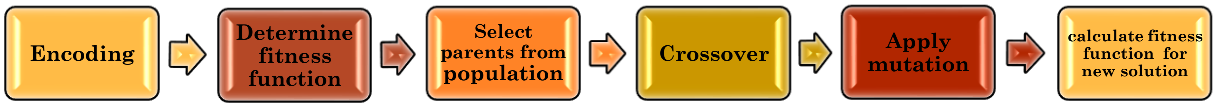


figure:1 ***process in GA***

Solution:

Step1: First of all, the initial population can be obtained by random arranging . We take four iterations. In iteration 1, suppose four chromosomes

Chromosome 1 = J4, J3, J2, J1

Chromosome 2 = J3, J2, J1, J4

Chromosome 3 = J2, J1, J4, J3

Chromosome 4 = J1, J4, J3, J2

Step 2: Now to determine the fitness function which is the minimum makespan.

Table 6: fitness value for the chromosome 1

Job(i)	A_1	A_2	A_3	**B**	**C**
4	0-20	0-4	0-36	36-50	50-64
3	-	4-10	-	50-90	90-110
2	-	10-30	-	90-110	110-130
1	20-50	-	-	110-120	130-155

The fitness value is 155.

Table 7: In-Out table for the chromosome 2

Job(i)	A_1	A_2	A_3	**B**	**C**
3	-	0-6	-	6-46	46-66
2	-	0-20	-	46-66	66-86
1	0-30	-	-	66-76	86-111
4	30-50	50-54	0-36	76-90	111-125

The fitness value is 125.

Table 8: In-Out table for the chromosome 3

Job(i)	A_1	A_2	A_3	B	C
2	-	0-20	-	20-40	40-60
1	0-30	-	-	40-50	60-85
4	30-50	50-54	0-36	54-68	85-99
3	-	54-60	-	68-108	108-128

The fitness value is 128.

Table 9: In-Out table for the chromosome 4

Job(i)	A_1	A_2	A_3	B	C
1	0-30	-	-	30-40	40-65
4	30-50	0-4	0-36	40-54	65-79
3	-	4-10	-	54-94	94-114
2	-	10-30	-	94-114	114-134

The fitness value is 134.

Step 3 Selection of parents from fitness function

From step 2 the two minimum fitness value is 124 and 125 which is corresponding to chromosome 3 and chromosome 2. Now selecting chromosome 3 as parent 1 and chromosome 2 as parent 2. i.e.

Parent 1 = J2, J1, J4, J3

Parent 2 = J3, J2, J1, J4

Step 4 Crossover to create new solution

Edge Table 1

jobs	Connected jobs		
1	4	3	2
2	1	4	3
3	2	1	4
4	1	2	3

Edge Table2

jobs	Connected jobs		
1	3	4	2
2	4	1	3
3	1	4	2
4	3	2	1

By crossover, from edge table 1, Child 1 is J2, J3, J4, J1 and from edge table 2, child 2 is J3, J4, J2, J1.

Step 5 Apply mutation on new solution

Interchange J4 and J1 in child 1, the final sequence is J2, J3, J1, J4

And interchange J2 and J1 in child 2, the final sequence is J3, J4, J1, J2.

The in-out table for child 1 and child 2 is as follow in table 10;

Table10

Makespan for child 1						Makespan for child 2					
jobs	A_1	A_2	A_3	B	C	jobs	A_1	A_2	A_3	B	C
2	-	00-20	-	20-40	40-60	3	-	0-6	-	06-46	46-66
3	-	20-26	-	40-80	80-100	4	0-20	20-24	00-36	46-60	66-80
1	0-30	-	-	80-90	100-125	1	20-50	-	-	60-70	80-105
4	30-50	50-54	00-36	90-104	125-139	2	-	24-44	-	70-90	105-135

Here minimum make-span is 135 as shown in table 10 ,which corresponds to child 2. So { 3, 4, 1, 2 } is the optimal sequence for iteration 1.

Note: For iteration 2 ,iteration 3 and iteration 4 we get the optimal schedule {4 ,2,1,3},{3,1,2,4}and {4,3,1,2} respectively, with the makespan 140,125 and 155.

