Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2025 (43)** : 1–13.
ISSN-0037-8712
doi:10.5269/bspm.76385

Optimization in Three Phase Flowshop Scheduling Problem with Multiple Processor at First Level Including Transportation Time

[1]Sonia, *[2]Sonia Goel,[3]Deepak Gupta

ABSTRACT: The methodical process of organizing, managing, and maximizing work while ensuring the greatest possible use of both time and resources is known as scheduling. This paper indicates the desirable and necessary steps to discover the optimum solution for the three-phase flowshop scheduling problems. Here the multiple processors have been taken at the first phase and single processor at 2^{nd} and 3^{rd} stage along with transportation time between machines. Here the first methodologies B&B(Branch and Bound) is compared with the different heuristic methodologies like NEH(Nawaz Enscore Ham) and CDS(Campbell Dudek Smith) to solve the mention problem. Comparative study considered to select the best methodology among the three with the help of numerical example.

Key Words: Transportation time, Multiple machines, Branch and Bound, Processing Time, CDS, NEH, Branch and Bound

Contents

1 Introduction **1**

2 Importance and practical situations **2**
2.1 Assumption . . . 2
2.2 Mathematical Development . . . 2
2.3 Problem solving Approach . . . 3

3 Explanation of algorithm by a numerical problem **4**

4 Solution by the NEH method **6**

5 Solution by the CDS method **9**

6 Comparison between B&B, NEH, CDS **12**

7 Conclusion **12**

1. Introduction

In today real word scenarios, any organization or business industries requires a lot of advanced knowledge for success. Scheduling is one of the important concepts for the required purpose. Due to scheduling all the manufacturing systems follow a pre-planned schedule of machines making the best and most effective use of time and money and reduced the completion time, so scheduling is an important factor for the modern productivity. In this study we deal with the one of the essential types of scheduling that is flow-shop scheduling that consist of a number of machines with different tasks and every task is based on different operations that needed different machines to operate on. The concept of parallel machines at different stages in flow shop scheduling have solved many complexities and increased the production capacities and profitability. In the present paper we have taken parallel machines at the 1^{st} phase and single machines at 2^{nd} and 3^{rd} phase with the transportation time, which is very important model for business or production industries. The key point of this study is to minimize the make-span and keep due date of customer's demand with the maximum profit.

* Corresponding author
2010 *Mathematics Subject Classification:* 90B35, 68M20.
Submitted April 15, 2025. Published September 01, 2025

The flow shop scheduling problems have different financial and modern applications , as a result these applications are studied by a number of researchers and scientists from time to time. Johnson [13] was the first to solve two or three stage machine problems to minimize the total elapse time. Palmer et.al [10] studied the sequencing of jobs through a multistage process and compared with different methods .Yeh et.al [14] gave an algorithm B&B incorporating developed bound and dominance relation in three stage flow shop scheduling .Yingjie et al. [?] extended application of hybrid flowshop scheduling based on parallel sequence movement and multi-equipment hybrid flow shop scheduling. Allahverdi et.al [?] gave a survey on scheduling-problems and categorised the literature survey according to job-shop, open-shop, flow-show and others. Smith et.al [12] extended the work by giving an algorithm on n job, m machines. There was a notable method by Ignall et.al [6] which was Branch and Bound for $n-$job, 3 machines problem. Gupta et al. [3] developed the heuristic algorithm and reduced weightage of machines at different stages. Yee Chung et al. [8] worked on m different identical machines for n-independent jobs and generalized classical-multiprocessor scheduling problems. Ismail et al. [7] used the dynamic approach and job block methods on the minimization of the tardiness of machines. Malhotra Khushboo et al. [9] extended the vision of parallel machines at all stages with utilization time as triangular fuzzy numbers, also implemented the heuristic methodology like genetic algorithm. Sonia Goel et al. [4], [5] extended the work with multiple stage flow-shop scheduling on parallel machines and the study was inspired by B&B algorithm.

