BI STAGE FSSP WITH TRIANGULAR INTUITIONISTIC FUZZY NUMBER AND COMPARISON BETWEEN HEURISTICS

Pooja Kaushik[1*], Sonia[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), Haryana, India
[1*]drtobekaushik2024@gmail.com, [2]dhimansonia13@gmail.com, [3]guptadeepak@yahoo.co.in, [4*]sonia.mangla14@gmail.com

Abstract

Scheduling theory focuses on the best way to distribute limited resources among tasks throughout time and allocating limited resources to tasks in the best possible way. This field's history begins when two people first fought over a resource and devised a strategy to divide it without resorting to violence. In real world applications, scheduling is a vital yet highly intricate issue. In industrial processes, managing production with the goal of reducing makespan is an essential duty. The processing time of each job on each machine has been taken to be a real number for the majority of scheduling problems studied already. However, processing times in real-world applications are frequently uncertain, meaning that human error or operational flaws may cause them to fluctuate easily. This study explores a two-stage FSSP, where parallel machines execute jobs in the both stages. The processing times in the flow shop scheduling are represented by fuzzy numbers in this work. This paper gives comparative study to address the FSSP under an uncertain environment based on triangular intuitionistic fuzzy numbers (TrIFNs). To accurately model uncertainty and imprecision in processing times, we employ TIFNs, which provide a more comprehensive representation of vagueness compared to traditional fuzzy sets. To find the best work sequence with the goal of minimizing the makespan, fuzzy and scheduling principles are applied to the flow shop scheduling challenges. The main goal is to optimize the flow time by comparing the efficacy of three distinct optimization approaches. A rigorous computational analysis is conducted to evaluate their performance, leveraging fuzzy arithmetic operations and defuzzification techniques. The three heuristics' performances are compared and their performance is analyzed. An analysis of the performance of three heuristics i.e. B&B, Dannenbring exhaustive search and the division algorithm proposed by santhi in previous paper has been conducted. The findings offer valuable insights into the applicability of fuzzy scheduling methodologies in complex manufacturing environments, contributing to the broader discourse on uncertainty modelling in production systems. The comparison is shown by a numerical illustration. This study provides a comparison and evaluation of the scheduling problem which also helps the reader comprehend various environmental assumptions, system limitations, and useful heuristics for further research.

Keywords: flow time (makespan), triangular intuitionistic fuzzy number, flow shop scheduling, transportation technique, B&B, Dannenbring exhaustive search

I. Introduction

Scheduling is a fundamental decision-making process in operation research and industrial management, where tasks or jobs are assigned to resources such as machines, workers, or computers over a specified time horizon. The primary goal of scheduling is to optimize performance measures such as minimizing makespan, reducing job tardiness, improving resource utilization, and balancing workloads. Scheduling problems are encountered in various domains, including manufacturing, logistics, healthcare, and computing, making it a critical area of study. Depending on the nature of the system, scheduling problems can be categorized into single-machine, parallel-machine, job shop, open shop, and flow shop scheduling, each presenting unique challenges and optimization criteria. Among these, flow shop scheduling is a widely studied class of scheduling problems where jobs follow the same sequence of machines. In a typical flow shop, a set of jobs is processed across multiple machines in the same order, ensuring a structured workflow. The classical permutation FSSP restricts job sequences to remain unchanged across all machines, making it a constrained but well-defined problem. The objective in flow shop scheduling often revolves around minimizing total flow time, reducing machine idle time, or improving energy efficiency.

The two-machine FSSP was first introduced [1], which provided an optimal sequencing rule to minimize makespan. However, as the number of machines and complexity increase, the problem becomes NP-hard, requiring heuristic and metaheuristic approaches for efficient solutions. This flexibility increases practical applicability in real-world industries, such as automobile manufacturing, semiconductor production, and textile industries. Traditional deterministic models assumed fixed processing times, but uncertainties in real-world systems led to the adoption of fuzzy set theory [2]. This study considered uncertainties in real-life scheduling scenarios which led to the development of fuzzy FSSP, where processing times are represented as fuzzy numbers rather than deterministic values. This approach allows for handling imprecision and variability in job execution times, making scheduling more robust against disruptions.

FSSPs have been widely studied due to their applications in manufacturing and production systems. Work on two-stage FSSP was done[3] and later optimized[4] providing an optimal sequencing rule for makespan minimization. This work on sequencing and scheduling laid the foundation for understanding how to efficiently allocate tasks over time while optimizing various performance measures and focused on job sequencing (determining the order of tasks) and scheduling (assigning tasks to resources over time) in manufacturing and service systems. The study [5] provides an early evaluation of flow shop sequencing heuristics. This foundational work introduced and assessed heuristic techniques, which have influenced modern scheduling methods. The study's insights remain relevant as a benchmark for contemporary heuristic development.

The work [6] introduced a method to rank fuzzy subsets which is useful in scenarios where decision making involves imprecise or uncertain information and presented a method for ranking fuzzy subsets of the unit interval. Since fuzzy sets lack a strict ordering due to their membership functions, traditional ranking methods are inadequate. Yager introduced an approach based on integral and expectation-like computations to assign a numerical index to fuzzy subsets, facilitating comparison. However, the study [7] proposed intuitionistic fuzzy sets (IFSs) incorporating both membership and non-membership degrees, making them more effective in handling hesitation in scheduling. The study [8] addressed scheduling challenges in two machine flow shop where job processing times are uncertain and represented them as triangular fuzzy number.