5. Solution with the help of Gupta's Heuristics

Gupta's Heuristic [5] is well-known guideline for solving scheduling issues in flow shop. This is quick solving approach specifically designed for more than two machines flowshop problems. This method is based on a priority index. Process to apply Gupta's Heuristic is as follow;

1 Calculate the priority index for each job:

The priority index is designed to balance the processing times across the stages of flow shop . The general formula for priority index is;

$S_j = \frac{S_{j,k+1} - S_{J,k}}{S_{j,k+1} + S_{j,k+2}}$

Where:

S_j is the priority index for task j.

$S_{j,k}$ is the processing period of task j at level k^{th} .

$S_{j,k+1}$ is the processing period of task j at level $(k+1)^{th}$.

$S_{j,k+2}$ is the processing period of task j at level $(k+2)^{th}$.

2 Arrange the jobs in ascending order based on priority index and make the sequence.

3 Assign job to machines according to obtained sequence and calculate the makespan.

Solution: Consider the problem from table 4, we have four jobs and three stages. First of all, calculate the aggregate processing time at stage one ,we get ;

Table 11:

Job(i)	Level 1	Level 2	Level 3
1	30	10	25
2	20	20	20
3	06	40	20
4	60	14	14

Step1: Calculate the priority index for each job:

Here , $S_j = \frac{S_{j,2}-S_{j,1}}{S_{j,2}+S_{j,3}}$

Now, $S_1=\frac{10-30}{10+25} = \frac{-20}{35} = -0.57$

$S_2 = \frac{20-20}{20+20} = \frac{0}{40} = 0$

$S_3 = \frac{40-6}{40+20} = \frac{34}{60} = 0.567$

$S_4 = \frac{14-60}{14+14} = \frac{-46}{28} = -1.642$

Step2: Arrange the job in ascending order according to the priority index ,we get the result showing in table 12.

Table 12:

Priority index	-1.642	-0.57	0	0.567
Jobs	4	1	2	3

We have found the sequence { 4,1,2,3 }

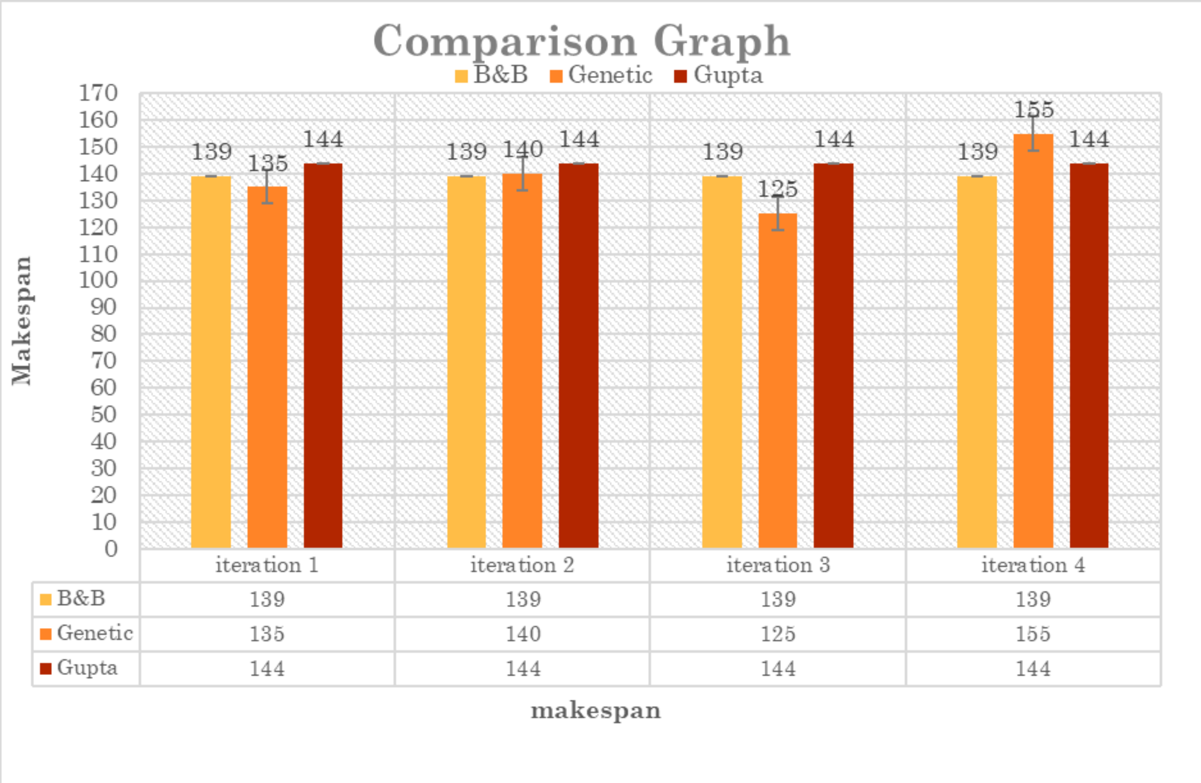
Step3: Calculate the makespan of the sequence obtained in step 2 ,

