Global Journal of Mathematics and Statistics
ISSN: 3078-8927
DOI: https://doi.org/10.61424/gjms.v3i1.1137
Date Published Online: November 05, 2026
Journal Homepage: https://bluemarkpublishers.com/index.php/GJME

GJMS
BLUEMARK PUBLISHERS

| RESEARCH ARTICLE

Exact Solution to Solve Multi-Objective Functions In Fuzzy Flow Shop Scheduling Problem

Mustafa Talal Kadhim
Ministry of Education, Babylon Education Directorate, Hila, Iraq
Corresponding Author: Mustafa Talal Kadhim, **E-mail**: mostafat2000k@gmail.com

| ABSTRACT
This paper deals with branch-and-bound algorithms to solve the flow shop scheduling problem to obtain a minimized multi-objective function under a fuzzy environment. Where the processing time is a triangular fuzzy number, the objective of this paper is to obtain an optimal sequence to fuzzy-minimize total weighted tardiness time, so we used three upper bounds and two lower bounds in a branch-and-bound algorithm on two machines. And also compared the branch-and-bound and complete enumeration methods, and compared results with the complete enumeration method in terms of processing time in a MATLAB program and the number of nodes.

| KEYWORDS

Fuzzy environment, enumeration methods, MATLAB program

| ARTICLE INFORMATION

ACCEPTED: September 11, 2026 **PUBLISHED:** November 05, 2026 **DOI:** https://doi.org/10.61424/gjms.v3i1.1137

1. Introduction

In this paper, we will solve fuzzy multi objective function define by fuzzy minimized total weighted tardiness time in flow shop by using exact methods, the important history of this felid started from Zadeh [**] proposed the idea of fuzzy set theory in 1965, McCahon, C. S., & Lee, E. S. [1990] was studied in job shop production systems, those ambiguous work processing times are represented by triangular and trapezoidal fuzzy numbers. The focus [Focus, n.d] was to solve the makespan in TFN flow shop by comparing two local search methods. Ishibuchi, H., Yamamoto, N., Murata, T., & Tanaka, H. [1994] used Genetic algorithms and neighborhood search algorithms to solve NP-hard problems in fuzzy flow shop scheduling problems. Hong, T. P., & Wang, T. N. [1999] used Palmer's heuristic method for fuzzy processing times. Hong, T. P., Huang, C. M., & Yu, K. M [2000] where a multi-objective function is defined by minimizing tardiness time in the fuzzy flow shop problem. Yeh, W. C. [2001] used a simple algorithm for multi-stage flow shop scheduling problems with fuzzy processing times to solve an NP-hard problem defined by minimizing maximum completion time. Wang, L., & Zheng, D. Z. [2003] used an effective hybrid heuristic for flow shop scheduling with fuzzy processing times. Behnamian, J., & Zandieh, M. [2011] used a discrete colonial competitive algorithm for fuzzy multi-objective high-tech sequence-dependent setup time flow shop scheduling. Khalili, M., & Sahebjamnia, N. [2012] solved a bi-objective function defined by minimizing the maximum lateness time for fuzzy flow shop scheduling problems. Engin, O., & Guclu, A. (2018) found a new hybrid metaheuristic for fuzzy hybrid flow shop scheduling problems. Gao, J. X., & Wang, L. (2020) used a hybrid genetic algorithm with local search for solve a fuzzy objective function defined by minimizing tardiness time in the flow shop scheduling problem. Castaneda, J., Martin, X., Ammouriova, M., Panadero, J., & Juan, A. (2022) studied A fuzzy simheuristic for the permutation flow shop problem under stochastic and fuzzy uncertainty. Mathematics. Utama, D. M., Primayesti, M. D., Umamy, S. Z., Kholifa, B. M. N., & Yasa, A. D. [2023]. In order to identify the trend of energy-efficient HFSP

Copyright: © 2026 The Author(s). This open access article is distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC- BY-NC-SA) License (https://creativecommons.org/licenses/by-nc-sa/4.0/), published by Bluemark Publishers.

research and research gaps for future research interests, this study offers a thorough assessment of works that address energy-efficient HFSP scheduling.