The work [9] explored the problem of FSSP with two machines and a single transport facility under a fuzzy environment and focused on optimizing job scheduling while considering uncertain setup times, which are modeled using fuzzy logic. Unlike traditional approaches that assume fixed setup times, this method incorporates fuzzy numbers to handle variability, making the model more applicable to real-world industrial settings. The extension of the Dannenbring method to optimize flow shop-type production while considering job waiting times and weights has been considered [10]. Their study suggests modifications or enhancements to the original heuristic, making it more applicable to modern production environments. A novel method incorporating triangular intuitionistic fuzzy numbers to address scheduling uncertainties was proposed [11]. This approach aligns with recent trends in fuzzy logic applications, aiming to enhance decision-making in complex, uncertain scheduling environments.

Also, the study [12] proposed an optimization model for a two-stage FSSP with equipotential machines at the first stage. Their approach likely incorporates mathematical optimization or heuristic techniques to enhance scheduling efficiency. The researchers [13] presented B&B technique for solving two-stage FSSPs. The B&B method is a well-known exact optimization approach, and this study extends its application to scenarios with equipotential machines, ensuring efficient job allocation across stages. The work [14] compared heuristic methods, particularly NEH (Nawaz-Enscore-Ham) and CDS (Campbell-Dudek-Smith), within a fuzzy environment and assessed their performance in bi-stage flow shop scheduling, highlighting the trade-offs between computational efficiency and optimality in uncertain conditions.

II.Preliminaries

I.Fuzzy Set

A fuzzy set, a mathematical representation of a set that allows for vagueness and uncertainty, is defined in equation (1).

$$\chi = \{(y, \mu_{\chi}(y)) \mid y \in Y\} \tag{1}$$

where χ is the fuzzy set, Y is the set of all possible elements, $\mu_{\chi}(y)$ is the membership function, which assigns a degree of membership to each element y in Y.

II.Fuzzy Number

A fuzzy number is a mathematical representation of a numerical value that is uncertain or imprecise, is defined as in equation (2):

$$\chi = \{(y, \mu_{\chi}(y)) \mid y \in \mathbb{R}\} \tag{2}$$

where, χ is the fuzzy number, $\mathbb{R}$ is the set of real numbers, $\mu_{\chi}(y)$is membership function, which assigns a degree of membership to each real number y.

III. Triangular Fuzzy Number

A fuzzy number Å = (à, ß, ç) is called triangular fuzzy number (graphically shown in figure 1)
if its membership function satisfy the following conditions

- à to ß is an increasing function
- ß to ç is decreasing function
- à≤ ß≤ ç

and is given by equation (3).

$$\mu_{\text{Å}}(x)=\begin{cases} 0 & for x < \text{à} \\ \frac{x-\text{à}}{\text{ß}-\text{à}} for \text{ à} \leq x \leq \text{ß} \\ \frac{\text{ç}-x}{\text{ç}-\text{ß}} for \text{ ß} \leq x \leq \text{ç} \\ 0 & for x > \text{ç} \end{cases} \tag{3}$$

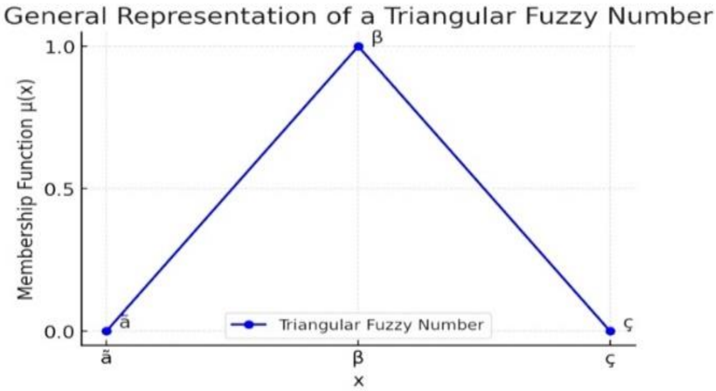


Figure 1: *Graph of triangular fuzzy number*

IV. Intuitionistic Fuzzy Number (IFN)

An IFN is a mathematical representation of a numerical value that is uncertain or imprecise, and also takes into account the degree of hesitation or uncertainty. It is defined as:

$$\chi = \{(y, \mu_\chi(y), \nu_\chi(y)) \mid y \in \mathbb{R}\} \tag{4}$$

where χ is the intuitionistic fuzzy number, $\mathbb{R}$ is the set of real numbers, $\mu_\chi(y)$ is membership and $\nu_\chi(y)$is non-membership function assigning a degree of membership and non-membership to each real number y respectively.