2. Importance and practical situations

The applications of flow-shop scheduling models have been discussed by a number of researchers because of its vast financial and modern uses. Production industries or manufacturing companies need different type of work on different type of machines. The quantities of these machines depend on which types of tasks has to be completed on. Sometime the set-up of the machines planted very far so the transportation time play very important role in that case. It increases the complete time of job so in this competitive word it is necessity of that organization to follow a pre-planned schedule for successful management. There are some practical situations where the scheduling can be used like in photocopiers and laser printers in printing circuit boards, in the textile industries, in fabric industries, when the fabric types are assigned and changed on machine. The changing instrument replaced according to fabric type, and takes time according to the current and last fabric type. A lot of applications of world wide web across a communication network require access, transfer and synchronize a large data object like images, audios and videos. So, an effective scheduling strategy is necessary to provide to distributed system. The model of these scheduling strategy can be made in two or three stage scheduling with different set up times. Another example is the tile industries, the problem of grouping the materials and products can be solved by multiple phase flowshop scheduling with separate time. Main concern for such type of problems will be reduce the production time and provide a necessary aspect for decision makers.

2.1. Assumption

1. Takeover of job is not recommended.
2. All jobs are independent to each other.
3. The processors are available throughout the period of processing.
4. The operating cost of each job should be different.
5. All jobs need not be processed on all the parallel machines on first stage.
6. Initial time may be same for all the equipotential parallel machines.

2.2. Mathematical Development

The concept of multiple machines at different stage of flow shop scheduling is very important in today competitive word scenarios. Because due to demand of customer and other important factor like time factor, breakdown of machines and other external imposed policies, companies and industries are using multiple machines instead of using single machine. In our study we are using multiple machines at 1^{st} and single machine at 2^{nd} and 3^{rd} phase. This type of model is very useful for producing a number of products at large scale in minimum time. The concept of transportation time worked as icing on the cake because sometimes the set of machines and raw materials are not available at the same place. Mathematical description of the problem is as follow:

Suppose there are three processor $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ (in the same order) on which n job ($i = 1, 2, 3, 4, \cdots, n$) are to be performed. The processors are in the first , second and third stage. There are m parallel machines on first stage of type $\tilde{A}_j (j = 1, 2, 3, 4, \cdots, m)$ and single machine at 2^{nd} and 3^{rd} phase. α_i, β_i and γ_i are the processing time of machine $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ respectively. $\tilde{A}_{ij}$ are the unit operational cost of i^{th} job at j^{th} machine on $\tilde{A}_i$ like parallel machines. Suppose T_i and T'_i are the transportation time of job i from machine on $\tilde{A}$ to $\tilde{B}$ and $\tilde{B}$ to $\tilde{C}$ respectively. Also, t_i is available time for like parallel machines of first stage. The motivation behind the model is to settle all jobs in such a way that the work can done in shortest time. The model of the problem is described below in table 1.

Processor $\tilde{A}$			Processor $\tilde{B}$		Processor $\tilde{C}$	
Tasks (i)	$\tilde{A}_1\ \tilde{A}_2\ \tilde{A}_3\ \tilde{A}_4 \cdots\cdots \tilde{A}_m$	Processing time of $\tilde{A}$ α_i	T_i	Processing time of $\tilde{B}$ β_i	T'_i	Processing time of $\tilde{C}$ γ_i
1	$\tilde{A}_{11}\ \tilde{A}_{12}\ \tilde{A}_{13}\ \tilde{A}_{14} \cdots \tilde{A}_{1m}$	α_1	T_1	β_1	T'_1	γ_1
2	$\tilde{A}_{21}\ \tilde{A}_{22}\ \tilde{A}_{23}\ \tilde{A}_{24} \cdots \tilde{A}_{2m}$	α_2	T_2	β_2	T'_2	γ_2
3	$\tilde{A}_{31}\ \tilde{A}_{32}\ \tilde{A}_{33}\ \tilde{A}_{34} \cdots \tilde{A}_{3m}$	α_3	T_3	β_3	T'_3	γ_3
4	$\tilde{A}_{41}\ \tilde{A}_{42}\ \tilde{A}_{43}\ \tilde{A}_{44} \cdots \tilde{A}_{4m}$	α_4	T_4	β_4	T'_4	γ_4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$\tilde{A}_{n1}\ \tilde{A}_{n2}\ \tilde{A}_{n3}\ \tilde{A}_{n4} \cdots \tilde{A}_{nm}$	α_n	T_n	β_n	T'_n	γ_n
t_i	$t_1\ t_2\ t_3\ t_4\ \cdots t_m$					

Table 1: Proposed Mathematical model

2.3. Problem solving Approach

We will solve the problem in different phases.