2. Triangular Fuzzy Number [McCahon, 1990]

Let $\tilde{A} = (a_1, a_2, a_3)$ is a triangular fuzzy number where $a_1$is the lower limit and $a_3$upper limits. The membership function $u_{\tilde{A}}(x), x \in R^+$ defined by

$$\tilde{A}(\chi) = \begin{cases} 0 & if\ \chi < a_1 \\ \dfrac{\chi - a_1}{a_2 - a_1} & if\ a_1 \le \chi < a_2 \\ \dfrac{a_3 - \chi}{a_3 - a_2} & if\ a_2 \le \chi < a_3 \\ 0 & if\ a_3 \le \chi \end{cases}$$

2.1 Fuzzy Arithmetic Rule [Hong, 2000]

Let us have two fuzzy triangular numbers. $M^\sigma = (m_1, m_2, m_3)$ and $N^\sigma = (n_1, n_2, n_3)$

- **Fuzzy addition** $(\boldsymbol{M^\sigma + N^\sigma}) = (\boldsymbol{m_1 + n_1, m_2 + n_2, m_3 + n_3})$
- **Fuzzy subtraction** $(\boldsymbol{M^\sigma - N^\sigma}) = (\boldsymbol{m_1 - n_1, m_2 - n_2, m_3 - n_3})$
- **Fuzzy Max** $(\boldsymbol{max}(M^\sigma, N^\sigma) = (\max(m_1, n_1), \max(m_2, n_2), \max(m_3, n_3))$
- **Defuzzification** $\boldsymbol{h(M^\sigma)} = \frac{\boldsymbol{m_1 + 2m_2 + m_3}}{\boldsymbol{4}}$

3. Problem Formulation

Let the problem of scheduling n-jobs of a flow shop to minimize total weighted tardiness time, where the processing time is p_j is a triangular fuzzy number (TFN). The problem is formulated as:

$F2\ /\tilde{p}j = TFN\ /\ \sum w_i(Tj)$subject to :

P

$C_j \ge \tilde{p}_j$; j=1,2,...,n

$C_j = C_{j-1} + \tilde{p}_j$; j=2,3,...,n

$T_j \ge C_j - d_j$ j=2,3,...,n

$T_j \ge 0$ j=1,2,3,...,n

4. fuzzy branch and bound algorithms

A branch and bound algorithm for the three-machine flow shop scheduling problem. In the first step, we used the average high ranking (AHR) to convert a triangular fuzzy processing time to a crisp number. The problem appeared as a tree of nodes, and each node has the possibility to issue a partial sequence. The lower bound of each node determines the best partial sequence; after calculating the lower bound, we chose the lowest lower bound of the node. We continued the procedure till the best one was found. At least one job is scheduled. The objective of this algorithm is to find an optimal sequence.

5. Upper bound NEH (Tardiness Heuristic) Or Constructive Heuristic

This upper bound can be computed by the following algorithms:

- **Initial Order**: sort Jobs in non-increasing order as
$$I_i = \frac{w_i}{\sum_{k=1}^{m} P_{i,k} \,.\, k_i}$$
- **Sequential Insertion**: Sort the first two jobs in the order that results in the lowest overall weighted tardiness. For task k = 3... n: Add job k to each of the partial schedule's k potential roles. Maintain the order that reduces the partial schedule's overall weighted tardiness.
- **Upper bound Calculation:** after computing the C_i we get
$$\boldsymbol{UP_1 = \sum_{i=1}^{n} w_i max(0, C_{i,m} - d_i)}$$

Example (1)

Consider 3-jobs available on two machines, and each job has a triangular fuzzy processing time; we calculate an upper bound as follows:

jobs	1	2	3
$M1$	(3,4,5)	(4,5,6)	(1.2.3)
M2	(6,8,10)	(4,6,8)	(8,10,12)
dj	(6,8,10)	(4,6.8)	(8,10.12)
w_i	(1,2,3)	(2,3,4)	(1,1,2)

Solution

Step 1: Create the initial job priority index by converting the ambiguous integers into accurate values.

$$I_i = \frac{h(w_i^\sigma)}{h\left(P_{i,1}^\sigma + P_{i,2}^\sigma\right).h(D_i^\sigma)}$$

Then we get

For Job 1 : $h\left(P_{1,1}^\sigma + P_{1,2}^\sigma\right) = 7, h(w_1^\sigma) = 2$

$h(D_1^\sigma) = 8$ then $I_1 = \frac{2}{7\times8} = 0.035$

For job2: $h\left(P_{2,1}^\sigma + P_{2,2}^\sigma\right) = 7, h(w_2^\sigma) = 3$

$h(D_2^\sigma) = 6$ then $I_2 = \frac{3}{6\times7} = 0.071$

For job3: $h\left(P_{3,1}^\sigma + P_{3,2}^\sigma\right) = 7, h(w_3^\sigma) = 1{,}2$

$h(D_3^\sigma) = 10$ then $I_3 = \frac{1.2}{5\times10} = 0.0.025$

Step 2: Assess the Initial Two Jobs (2,1). Use fuzzy schedule rules to compare sequences (2,1) and (1,2).