V. Triangular Intuitionistic Fuzzy Number (TrIFN)

An IFN G = $(g_1, g_2, g_3)(\acute{g}_1, g_2, \acute{g}_3)$ is said to be TrIFN if itsmembership and the non-membership function are given in equation (5) and graphically presented in figure 2.

$$\mu_G(y) = \begin{cases} \frac{y-g_1}{g_2-g_1}, & if\, g_1 \le y \le g_2 \\ \frac{g_3-y}{g_3-g_2}, & if\, g_2 \le y \le g_3 \\ 0, & otherwise \end{cases}$$

$$\nu_G(y) = \begin{cases} \frac{g_2-y}{g_2-\acute{g}_1}, & if\, \acute{g}_1 \le y \le g_2 \\ \frac{y-g_2}{\acute{g}_3-g_2}, & if\, g_2 \le y \le \acute{g}_3 \\ 1, & otherwise \end{cases} \tag{5}$$

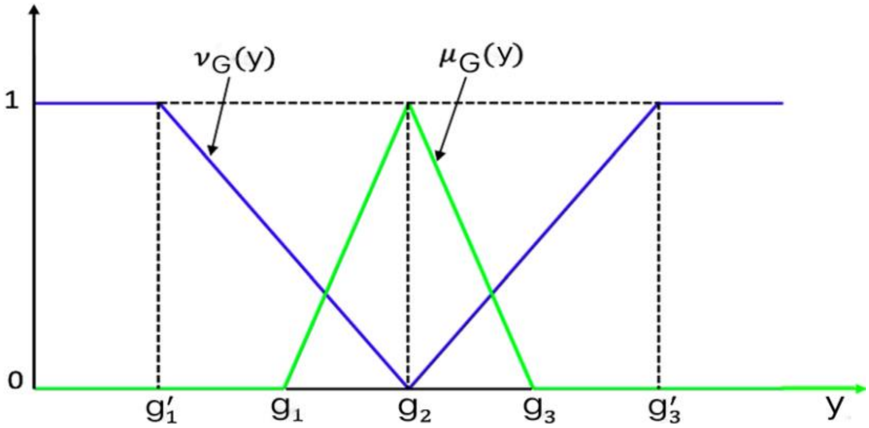


Figure 2: *Graph of triangular Intuitionistic fuzzy number*

VI.Arithmetic operators

Let G = (g_1, g_2, g_3);$(\acute{g}_1, g_2, \acute{g}_3)$ and H= (h_1, h_2, h_3);$(\acute{h}_1, h_2, \acute{h}_3)$ be two TrIFNs then

- Addition Let G+H = $(g_1 + h_1, g_2 + h_2, g_3 + h_3)$ $(\acute{g}_1 + \acute{h}_1, g_2 + h_2, \acute{g}_3 + \acute{h}_3)$
- Subtraction Let G-H = $(g_1 - h_1, g_2 - h_2, g_3 - h_3)$ $(\acute{g}_1 - \acute{h}_1, g_2 - h_2, \acute{g}_3 - \acute{h}_3)$

VII.Ranking of TrIFN

The ranking of TrIFN is defined as in equation (6)

$$R(A) = \frac{1}{3}[\frac{(\acute{g}_3 - g_1)(g_2 - 2\acute{g}_3 - 2\acute{g}_1) + (g_3 - g_1)(g_1 + g_2 + g_3) + 3(\acute{g}_3{}^2 - \acute{g}_1{}^2)}{(\acute{g}_3 - \acute{g}_1 + g_3 - g_1)} \quad (6)$$

If $R.G. \leq R.H.$, then $G \leq H$.

III. Problem Formulation

Assume that three parallel equipotential machines are to be used for both stages (i=1,2,3……, n; j=1,2,3……, m). Every job's processing time is taken as TrIFN. The available time and unit operating cost of each parallel machine are also provided. The following is the problem's model in matrix form, represented in table 1.

Table 1: *Mathematical model of the problem*

Job	Processor $\hat{U}$					Processor $\hat{V}$				
i	$\hat{U}_1$	$\hat{U}_2$	$\hat{U}_3$	$\hat{U}_n$	Processing time of$\hat{U}$	$\hat{V}_1$	$\hat{V}_2$	$\hat{V}_3$	$\hat{V}_m$	Processing time of $\hat{V}$
1	u_{11}	u_{12}	u_{13}	u_{1n}	(c_1, d_1, e_1);($\hat{c}_1, d_1, \hat{e}_1)$	v_{11}	v_{12}	v_{13}	v_{1n}	(g_1, h_1, k_1);($\hat{g}_1, h_1, \hat{k}_1)$
2	u_{21}	u_{22}	u_{23}	u_{2n}	(c_2, d_2, e_2);($\hat{c}_2, d_2, \hat{e}_2)$	v_{21}	v_{22}	v_{23}	v_{2n}	(g_2, h_2, k_2); $(\hat{g}_2, h_2, \hat{k}_2)$
3	u_{31}	u_{32}	u_{33}	u_{3n}	(c_3, d_3, e_3); $(\hat{c}_3, d_3, \hat{e}_3)$	v_{31}	v_{32}	v_{33}	v_{3n}	(g_3, h_3, k_3); $(\hat{g}_3, h_3, \hat{k}_3)$
4	u_{41}	u_{42}	u_{43}	u_{4n}	(c_4, d_4, e_4); $(\hat{c}_4, \hat{d}_4, \hat{e}_4)$	v_{41}	v_{42}	v_{43}	v_{4n}	(g_4, h_4, k_4); $(\hat{g}_4, h_4, \hat{k}_4)$
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
n	u_{n1}	u_{n2}	u_{n3}	u_{nn}	(c_n, d_n, e_n); $(\hat{c}_n, d_n, \hat{e}_n)$	v_{n1}	v_{n1}	v_{n1}	v_{nm}	(g_n, h_n, k_n); $(\hat{g}_n, h_n, \hat{k}_n)$
t_{pj}	t_{11}	t_{12}	t_{13}	t_{1n}		t_{21}	t_{22}	t_{23}	t_{2m}	