Phase1: Convert the problem by creating three fictitious machine $\tilde{K}$, $\tilde{L}$ and $\tilde{M}$ with processing time k_i, l_i and m_i defined by,

$$k_i = \alpha_i,\ l_i = T_i + \beta_i \text{ and } m_i = T'_i + \gamma_i$$

Phase2: Now check the condition $\sum t_i = \sum \alpha_i$, this condition indicates that problem is balanced and we will find the optimal allocations of processing time for each job in the first stage i.e. like parallel machines by applying VAM & MODI method.

If the condition fails i.e. the problem is not balanced ,then first make it balanced by adding dummy job or machines and repeat the above procedure.

Phase3: Now the B&B method for the required optimal schedule by applying the following steps;

1. Apply the formula

$$G''' = max\left(\sum_{i=1}^{n} \tilde{A}_{ij}\right) + \min_{i \in j_{r'}} \left(l_i + m_i\right)$$

$$G'' = \max_{i \in j_r} \left(\tilde{A}_{ij} + l_i\right) + \sum_{i \in j_{r'}} l_i + \min_{i \in j_{r'}} m_i$$

$$G' = \max_{i \in j_r} \left(\tilde{A}_{ij} + l_i + m_i\right) + \sum_{i \in j_{r'}} m_i$$

2. Calculate $G = max\{G''', G'', G'\}$
3. Find out G for all the corresponding jobs.
4. Find out minimum of all the G's obtained above corresponding to all jobs and repeat the process for optimal schedule.
5. Find the make-span for obtained schedule.

3. Explanation of algorithm by a numerical problem

We will explain the algorithm by giving a numerical problem for performing 5 tasks on processor $\tilde{A}$, $\tilde{B}\, and\, \tilde{C}$ in the same order. Here we have taken four parallel machines of type $\tilde{A}$. In table 2, The numerical example is given;

Processor $\tilde{A}$						Processor $\tilde{B}$		Processor $\tilde{C}$	
Jobs (i)	$\tilde{A}_1$	$\tilde{A}_2$	$\tilde{A}_3$	$\tilde{A}_4$	α_i	T_i	β_i	T'_i	γ_i
1	1	5	8	6	7	2	5	2	3
2	7	2	6	9	8	1	2	3	4
3	5	9	3	7	5	3	2	4	5
4	3	6	7	4	6	4	3	5	2
5	5	4	2	8	4	5	4	2	1
t_i	9	7	6	8					

Table 2: Numerical example

Solution:
Phase1: Convert the problem by creating three fictitious machine $\tilde{K}$, $\tilde{L}\, and\, \tilde{M}$ with processing time k_i, l_i andm_i defined by,

$$k_i = \alpha_i,\ l_i = T_i + \beta_i \text{ and } m_i = T'_i + \gamma_i$$

the converted problem takes the form

Processor $\tilde{K}$						*Processor* $\tilde{L}$	*Processor* $\tilde{M}$
Jobs(i)	$\tilde{A}_1$	$\tilde{A}_2$	$\tilde{A}_3$	$\tilde{A}_4$	k_i	l_i	m_i
1	1	5	8	6	7	7	5
2	7	2	6	9	8	3	7
3	5	9	3	7	5	5	9
4	3	6	7	4	6	7	7
5	5	4	2	8	4	9	3
t_i	9	7	6	8			

Table 3: Converted problem

Phase2: Here sum of all k_i's and t_i's are same i.e.30
So, the problem is balanced. Now, we will find the optimal allocations of processing time and optimal allotment of unit cost for each job by applying VAM and MODI method. After phase 2 the reduced table takes the form given in table 4;

Processor $\tilde{K}$					Processor $\tilde{L}$	Processor $\tilde{M}$
Jobs (i)	$\tilde{A}_1$	$\tilde{A}_2$	$\tilde{A}_3$	$\tilde{A}_4$	l_i	m_i
1	7	0	0	0	7	5
2	1	7	0	0	3	7
3	1	0	2	2	5	9
4	0	0	0	6	7	7
5	0	0	4	0	9	5

Table 4: Optimal allotment of operating time on parallel machines

Phase3: Now the B&B method outcomes are shown in table 5.