Now we are testing the sequence (2,1)

- **For job 2**
  $C_{2,1}^\sigma = P_{2,1}^\sigma = (1{,}2{,}3)$
  $C_{2,1}^\sigma + P_{2,2}^\sigma = (5{,}7{,}9)$
  $T_2^\sigma = \max(0^\sigma, (5{,}7{,}9) - (4{,}6{,}8) = (0{,}1{,}5))$
  Then the cost of sequence =(0,6,22)
- **For job 1**
  $C_{2,1}^\sigma = P_{1,1}^\sigma = (3{,}5{,}7)$
  $C_{2,1}^\sigma + P_{2,2}^\sigma = (8{,}11{,}14)$
  $T_1^\sigma = \max(0^\sigma, (8{,}11{,}14) - (6{,}8{,}10) = (0{,}3{,}8))$
  Then the cost of sequence =(0,6,26)

The total cost for jobs (2,1) equals (0,6,22)+(0,6,26)=(012,48)

The final cost $\frac{0+2(12)+48}{4} = 18$

Now we are testing the sequence (1,2)

For the same methods, the total cost is equal (0,0,12)+(0,18,64)

The final cost$\frac{0+2(18)+64}{4} = 25$

The best sequence (2,1)

Step 3: Put job 3 in Every Position Test, adding job 3 to the three potential sequence places (2,1), we get

- **The first position: (3,2,1)**
  The total cost (0,0,0)+(0,12,36)= (0,12,39)
  The final cost $\frac{0+2(12)+39}{4} = 30.7$
- **The second position** (2,3,1)
  The final cost =21.7

- **The third position (2,1,3)**
  The final cost = 18.5

Then the best sequence is (2,1,3)
We get the $UP_1 = 18.5$

6. Lower Bound
This upper bound can be computed by the following algorithms:

Step 1: find crisp weighted shortest processing time
$P_1^\sigma = P_{i,1}^\sigma + P_{i,2}^\sigma$ with w_i^σ for each job
$Rat(w_i^\sigma) = \frac{h(w_i^\sigma)}{h(P_i^\sigma)}$
Step 2: determine weighted shortest processing time by sorting non-increasing $Rat(w_i^\sigma)$
Step 3: Find the fuzzy $C_{i,k}^\sigma$ for the (WSPT)
Step 4: compute $F2\ /\tilde{p}j = TFN\ /\ \sum w_i(Tj)$ to find the lower bound

Example 2
Consider 43jobs available on two machines, and each job has a triangular fuzzy processing time; we calculate a lower bound by the following:

jobs	(2,1,4)	(2,4,3)	(1,2,6)
M1	(1,2,4)	(2,3,5)	(4,1,5
M2	(2,3,4)	(4,1,2)	(2,3,5)
dj	(2,3,5)	(4,3,2)	(1,3,5)
Wi	(1,2,3)	(1,5,7)	(3,1,6)

Solution:
Step 1:

- **Job1**
  $P_1^\sigma = (2+1,1+2,4+4) = (3,3,8) = h(P_i^\sigma)$
  $h(w_1^\sigma) = \frac{1+2(2)+3}{4} = 2$
  $Rat(w_1^\sigma) = \frac{h(w_i^\sigma)}{h(P_i^\sigma)} = \frac{2}{3} = 0.66$
- **Job2**
  $P_2^\sigma = ((2+2,4+3,3+5) = (4,7,8) = h(P_i^\sigma)$
  $h(w_1^\sigma) = \frac{1+2(5)+7}{4} = 4.5$
  $Rat(w_2^\sigma) = \frac{h(w_i^\sigma)}{h(P_i^\sigma)} = \frac{4,5}{7} = 0.64$
- **Job3**
  $P_1^\sigma = (1+4,2+1,6+5) = (5,3,11) = h(P_i^\sigma)$
  $h(w_1^\sigma) = \frac{3+2(1)+6}{4} = 2.75$
  $Rat(w_3^\sigma) = \frac{h(w_i^\sigma)}{h(P_i^\sigma)} = \frac{2}{3} = 0.91$