I. Notations

(c_n, d_n, e_n);$(\hat{c}_n, d_n, \hat{e}_n)$: Processing time of n^{th} job on processor $\hat{U}$
(g_n, h_n, k_n);$(\hat{g}_n, h_n, \hat{k}_n)$: Processing time of n^{th} job on processor $\hat{V}$

$\hat{U}_n$: n equipotential machines of processor $\hat{U}$
$\hat{V}_m$: n equipotential machines of processor $\hat{V}$
t_{pj}: Transportation time of moving jobs from machine A to B
u_{ij}: unit cost of i^{th} job on $\hat{U}_j^{th}$ parallel machine
v_{ij}: unit cost of i^{th} job on $\hat{V}_j^{th}$ parallel machine

II. Assumptions

- Each device operates the same set of tasks.
- Each job is processed on machine B once it has been finished on machine A.
- All machines can start simultaneously.
- The operational costs of each type A and type B machine vary.
- Machines are always operational and never experience downtime or failures, ensuring continuous availability for scheduling.
- Once a task begins, it must be finished without interruption.

III. Algorithm

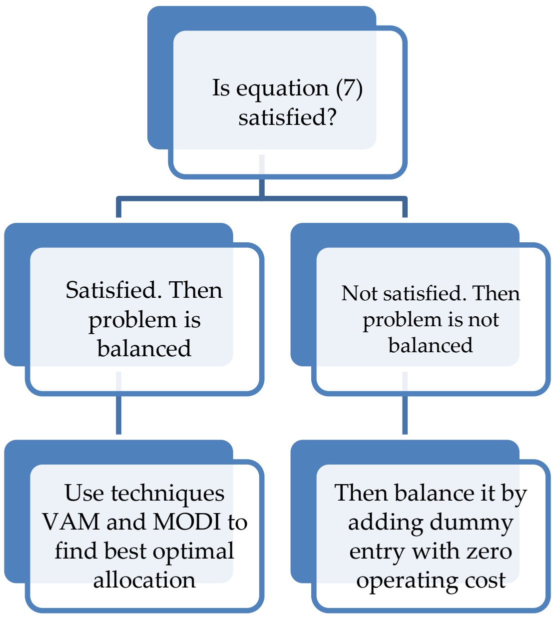
The following course of action can be used to solve the FSSP given in table 1. Using the ranking formula, we convert the triangular intuitionistic fuzzy utilisation time into a crisp value in both phases. Next, in order to determine the best allocation, we use transportation techniques such as VAM. The MODI method can then be used to optimize the working time of each job on the parallel equipotential machines and to verify that the solution reached is optimal. The best sequence is then determined using B&B.

Step 1: Use ranking rule given in equation (6) to defuzzify the prosessing time taken as TrIFNs into crisp values.

Step 2: Apply transportation technique like VAM and MODI. To apply any transportation technique, we examine the constraint

$$\sum_{j=1}^{3} t_{1j} = \sum_{i=1}^{n} d_i \quad (7)$$

To check that the problem is balanced or not. Here come two situations:



Step 3: Use branch and bound method by applying the rule

$$g^{'}=\max(\sum_{i=1}^{n} U_{ij}\}_{j=1,2..m}+\min_{i\in J_k}\{\max_{j=1,2..p} V_{ij})\&g^{''}=\max_{i\in J_k}\{U_{ij}\}_{j=1,2..m}+\max\{\sum_{i}^{n} V_{ij}\}_{j=1,2..p}$$

Then evaluate $G = \max(g^{'}, g^{''})$ for all the jobs.

Find the minimum of all values of G and start from that vertex as the first job in the optimal subsequence. Repeat the above process till we reach termination point of branch which will give us the most ideal optimal sequence of jobs.

IV.Numerical problem

Three machines $\hat{U}_1, \hat{U}_2$ $and \hat{U}_3$ of type $\hat{U}$ and three machines $\hat{V}_1, \hat{V}_2$and $\hat{V}_3$ of type $\hat{V}$ are accessible for various amounts of time and have differing running expenses, as presented in table 2.

Table 2: *Numerical problem*

Job	Processor $\hat{U}$				Processor $\hat{V}$			
i	$\hat{U}_1$	$\hat{U}_2$	$\hat{U}_3$	Processing time of $\hat{U}$	$\hat{V}_1$	$\hat{V}_2$	$\hat{V}_3$	Processing time of $\hat{V}$
1	12	15	13	(10,11,12) ;(8,11,13)	11	12	8	(6,8,9); (5,7,10)
2	9	11	10	(9,10,11) ;(8,10,12)	9	11	7	(10,11,12);(8,11,13)
3	9	7	11	(11,12,13);(10,12,15)	7	9	10	(8,9,11); (7,9,12)
4	13	8	8	(11,12,13);(10,12,15)	15	12	13	(6,7,8); (5,7,9)
5	10	12	9	(12,14,15);(11,14,16)	10	9	12	(8,9,10); (7,9,11)
Available time	19	21.5	18.1		15	12.7	16	

Solution:

Step 1: Changing the TrIFN into crisp value using the ranking formula given in equation (6)

(10,11,12) ;(8,11,13) changed into 10.7 In similar fashion (6,8,9); (5,7,10) changed into 7.5

(9,10,11) ;(8,10,12) changed into 10 (10,11,12);(8,11,13) changed into 10.8

(11,12,13);(10,12,15) changed into 12 (8,9,11); (7,9,12) changed into 9.4

(11,12,13);(10,12,15) changed into 12.2 (6,7,8); (5,7,9) changed into 7

(12,14,15);(11,14,16) changed into 13.7 (8,9,10); (7,9,11) changed into 9

and the changed crisp values are given in table 3.