Jobs (i)	$G' = \max_{i \in j_r}\left(\tilde{A}_{ij} + l_i + m_i\right) + \sum_{i \in j_{r'}} m_i$	$G'' = \max_{i \in j_r}\left(\tilde{A}_{ij} + l_i\right) + \sum_{i \in j_{r'}} l_i + \min_{i \in j_{r'}} m_i$	$G''' = max\left(\sum_{i=1}^{n} \tilde{A}_{ij}\right) + \min_{i \in j_{r'}}\left(l_i + m_i\right)$	$G = max\{G'''.G'', G'\}$
1	$7+7+31=45$	$7+31+3=41$	$9+10=19$	45
2	$7+3+31=41$	$7+31+3=41$	$9+12=21$	41
3	$2+5+31=38$	$2+31+3=36$	$9+10=19$	38
4	$6+7+31=44$	$6+31+3=40$	$9+10=19$	44
5	$4+9+31=44$	$4+31+5=40$	$9+10=19$	44

Table 5: Optimal allotment of operating time on parallel machines

Here minimum of all the G's i.e. $min\{45, 41, 38, 44, 44\} = 38$, which corresponds to task 3. So, position of job 3 in the optimum sequence will be at first place. Now repeat the above process for the subsequence $\{3, 1\}$, $\{3, 2\}$, $\{3, 4\}$ and $\{3, 5\}$ we get the required result in table 6 as below;

Job subsequence	G'	G''	G'''	$G = max\{G''', G'', G'\}$
$\{3, 1\}$	$9+10=19$	$15+19+3=37$	$20+17=37$	37
$\{3, 2\}$	$9+12=21$	$9+23+3=35$	$22+15=37$	37
$\{3, 4\}$	$9+10=19$	$14+19+3=36$	$22+15=37$	37
$\{3, 5\}$	$9+10=19$	$15+17+5=37$	$9+10=19$	37

Table 6: Branch and Bound procedure for 2nd position in optimal sequence

Here minimum of all the G's is 37, which corresponds to all the sub sequences in table 6. So, there will be more than one optimal schedule. We may fix any of the jobs 1, 2, 4 and 5 on 2^{nd} place of optimal schedule. We proceed by taking job 1 at the second place. In table 7 repeat the process for sub sequence $\{3, 1, 2\}$, $\{3, 1, 4\}$ and $\{3, 1, 5\}$, we get;

Job subsequence	G'	G''	G'''	$G = max\{G''', G'', G'\}$
$\{3, 1, 2\}$	$9 + 12 = 21$	$18 + 16 + 3 = 37$	$27 + 10 = 37$	37
$\{3, 1, 4\}$	$9 + 10 = 19$	$22 + 12 + 3 = 37$	$29 + 10 = 39$	39
$\{3, 1, 5\}$	$9 + 10 = 19$	$24 + 10 + 7 = 41$	$27 + 14 = 41$	41

Table 7: Branch and Bound procedure for 3rd position in optimal sequence

Now, the $min\{37, 39, 41\} = 37$, which corresponds to sub sequence $\{3, 1, 2\}$. Therefore job 2 will be at the 3^{rd} position of optimal schedule. In table 8 repeat the process for sub sequence $\{3, 1, 2, 4\}$ and$\{3, 1, 2, 5\}$

Job subsequence	G'	G''	G'''	$G = max\{G''', G'', G'\}$
$\{3, 1, 2, 4\}$	$9 + 12 = 21$	$25 + 9 + 3 = 37$	$34 + 3 = 37$	37
$\{3, 1, 2, 5\}$	$9 + 14 = 23$	$27 + 7 + 7 = 41$	$30 + 7 = 37$	41

Table 8: Branch and Bound procedure for 4^{th} position in the optimal sequence

Clearly, $min\{37, 41\} = 37$, which corresponds to sub sequence $\{3, 1, 2, 4\}$. So, job 4 will be at the 4^{th} position of optimal schedule and remaining job 5 will be at last .so the optimal schedule will be $\{3, 1, 2, 4, 5\}$. Table 9 show the in-out table,