Step 2: sort non- increasing for $Rat(w_i^\sigma)$

$Rat(w_3^\sigma) > Rat(w_1^\sigma) > Rat(w_2^\sigma)$
The sequence (3,1,2)
Step3: Find the fuzzy $C_{i,k}^\sigma$ for the (WSPT)

jobs	(1,2,6)	(2,1,4)	(2,4,3)
M1	(4,1,5)	(1,2,4)	(2,3,5)
M2	(2,3,5)	(2,3,4)	(4,1,2)
Dj	(1,3,5)	(2,3,5)	(4,3,2)
Wi	(3,1,6)	(1,2,3)	(1,5,7)

Job3: In machine 1 : $C_{3,1}^{\sigma} = P_{3,1}^{\sigma} = (1,2,6)$

In machine 2 : $C_{3,2}^{\sigma} = P_{3,1}^{\sigma} + C_{3,1}^{\sigma} = (1+4,2+1,6+5) = (5,3,11)$

Job 1: In machine 1 : $C_{1,1}^{\sigma} = C_{3,1}^{\sigma} + P_{1,1}^{\sigma} = (1+2,2+1,6+4) = (3,3,10)$

In machine 2 : $C_{1,2}^{\sigma} = P_{1,2}^{\sigma} + C_{3,2}^{\sigma} = (5+2,+3+3,11+4) = (7,6,14)$

Job 2: In machine 1 : $C_{2,1}^{\sigma} = C_{1,1}^{\sigma} + P_{2,1}^{\sigma} = (3+3,3+4,3+10) = (6,7,13)$

In machine 2 : $C_{2,2}^{\sigma} = P_{2,2}^{\sigma} + C_{1,2}^{\sigma} = (4+7,1+6,2+14) = (11,7,16)$

Step 4: compute the upper bound of the multi-objective function $F2\ /\tilde{p}j = TFN\ /\ \sum w_i(Tj)$

$T_i^{\sigma} = \max(0^{\sigma}, C_{i,2}^{\sigma} - D_i^{\sigma})$

For job3: $T_3^{\sigma} = max(0,(5,3,11) - (1,3,5) = (4,0,6)$

For job1: $T_1^{\sigma} = max(0,(7,6,14) - (2,3,5) = (3,3,9)$

For job2: $T_2^{\sigma} = max(0,(11,7,16) - (4,3,2) = (7,4,14)$

Now $(w_i^{\sigma} \times T_i^{\sigma})$

For job 3 : $(4,0,6) \times (3,1,6) = (12,0,36)$

For job1: $(3,3,9) \times (1,2,3) = (3,6,27)$

For job 2 $(7,4,14) \times (1,5,7) = (7,20,98)$

Now we find fuzzy $F2\ /\tilde{p}j = TFN\ /\ \sum w_i(Tj)$

$$\boldsymbol{LB=}\ (12,0,36) + (3,6,27) + (7,20,98) = (22,26,161)$$

$$LB = \frac{22 + 2(26) + 161}{4} = 58.75$$

7. Computational results

In this paper, we used MATLAB code to compare a branch-and-bound method and a complete enumeration method, from which we measured the processing time in MATLAB, and also compared the lower bounds of a branch-and-bound method. In the below used computer core i7 and CPU @2.40GHs to run MATLAB R2025a, where n = 3,4,5,6,7,8 as follows:

n: no. of jobs,
Ex: no. of examples,
Opt (BAB): The optimal value of the function using (BAB) branch and bound
Opt (CEM): The optimal value of the function using (CEM), the complete enumeration method.

NOD: Number of nodes

Time: The execution time of the problem (in seconds).