Table 13: Makespan for schedule { 4,1,2,3 }

jobs	A_1	A_2	A_3	B	C
4	0-20	0-4	0-36	36-54	54-68
1	20-30	-	-	54-64	68-93
2	-	4-24	-	64-84	93-113
3	-	24-30	-	84-124	124-**144**

The makespan is 144.

Comparison between B&B, Genetic Algorithm , Gupta's Heuristic



Conclusion:

In our study we evaluated the technique of finding step by step process for solving the three-stage scheduling problem with processing time accepted as triangular fuzzy number at each stage . Various parameters like transportation time in between machines, no-idle time ,job block etc. can be extended work for this study. In this study the point that should be observed that the Branch and Bound gave the makespan 139 but on the other hand by genetic algorithm the make span is different according to different iteration . It may happen that by chance, at any one iteration genetic algorithm gives more optimum solution ,but this will not possible always. Also, Gupta's Heuristic does not guarantee the optimum solution but prioritizes jobs based on the balanced between stages .This is useful when there is no need to minimum makespan but ensuring a smooth flow across the multiple stages.

References

[1] Johnson, Selmer Martin. "Optimal two-and three-stage production schedules with setup times included." *Naval research logistics quarterly* 1, no. 1 (1954): 61-68.

[2] Palmer, Douglas S. "Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum." *Journal of the Operational Research Society* 16, no. 1 (1965): 101-107.

[3] Campbell, Herbert G., Richard A. Dudek, and Milton L. Smith. "A heuristic algorithm for the n job, m machine sequencing problem." *Management science* 16, no. 10 (1970): B-630.

[4] Ashour, Said, T. E. Moore, and Kung-Ying Chiu. "An implicit enumeration algorithm for the nonpreemptive shop scheduling problem." *AIIE Transactions* 6, no. 1 (1974): 62-72.

[5] Gupta, Jatinder ND. "A functional heuristic algorithm for the flowshop scheduling problem." *Journal of the Operational Research Society* 22, no. 1 (1971): 39-47.

[6] Dutta, Sujit K., and Andrew A. Cunningham. "Sequencing two-machine flow-shops with finite intermediate storage." *Management Science* 21, no. 9 (1975): 989-996.

[7] Nawaz, Muhammad, E. Emory Enscore Jr, and Inyong Ham. "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem." *Omega* 11, no. 1 (1983): 91-95.

[8] Nagar, Amit, Sunderesh S. Heragu, and Jorge Haddock. "A combined branch-and-bound and genetic algorithm-based approach for a flowshop scheduling problem." *Annals of Operations Research* 63 (1996): 397-414.

[9] Neppalli, Venkata Ranga, Chuen-Lung Chen, and Jatinder ND Gupta. "Genetic algorithms for the two-stage bicriteria flowshop problem." *European journal of operational research* 95, no. 2 (1996): 356-373.

[10] Jungwattanakit, Jitti, Manop Reodecha, Paveena Chaovalitwongse, and Frank Werner. "An evaluation of sequencing heuristics for flexible flowshop scheduling problems with unrelated parallel machines and dual criteria." *Otto-von-Guericke-Universitat Magdeburg* 28, no. 05 (2005): 1-23.