Jobs (i)	$\tilde{A}_1$ In-out	$\tilde{A}_2$ In-out	$\tilde{A}_3$ In-out	$\tilde{A}_4$ In-out	$\tilde{L}$ In-out	$\tilde{M}$ In-out
3	$0 - 1$	–	$0 - 2$	$0 - 2$	$2 - 7$	$7 - 16$
1	$1 - 8$	–	–	–	$8 - 15$	$16 - 21$
2	$8 - 9$	$0 - 7$	–	–	$15 - 18$	$21 - 28$
4	–	–	–	$2 - 8$	$18 - 25$	$28 - 35$
5	–	–	$2 - 6$	–	$25 - 34$	35–**38**

Table 9: In-Out table for the job order $\{3, 1, 2, 4, 5\}$

This show that the minimum make span is 38.
Note: Here $\{3, 2, 1, 4, 5\}$, $\{3, 4, 2, 1, 5\}$ and $\{3, 5, 2, 4, 1\}$ will also the optimal schedule with the same makespan 38.

4. Solution by the NEH method

NEH (Nawaz Enscore Ham) method is one of the heuristic methods which was specially created in 1983 [11] to address flow-shop scheduling problems. This method is based on the idea that jobs with longer total processing times should be prioritized above those with shorter total times. The NEH can be very effective due to its global search exploration capabilities. In present problem scenario of flow shop scheduling, the following steps can effectively explore the search space and will help in finding the optimal solution.

Algorithm steps:
Step1: First of all, calculate the total processing time i.e. $\sum \alpha_i + \beta_i + \gamma_i$ for each job.
Step2: After finding the total processing time, keep all the jobs in decreasing order based on the total

processing time.
Step3: From the descending order schedule, select the first two jobs and find the total makespan.
Step4: Insert next subsequent job into the existing schedule at the possible position and calculate the minimum make span.
Step5: Repeat the process until all the jobs included in the schedule.
Solution: Taking the problem from table 4

Jobs	$\tilde{K}$	$\tilde{L}$	$\tilde{M}$	Total Processing time
1	7	7	5	19
2	8	3	7	18
3	5	5	9	19
4	6	7	7	20
5	4	9	3	16

Table 10: Total processing time

Step1: Calculate the total processing time of each job as shown in table 10.
Step2: Now keep all the jobs in descending order, which are $J_4 - J_1 - J_3 - J_2 - J_5$ or it can be $J_4 - J_3 - J_1 - J_2 - J_5$. we will proceed the problem by taking first sequence.
Step3: Select the first two jobs from the selected schedule which are $J_4 - J_1$ or we can write $J_1 - J_4$ calculate the makespan

Jobs	$\tilde{K}$	$\tilde{L}$	$\tilde{M}$
J_4	6	13	20
J_1	13	20	**25**
J_1	7	14	19
J_4	13	21	28

Table 11: Makespan for $J_4 - J_1$ and $J_1 - J_4$

since the minimum makespan is 25 so we choose $J_4 - J_1$.
Step4: Now insert the next job J_3 in the current schedule. There are three ways $J_3 - J_4 - J_1$, $J_4 - J_3 - J_1$ and $J_4 - J_1 - J_3$ to inserting. Again, calculate the makespan for all the three schedule.

Jobs	$\tilde{K}$	$\tilde{L}$	$\tilde{M}$
J_3	5	10	19
J_4	11	18	26
J_1	18	25	**31**
J_4	6	13	20
J_3	11	18	29
J_1	18	25	34
J_4	6	13	20
J_1	13	20	25
J_3	18	25	34

Table 12: Makespan for $J_3 - J_4 - J_1$, $J_4 - J_3 - J_1$ and $J_4 - J_1 - J_3$

Here the minimum among all the three makespan is 31, so we choose the schedule $J_3 - J_4 - J_1$.
Step5: Repeat the above process, insert job fourth, which is J_2 in the schedule. It can be inserted by four ways $J_2 - J_3 - J_4 - J_1$, $J_3 - J_2 - J_4 - J_1$, $J_3 - J_4 - J_2 - J_1$ and $J_3 - J_4 - J_1 - J_2$ by calculating the make span we get,

Schedule	**Makespan**
$J_2 - J_3 - J_4 - J_1$	39
$J_3 - J_2 - J_4 - J_1$	**38**
$J_3 - J_4 - J_2 - J_1$	**38**
$J_3 - J_4 - J_1 - J_2$	**38**