Table 3: *Reduced problem after defuzzifying TrIFN into crisp value*

Job	Processor $\hat{U}$				Processor $\hat{V}$			
i	$\hat{U}_1$	$\hat{U}_2$	$\hat{U}_3$	Processing time of $\hat{U}$	$\hat{V}_1$	$\hat{V}_2$	$\hat{V}_3$	Processing time of $\hat{V}$
1	12	15	13	10.7	11	12	8	7.5
2	9	11	10	10	9	11	7	10.8
3	9	7	11	12	7	9	10	9.4
4	13	8	8	12.2	15	12	13	7
5	10	12	9	13.7	10	9	12	9
Available time	19	21.5	18.1		15	12.7	16	

Step 2: We determine the best way to distribute processing time by applying transportation technique VAM first and then MODI which is given in table 4&5.

Table 4 :*Optimal distribution of processing time on machine* $\hat{U}_1, \hat{U}_2$ *and* $\hat{U}_3$

i	$\hat{U}_1$	$\hat{U}_2$	$\hat{U}_3$
1	9	0	1.7
2	10	0	0
3	0	12	0
4	0	9.5	2.7
5	0	0	13.7

Table 5: *Optimal distribution of processing time on machine* $\hat{V}_1, \hat{V}_2$*and* $\hat{V}_3$

i	$\hat{V}_1$	$\hat{V}_2$	$\hat{V}_3$
1	0	0	7.5
2	2.3	0	8.5
3	9.4	0	0
4	0	7	0
5	3.3	5.7	0

After applying transportation techniques VAM and MODI, the time is allocated on machines and the reduced problem is given in table 6.

Table 6: *Reduced problem's tabular representation*

Job	Processor $\hat{U}$				Processor $\hat{V}$			
i	$\hat{U}_1$	$\hat{U}_2$	$\hat{U}_3$	Processing time of $\hat{U}$	$\hat{V}_1$	$\hat{V}_2$	$\hat{V}_3$	Processing time of $\hat{V}$
1	9	0	1.7	10.7	0	0	7.5	7.5
2	10	0	0	10	2.3	0	8.5	10.8
3	0	12	0	12	9.4	0	0	9.4
4	0	9.5	2.7	12.2	0	7	0	7
5	0	0	13.7	13.7	3.3	5.7	0	9

Now we apply Branch and bound (B&B) algorithm and calculate the lower bound of jobs to obtain the sequence of jobs for optimal solution. Branch and bound algorithm is applied to further optimize the allocation obtained after the application of VAM and MODI.

Table 7: *Depicting lower bound of jobs*

(i)	$g^{'}$ =max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in J_k}(p_i)$	$g^{''}$ = $\max_{i\in J_k} K_{ij}$+$\sum_{i}^{n} p_i$	G=max $\{g^{'}, g^{''}\}$
1	max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in\{2,3,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{2,3,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =9+16=25	27.2
2	max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in\{1,3,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{1,3,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =10+16=25	27.2
3	max $(\sum_{i=1}^{n} K_{ij})$+ $\min_{i\in\{1,2,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{1,2,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =12+16=25	28
4	max $(\sum_{i=1}^{n} K_{ij})$+ $\min_{i\in\{1,2,3\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{1,2,3\}} K_{ij}$+$\sum_{i}^{n} p_i$ =9.5+16=25	27.2
5	max $(\sum_{i=1}^{n} K_{ij})$+ $\min_{i\in\{1,2,3\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{1,2,3\}} K_{ij}$+$\sum_{i}^{n} p_i$ =13.7+16=25	29.7

Hence from table 7, lower bound is 27.2 corresponding to job 1,2 and 4.Without loss of generality let us fix job 1 at 1^{st} place. Now we continue the same process and determine the nodes for the second branch.This process is continued till all jobs gets place in the optimal order.

Table 8: *Depicting lower bound of jobs when job 1 is fixed at first place*

(i)	$g^{'}$ =max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in J_k}(p_i)$	$g^{''}$ = $\max_{i\in J_k} K_{ij}$+$\sum_{i}^{n} p_i$	G=max $\{g^{'}, g^{''}\}$
12	max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in\{2,3,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{2,3,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =27.5+16=43.5	43.5
13	max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in\{1,3,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{1,3,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =21.4+16=37.4	37.4
14	max $(\sum_{i=1}^{n} K_{ij})$+ $\min_{i\in\{1,2,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{1,2,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =16.5+16=32.5	32.5
15	max $(\sum_{i=1}^{n} K_{ij})$+ $\min_{i\in\{1,2,3\}}(p_i)$ =21.5+7=28.5	$\max_{i\in\{1,2,3\}} K_{ij}$+$\sum_{i}^{n} p_i$ =21.1+16=37.1	37.1

Lower bound comes to be 32.5 as shown in table 8 which is corresponding to subsequence 14. Therefore job 4 is fixed at2^{nd} place. We continue searching for the next task in the ideal order.