[11] Eren, Tamer, and Ertan Güner. "A bicriteria flowshop scheduling with a learning effect." *Applied Mathematical Modelling* 32, no. 9 (2008): 1719-1733.

[12] Tseng, Chao-Tang, and Ching-Jong Liao. "A discrete particle swarm optimization for lot-streaming flowshop scheduling problem." *European Journal of Operational Research* 191, no. 2 (2008): 360-373.

[13] Cheng, TC Edwin, and Bertrand MT Lin. "Johnson’s rule, composite jobs and the relocation problem." *European Journal of Operational Research* 192, no. 3 (2009): 1008-1013.

[14] Baker, Kenneth R., and Dan Trietsch. *Principles of sequencing and scheduling*. John Wiley & Sons, 2018.

[15] Gupta, Deepak, Sourav Singla, Payal Singla, and Sukhvir Singh. "Minimization of elapsed time in N× 3 flow shop scheduling problem, the processing time associated with probabilities including transportation time." *International Journal of Innovations in Engineering and Technology Special Issue–ICAECE* (2013).

[16] Yenisey, Mehmet Mutlu, and Betul Yagmahan. "Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends." *Omega* 45 (2014): 119-135.

[17] Naderi, Bahman, and Vahid Roshanaei. "No-idle time Scheduling of Open shops: Modeling and Meta-heuristic Solution Methods." *International Journal of Supply and Operations Management* 1, no. 1 (2014): 54-68.

[18] Pan, Quan-Ke, and Rubén Ruiz. "An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem." *Omega* 44 (2014): 41-50.

[19] Gupta, Arun, and S. Chauhan. "A heuristic algorithm for scheduling in a flow shop environment to minimize makespan." *International Journal of Industrial Engineering Computations* 6, no. 2 (2015): 173-184.

[20] Han, Yuyan, et al. "Solving the blocking flow shop scheduling problem with makespan using a modified fruit fly optimisation algorithm." *International Journal of Production Research* 54.22 (2016): 6782-6797.

[21] Malhotra, Khushboo, Deepak Gupta, Sonia Goel, and A. K. Tripathi. "Optimization in makespan working with like parallel machines." In *Sustainability in Digital Transformation Era: Driving Innovative & Growth*, pp. 20-25. CRC Press, 2024.

[22] Marichelvam, M. K., and M. Geetha. "A hybrid algorithm to solve the stochastic flow shop scheduling problems with machine break down." *International Journal of Enterprise Network Management* 10.2 (2019): 162-175

[23] Marichelvam, M. K., M. Geetha, and Ömür Tosun. "An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors–A case study." *Computers & Operations Research* 114 (2020): 104812.

[24] Gupta, D. E. E. P. A. K., and S. O. N. I. A. Goel. "Nx2 flow shop scheduling problem with parallel machines at every stage, processing time associated with probabilities." *Adv Math Sci J* 9, no. 3 (2020): 1061-1069.

[25] Brauner, Nadia, Mikhail Y. Kovalyov, Alain Quilliot, and Hélène Toussaint. "No-idle parallel-machine scheduling of unit-time jobs with a small number of distinct release dates and deadlines." *Computers & Operations Research* 132 (2021): 105315.

[26] Malhotra, K., D. Gupta, S. Goel, and A. K. Tripathi. "Bi-objective flow shop scheduling with equipotential parallel machines." *Malays J Math Sci* 16, no. 3 (2022): 451-470.

[27] Gupta, Deepak, and Sonia Goel. "Branch and bound technique for two stage flow shop scheduling model with equipotential machines at every stage." *International Journal of Operational Research* 44, no. 4 (2022): 462-472.

[28] Malhotra, Khushboo, Deepak Gupta, Sonia Goel, and A. K. Tripathi. "Comparison of Bb with Meta-heuristic Approach in Optimization of Three Stage Fss with Multiple Processors." (2023).

[29] Gupta, Deepak, Sonia Goel, and Neeraj Mangla. "Optimization of production scheduling in two stage flow shop scheduling problem with m equipotential machines at first stage." *International Journal of System Assurance Engineering and Management* 13, no. 3 (2022): 1162-1169.