Table 13: Makespan for inserting J_2

Here we found the minimum makespan 38 corresponding to three sub sequences. We will take all the three schedule one by one starting from schedule $J_3 - J_2 - J_4 - J_1$.
Insert remaining job J5 in the selected schedule, we get the five possibilities $J_5 - J_3 - J_2 - J_4 - J_1$, $J_3 - J_5 - J_2 - J_4 - J_1$, $J_3 - J_2 - J_5 - J_4 - J_1$, $J_3 - J_2 - J_4 - J_5 - J_1$ and $J_3 - J_2 - J_4 - J_1 - J_5$. The makespan is given in table 14.

Schedule	**Makespan**
$J_5 - J_3 - J_2 - J_4 - J_1$	**42**
$J_3 - J_5 - J_2 - J_4 - J_1$	**42**
$J_3 - J_2 - J_5 - J_4 - J_1$	45
$J_3 - J_2 - J_4 - J_5 - J_1$	47
$J_3 - J_2 - J_4 - J_1 - J_5$	45

Table 14: Makespan for inserting J_5

Here the minimum makespan is 42 corresponding to two sequences. By the same process, inserting J_5 in the schedule $J_3 - J_4 - J_2 - J_1$ and $J_3 - J_4 - J_1 - J_2$, we get the following results in table 15 and table 16;

Schedule	**Makespan**
$J_5 - J_3 - J_4 - J_2 - J_1$	46
$J_3 - J_5 - J_4 - J_2 - J_1$	**42**
$J_3 - J_4 - J_5 - J_2 - J_1$	**42**
$J_3 - J_4 - J_2 - J_5 - J_1$	44
$J_3 - J_4 - J_2 - J_1 - J_5$	45

Table 15: Makespan for inserting J_5 in $J_3 - J_4 - J_2 - J_1$

Schedule	**Makespan**
$J_5 - J_3 - J_4 - J_1 - J_2$	44
$J_3 - J_5 - J_4 - J_1 - J_2$	**41**
$J_3 - J_4 - J_5 - J_1 - J_2$	42
$J_3 - J_4 - J_1 - J_5 - J_2$	46
$J_3 - J_4 - J_1 - J_2 - J_5$	42

Table 16: Makespan for inserting J_5 in $J_3 - J_4 - J_1 - J_1$

By comparing all the sequences in step 5, we get the minimum makespan 41 corresponding to the schedule $J_3 - J_5 - J_4 - J_1 - J_2$.

5. Solution by the CDS method

CDS (Campbell Dudek Smith) (1970) [2] is another heuristic method designed for solving more than two machines in flowshop problems. It extended the Johnson's method from two machines to more than two machines by dividing the m-machines into the series of two $(m-1)$ machines. The target of this method is to find the minimum make-span.
Procedure to apply CDS algorithm:
Step1: Convert the problem into two-machines problems:

- For an m-machine problem, create m-1 two machine problem. Here the iteration will be $m-1$ in number.
- Each $m-1$ two machine problems will obtain by combining the processing time of original machines. Processing time of 1^{st} machine will be the sum of first k original machine and processing time of 2^{nd} machine will be the sum of remaining $m-k$ machines.

Step2: Application of Johnson's algorithm:

- Solve each $m-1$ two machines problem by Johnson's Algorithm.
- Apply Johnson's rule, find the minimum processing time of each two machines problem. If minimum corresponds to first machine, then place that machine to the first of optimal schedule, if the minimum corresponds to second machine, then place that machine to the last in the optimal schedule.
- Repeat the process until all the jobs take place in the optimal schedule.

Step3: Evaluate the make-span:

- Calculate the makespan for the optimal schedule obtained from each $m-1$ machines problem.