Table 9: *Depicting lower bound of jobs when job 1 is fixed at first place and 4 at second place*

(i)	$g^{'}$ =max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in J_k}(p_i)$	$g^{''}$ = $\max_{i\in J_k} K_{ij}$+$\sum_{i}^{n} p_i$	G=max $\{g^{'}, g^{''}\}$
142	max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in\{2,3,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{2,3,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =18.5+16=34.5	34.5
143	max $(\sum_{i=1}^{n} K_{ij})$ + $\min_{i\in\{1,3,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{1,3,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =30.9+16=46.9	46.9
145	max $(\sum_{i=1}^{n} K_{ij})$+ $\min_{i\in\{1,2,4\}}(p_i)$ =21.5+7=28.5	$\max_{i\in\{1,2,4\}} K_{ij}$+$\sum_{i}^{n} p_i$ =23.8+16=39.8	39.8

Lower bound comes to be 34.5 which is corresponding to subsequence 142 given in table 9. Therefore, job 2 is fixed at third place. Now proceeding in the similar fashion to find the next job.

Table 10: *Depicting lower bound of jobs when job 1,4 and 2 is fixed at first, second and third placerespectively.*

(i)	$g' = \max(\sum_{i=1}^{n} K_{ij}) + \min_{i\in J_k}(p_i)$	$g'' = \max_{i\in J_k} K_{ij} + \sum_i^n p_i$	G=max $\{g', g''\}$
1423	$\max(\sum_{i=1}^{n} K_{ij}) + \min_{i\in\{2,3,4\}}(p_i)$ =21.5+5.7=27.2	$\max_{i\in\{2,3,4\}} K_{ij} + \sum_i^n p_i$ =21.4+16=37.4	37.4
1425	$\max(\sum_{i=1}^{n} K_{ij}) + \min_{i\in\{1,3,4\}}(p_i)$ =21.5+9.4=30.9	$\max_{i\in\{1,3,4\}} K_{ij} + \sum_i^n p_i$ =19.4+16=35.4	35.4

From table 10, min.{G}=35.4 which is related to the subsequence 1425. So first, second, third and fourth place in the required optimal sequence is acquired by job 1,4,2,5 respectively. Also, by default job 3 will take the last i.e. fifth place. Therefore {1,4,2,5,3} is the required optimal sequence. Best feasible solution is given in table 11 .

Table 11: *In-out table*

Job	Processor $\hat{U}$			Processor $\hat{V}$		
i	$\hat{U}_1$	$\hat{U}_2$	$\hat{U}_3$	$\hat{V}_1$	$\hat{V}_2$	$\hat{V}_3$
1	0-9	0	0-1.7	0	0	9-16.5
4	0	0-9.5	1.7-4.4	0	9.5-16.5	0
2	9-19	0	0	19-21.3	0	19-27.5
5	0	0	4.4-18.1	21.3-24.6	18.1-23.8	0
3	0	9.5-21.5	0	24.6-34	0	0

V. Gantt Chart

The solution obtained in table 11 is shown by Gantt chart in figure 3.

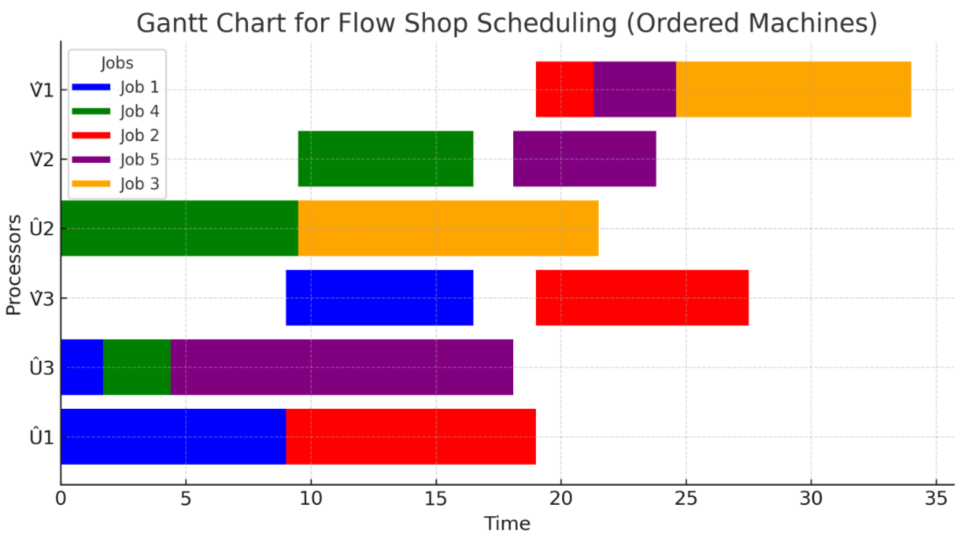


Figure 3: *Gantt chart*

The Gantt Chart as shown in figure 3, helps in scheduling, tracking progress, and ensuring efficient resource. It helps to define tasks, deadlines, resource allocation, organizes work in a logical sequence and tracks completed, ongoing, and pending tasks allocation.