Step4: Selection of the best schedule:
The optimal schedule which has the minimum makespan will be the best schedule for the original problem.
Solution: From table 4 we get;

$Jobs$	$\tilde{K}$	$\tilde{L}$	$\tilde{M}$
1	7	7	5
2	8	3	7
3	5	5	9
4	6	7	7
5	4	9	3

Table 17: Operating Time on machines

Step1: Convert the problem into two-machines problems. Since we have three machines so there are $3 - 1 = 2$ subsequence. which are in following tables,

$Jobs$	$\tilde{K}$	$\tilde{M}$
1	7	5
2	8	7
3	5	9
4	6	7
5	4	3

Table 18: first Subsequence

and

$Jobs$	$\tilde{K} + \tilde{L}$	$\tilde{L} + \tilde{M}$
1	14	12
2	11	10
3	10	14
4	13	14
5	13	12

Table 19: Second subsequence

Step2: Apply the Johnson's algorithm to each of the subsequence we get the optimal schedule

J_3	J_4	J_2	J_1	J_5

For first subsequence and two optimal schedules for second subsequence which are;

J_3	J_4	J_5	J_1	J_2

and

J_3	J_4	J_1	J_5	J_2

Step3. Now evaluate the make-span for optimal schedule obtained in step 2 we get;

Schedule	**Makespan**
$J_3 - J_4 - J_2 - J_1 - J_5$	45
$J_3 - J_4 - J_5 - J_1 - J_2$	**42**
$J_3 - J_4 - J_1 - J_5 - J_2$	46

Table 20: Makespan

Step4: Since the minimum makespan is 42 which corresponds to the optimal sequence $J_3 - J_4 - J_5 - J_1 - J_2$. This is the best schedule by CDS algorithm.

Problem2: let us assume another problem with same condition, we will take it from the initial table after using VAM and MODI method.

Jobs	X	Y	Z
1	30	10	25
2	20	20	20
3	06	40	20
4	60	14	14

Table 21: Problem 2

Here we found the minimum makespan by B&B is 139, by NEH method 184 and by CDS method 144.

Problem3: let us consider another problem,

Jobs	X	Y	Z
1	16	11	15
2	18	13	19
3	12	20	15
4	14	15	16
5	10	19	16

Table 22: Problem 3

Minimum makespan by B&B is 105, by NEH 110 and by CDS 117.

Problem4: let us try on a small problem:

Jobs	P	Q	R
1	3	2	2
2	2	1	4
3	4	3	1

Table 23: Problem 4

Here the Minimum makespan by B&B is 13, by NEH 15 and by CDS 13. Table 24 shows the comparison between all the problems

No. of jobs	**B&B**	**NEH**	**CDS**
5	38	41	42
5	105	110	117
4	139	184	144
3	13	15	13

Table 24: Comparison Table

6. Comparison between B&B, NEH, CDS

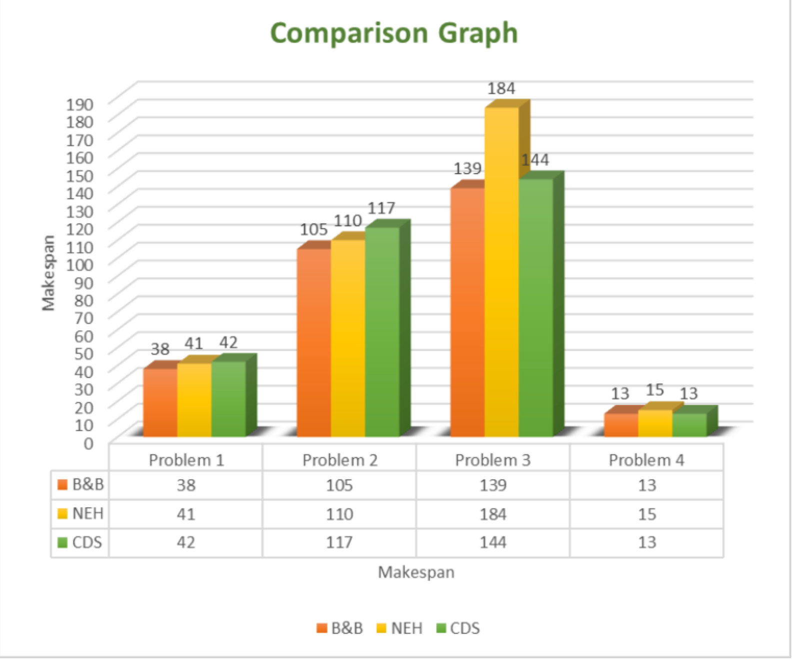

Figure 1: Comparison Graph

7. Conclusion

In this study, we observe (from table 24) that the Branch & Bound is exact and assurance finding optimal solution which gives the minimum makespan most of the time as compare to NEH and CDS heuristic method. Although, the choice of method depends on the size and specific requirement of problem. The heuristic methods are easy to understand and implement but does not guarantee the optimal solution. B&B is the best method in this study. This study can be extended by taking more characteristics such as no-idle time, job block criterion etc.