Here

- Each row represents a task in the project.
- The horizontal axis shows time.
- The length of each bar indicates the task duration.
- Colour coding shows task completion and different colours indicating different jobs.

VI. The Dannenbring Method

Dannenbring Method is a heuristic approach for solving the FSSP. It was proposed by D.G. Dannenbring in 1977 as an improvement over traditional heuristics like Johnson's rule, aiming to minimize total time to complete all jobs. Johnson method (1954) served as the basis for the Dannenbring approach. Scheduling issues with more than two machines can be resolved using this approach [Panneerselvan 1999]. The method prioritizes job sequencing based on mean processing times and positional weight calculations. It is particularly useful for FSSPsincluding machines more than two where Johnson'srule may not be optimal.

The Dannenbring method (1977) gives the following formula in equation (8)

$$Þ_{i1} = \sum_{i=1}^{m}(m - i + 1) * t_{ij}$$
$$Þ_{i2} = \sum_{i=1}^{m} i * t_{ij} \quad (8)$$

Steps of the Dannenbring Method:

- Compute the average processing time of each job across all machines.
- Compute Positional Weights: Positional weight is calculated to balance job placement in the sequence, considering how a job affects the overall makespan.
- Rank Jobs Based on Weight & Priority Rules: Jobs with higher priority (based on mean time and weight) are placed in the sequence to optimize flow.
- Sequence Jobs Accordingly: The method assigns jobs in a way that balances bottleneck and reduces idle time between machines.

Solution:

First step in the Dannenbring approach is obtaining the expected values for every job given in table 12.

Table12: *Expected values of each job*

Job	Processor $\hat{U}$	Processor $\hat{V}$
1	10.7	7.5
2	10	8.5
3	12	9.4
4	12.2	7
5	13.7	5.7

Now, we find the values of $Þ_{i1}$ and $Þ_{i2}$.

Table13: *Positional weight of jobs*

Job	Processor $\widehat{U}$	Processor $\widehat{V}$
1	36.4	25.7
2	37	27
3	42.8	30.8
4	38.4	26.2
5	38.8	25.1

From table 13, the optimal order is 3-2-4-1-5.
Table 14 gives in-out table and optimal flow time.

Table14: *Makespan Value*

Job	Processor $\widehat{U}$			Processor $\widehat{V}$		
i	$\widehat{U}_1$	$\widehat{U}_2$	$\widehat{U}_3$	$\widehat{V}_1$	$\widehat{V}_2$	$\widehat{V}_3$
3	0	0-12	0	12-21.4	0	0
2	0-10	0	0	21.4-23.7	0	10-18.5
4	0	12-21.5	0-2.7	0	21.5-28.5	0
1	10-19	0	2.7-4.4	0	0	19-26.5
5	0	0	4.4-18.1	23.7-27	28.5-34.2	0

Therefore, makespan from the optimal sequence comes out to be 34.2

VII. Division Algorithm (Method proposed by Selvakumari and S. Santhi [11])

Step 1: Convert TrIFN into crisp value.
Step 2: Divide all entries of rows by row's highest value and place that value in the top right corner of each cell.
Step 3: Divide all the entries of the columns by column's highest value and place that value in the bottom left corner of each cell.
Step 4: Add both the numbers obtained in step 2&3 and write in a table.
Step 5: Choose the row corresponding to minimum value, mark that job as first job in optimal order and remove the row.
Step 6: Repeat the process and obtain the optimal order.

Solution: Defuzzified values are given the table 15.

Table15: *Problem after defuzzification*

Job	Processor $\widehat{U}$	Processor $\widehat{V}$
1	10.7	7.5
2	10	8.5
3	12	9.4
4	12.2	7
5	13.7	5.7

In table 15, divide all the entries of each row by the row's highest value and place that value in the top right corner of each cell.
Similarly, divide all the entries of the columns by column's highest value and place that value in

the bottom left corner of each cell.
Allocate these calculated effective values to each cell and place them as shown in table 16.

Table16: *Effective value assigned to jobs*

Job	Processor $\hat{U}$			Processor $\hat{V}$		
			1			0.7
1		10.7			7.5	
	0.78			0.79		
			1			0.85
2		10			8.5	
	0.73			0.90		
			1			0.78
3		12			9.4	
	0.87			1		
			1			0.57
4		12.2			7	
	0.89			0.74		
			1			0.42
5		13.7			5.7	
	1			0.61		

Add both the numbers obtained in the table 16. Then choose the row corresponding to minimum value, marking that job as first job in optimal order and remove the row. Repeat the process and obtain the optimal order.

Table17: *Total effective values of each job*

Job	Processor $\hat{U}$	Processor $\hat{V}$
1	1.78	1.49
2	1.73	1.75
3	1.87	1.78
4	1.89	1.31
5	2	1.03

In table 17, value in the fifth row is lowest. Therefore, delete the 5^{th} row and place 5th job at first place in optimal sequence as S_5. Proceeding in this manner, optimal order obtained is S_5>S_4>S_1>S_2>S_3 given in table 18.