References

1. Allahverdi, *A survey of scheduling problems with setup times or costs*, European Journal of Operational Research, **187**, 985–1032,(2008).
2. Campbell, G., Herbert, A. Richard, Dudek, and L. Smith Milton, *A heuristic algorithm for the n job, m machine sequencing problem*, Management Science, **16**, B.630,(1970).
3. Gupta, Arun, and Sant Chauhan, *A heuristic algorithm for scheduling in a flow shop environment to minimize makespan*, International Journal of Industrial Engineering Computations, **6**, 173–184,(2015).
4. Gupta, Deepak, and Sonia Goel, *Three stage flow shop scheduling model with m-equipotential machines*, International Journal on Future Revolution in Computer Science & Communication Engineering, **4**, 269–274, (2018).
5. Gupta, Deepak, and Sonia Goel, *N×2 flow shop scheduling problem with parallel machines at every stage, processing time associated with probabilities*, Adv Math Sci J. **9**, 1061–1069, (2020).
6. Ignall, Edward, and Linus Schrage, *Application of the branch and bound technique to some flow-shop scheduling problems*, Operations Research, **13**, 400–412, (1965).
7. Ismail, E. Janakiriiand, A. Mohamed, and Sini Rahuman, *Utilising a dynamic approach to reduce tardiness for scheduling issues with distinct due dates and job blocking technique to reduce cost of tardiness and early arrival*, Journal of Theoretical and Applied Information Technology, **101**, 6165–6177, (2023).
8. Lee, Chung-Yee, *Parallel machine scheduling with no simultaneous machine available time*, Discrete Applied Mathematics, **30**, 53–61, (1991).
9. Malhotra, Khushboo, *Comparison of Bb with Meta-heuristic Approach in Optimization of Three Stage Fss with Multiple Processors*, **1**(2023).
10. Palmer, S. Douglas, *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**, 101–107, (1965).
11. Sauvey, Christophe, and Nathalie Sauer, *Two NEH heuristic improvements for flowshop scheduling problem with makespan criterion*, Algorithms, **13**, 112, (2020).
12. Smith, D. Richard, and A. Richard Dudek, *A General Algorithm for Solution of the N-Job, M-Machine Sequencing Problem of the Flow Shop*, Operations Research, **15**, 71–82, (1967).
13. S. M. Johnson, *Optimal two-and three-stage production schedules with setup times included*, Naval Research Logistics Quarterly, **1**, 61–68, (1954).
14. Yeh, Wei-Chang, and Ali Allahverdi, *A branch-and-bound algorithm for the three-machine flow shop scheduling problem with bicriteria of makespan and total flowtime*, International Transactions in Operational Research, **11**, 323–339, (2004).
15. Y. Feng, and J. Kong, *Multi-Objective Hybrid Flow-Shop Scheduling in Parallel Sequential Mode While Considering Handling Time and Setup Time*, Appl. Sci., **13**(2023), 3563, (2023).

Sonia,
Department of Mathematics,
Maharishi Markandeshwar Engineering College,
Maharishi Markandeshwar (Deemed to be University),
Mullana (Ambala), Haryana, India
E-mail address: dhimansonia13@gmail.com, ORCID: 0009-0006-7262-4762

and

Sonia Goel,
Department of Mathematics,
Maharishi Markandeshwar Engineering College,
Maharishi Markandeshwar (Deemed to be University),

Mullana (Ambala), Haryana, India
E-mail address: sonia.mangla14@gmail.com, ORCID: 0000-0003-1211-6003

and

Deepak Gupta,
Department of Mathematics,
Maharishi Markandeshwar Engineering College,
Maharishi Markandeshwar (Deemed to be University),
Mullana (Ambala), Haryana, India
E-mail address: guptadeepak@yahoo.co.in, ORCID: 0000-0002-9461-8770