Table 18: *In-Out table*

Job	Processor $\hat{U}$			Processor $\hat{V}$		
i	$\hat{U}_1$	$\hat{U}_2$	$\hat{U}_3$	$\hat{V}_1$	$\hat{V}_2$	$\hat{V}_3$
5	0	0	0-13.7	13.7-17	13.7-19.4	0
4	0	0-9.5	13.7-16.4	0	16.4-23.4	0
1	0-9	0	16.4-18.1	0	0	18.1-25.6
2	9-19	0	0	19-21.3	0	25.6-34.1
3	0	9.5-21.5	0	21.5-30.9	0	0

Therefore, minimum flow time is 34.1

VIII. Practical Situation

A pharmaceutical company is developing three new drugs (D1, D2, D3) for different medical conditions.Since processing times in both stages involve uncertainty (due to variations in formulation success rates, lab conditions, patient recruitment, and regulatory delays), they are modeled using Triangular Intuitionistic Fuzzy Numbers (TrIFNs). The development process consists of two main stages:

Stage 1: Drug Formulation & Initial Testing
The drug is formulated in the lab, and its chemical composition is tested. Processing times for different formulations are uncertain due to variability in raw materials, laboratory conditions, and expert evaluations. These uncertain processing times are modeled using Triangular Intuitionistic Fuzzy Numbers (TIFNs) to capture both hesitation and imprecision in expert assessments.

Stage 2: Clinical Trials & Regulatory Approval
Once initial tests are passed, the drug enters clinical trials, followed by regulatory approval. The duration of trials depends on patient responses, ethical approvals, and unexpected side effects, introducing further uncertainty in scheduling.
The aim is to optimizetotal time for all drugs to pass through both stages while considering uncertainty in processing times. Since Stage 2 depends on Stage 1 completion, delays in Stage 1 affect the overall timeline.

IX. Comparison between B&B, Dannenbring Method and Divison Algorithm

B&B is a systematic search method that explores solution spaces using a tree structure, pruning branches that cannot yield optimal solutions. This significantly reduces computational complexity and makes it suitable for large-scale problems like the Traveling Salesman Problem (TSP), knapsack, and job scheduling. In contrast, Dannenbring's Exhaustive Search evaluates all possible solutions without pruning, ensuring that the optimal solution is found but at the cost of high computational expense. This method becomes impractical for large problems due to its exponential time complexity and high memory requirements. Division algorithm offers a pragmatic solution by incorporating fuzzy logic to handle imprecise data,near optimal solutions suitable for complex and real world applications.

X. Future Scope

Future studies should integrate deep learning, reinforcement learning, and Type-2 fuzzy systems to improve large-scale scheduling under extreme uncertainty. The growing adoption of metaheuristic techniques, fuzzy ranking models, and hybrid optimization approaches indicates that TrIFN-based scheduling will continue to evolve, addressing more complex industrial challenges.

Conflicts of interests: The authors declare that there are no conflicts of interest.

References

[1] Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1:61–68.

[2] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

[3] Gupta, J. N. D. (1971). A functional heuristic algorithm for the flow shop scheduling problem. *Operations Research*, 19:343–351.

[4] Panwalkar, S. S., Dudek R. A., and Smith M. L. (1973). Sequencing research and the industrial scheduling problem. *Symposium on the Theory of Scheduling and its Application, 86:29-38.*

[5] Dannenbring, D. G. (1977). An evaluation of flow shop sequencing heuristics. *Management Sci*., 23: 1174-1182.

[6] Yager R. R. (1981). A procedure for ordering fuzzy subsets of the unit interval. *Information Sciences*, 24:143–61.

[7] Atanassov, K.T. and Atanassov, K.T. (1999). Intuitionistic fuzzy sets. *Physica-Verlag HD*, 35:1-137.

[8] Sanuja, P., Xueyan, S. (2006). A new approach to two machine flow shop problem with uncertain processing time. *Optimization and Engineering*, 7:329–43.

[9] Gupta, D., Sharma, S., and Aggarwal, S. (2013). Flow shop scheduling on 2-machines with setup time and single transport facility under fuzzy environment. *Opsearch*, 50: 14-24.

[10] Rahmani, U., Wulan, E.R., Huda, A.F., Supian, S. and Subiyanto, S. (2019). Scheduling of flowshop Type Production with Dannenbring Method to Obtain Optimal Results with Observing Waiting Time and Weight of Jobs. *An International Scientific Journal*, 127:212-224.

[11] Selvakumari, K., and Santhi, S. (2020). A new approach for solving flow shop scheduling problem with triangular intuitionistic fuzzy number. *Advances in Mathematics: Scientific Journal*, 9:5815-5825.

[12] Gupta, D.,Goel, S., and Mangla, N. (2022). 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*, 13:1162-1169.

[13] Gupta, D., and Goel, S. (2022). Branch and bound technique for two stage flow shop scheduling model with equipotential machines at every stage. *International Journal of Operational Research*, 44: 462-472.

[14] Kaushik, P., Gupta D. And Goel, S. (2025). Comparative Study Of B&B With Heuristics NEH And CDS For Bi Stage Flow Shop Scheduling Problem Under Fuzzy Environment. *Journal of Mechanics of Continua and Mathematical Sciences*, 20: 73-89.