Table (1) compares CEM and branch and bounds

n	EX	CEM	TIME	UB	LB	TIME	NOD
3	1	77.334	0.012	78.860	77.334	0.015	8
	2	83.235	0.009	89.454	83.235	0.025	10
	3	75.123	0.067	75.980	75.123	0.051	14
	4	98.437	0.012	100.769	98.437	0.023	10
	5	103.650	0.101	106.656	103.650	0.067	17
4	1	193.349	0.033	197.656	193.349	0.090	42
	2	204.44	0.123	209.544	204.44	0.093	54
	3	350.454	0.322	370.655	350.454	0.101	68
	4	187.879	0.223	190.544	187.879	0.121	72
	5	111.431	0.223	112.334	111.431	0.092	48
5	1	202.980	0.675	206.554	202.980	0.345	209
	2	195.769	0.342	195.901	195.769	0.409	304
	3	308.933	0.443	330.233	308.933	0.325	204
	4	206.554	0.332	209.433	206.554	0.301	201
	5	401.454	0.233	442.433	401.454	0.201	130
6	1	304.445	2.332	308.677	304.445	0.298	185
	2	196.345	2.390	201.445	196.345	0.299	177
	3	314.564	3.432	360.544	314.564	0.204	195
	4	501.876	1.233	556.545	501.876	0.301	205
	5	309.877	2.344	350.554	309.877	0.133	105
7	1	609.550	12.334	680.453	609.550	99.443	967
	2	640.654	14.233	701.330	640.654	98.344	913
	3	569.556	15.554	602.334	569.556	7.335	1006
	4	501.565	21.344	502.344	501.565	12.345	1102
	5	401.445	25.332	415.553	401.445	11.334	1001
8	1	1023.434	109.223	1102.110	1023.434	120.222	2020
	2	980.454	123.334	103.444	980.454	109.233	1190
	3	1205.454	154.332	1309.443	1205.454	109.22	1204
	4	1103.345	130.442	1120.444	1103.345	111.233	1402
	5	1309.334	165.443	1401.330	1309.334	106.332	1930

8. Conclusion

The two machine fuzzy flow shop is solved by using branch and bound method for the lower bound using some algorithms and upper bound by use NEH to solved NP-hard problem (P) of multi objective function under triangular fuzzy number with compare the result with the complete enumeration method by using coding of MATLAB where $n \geq 8$ through the table (1).

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers

Artificial Intelligence (AI) Use Disclosure: The authors declare that no artificial intelligence tools were used in the preparation of this manuscript.

References

[1] Behnamian, J., & Zandieh, M. (2011). A discrete colonial competitive algorithm for fuzzy multi-objective high-tech sequence-dependent setup time flowshop scheduling. International *Journal of Computer Integrated Manufacturing, 24(*6), 493–507.

[2] Castaneda, J., Martin, X., Ammouriova, M., Panadero, J., & Juan, A. (2022). A fuzzy simheuristic for the permutation flow shop problem under stochastic and fuzzy uncertainty. Mathematics, 10(10), 1760.

[3] Engin, O., & Guclu, A. (2018). A new hybrid metaheuristic for fuzzy hybrid flow shop scheduling problems. Soft Computing, 22(12), 3925–3939.
[4] Focus (n.d) Demonstrates GA formulations for handling triangular fuzzy processing times with ranking functions.
[5] Focus (n.d) Applies Simulated Annealing (SA) metaheuristics to minimize fuzzy makespan with various ranking methods.
[6] Focus (n.d) Formulates L-R fuzzy numbers instead of simple triangular numbers to represent job processing durations.
[7] Focus (n.d) Integrates local search procedures with global optimization for multi-objective fuzzy scheduling.
[8] Focus (n.d) The earliest landmark paper introducing triangular fuzzy numbers (TFNs) to model uncertain job processing times in flow shop environments.
[9] Gao, J. X., & Wang, L. (2020). A hybrid genetic algorithm with local search for solving fuzzy flow shop scheduling problem. IEEE Access, 8, 114210–114221.
[10] Hong, T. P., & Wang, T. N. (1999). Palmer's heuristic method for fuzzy processing times. IEE Proceedings - Control Theory and Applications, 146(6), 512–516.
[11] Hong, T. P., Huang, C. M., & Yu, K. M. (2000). L-R fuzzy processing times in flow shop scheduling. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 30(1), 78–86.
[12] Ishibuchi, H., Yamamoto, N., Murata, T., & Tanaka, H. (1994). Genetic algorithms and neighborhood search algorithms for fuzzy flow shop scheduling problems. Fuzzy Sets and Systems, 67(1), 81–100.
[13] Khalili, M., & Sahebjamnia, N. (2012). A bi-objective solving approach for fuzzy flow shop scheduling problems. *International Journal of Production Research, 50*(24), 7256–7270.
[14] McCahon, C. S., & Lee, E. S. (1990). Job sequencing with fuzzy processing times. Computers & Mathematics with Applications, 19(7), 31–41.
[15] Temiz, I., & Suer, G. A. (2014). Simulated annealing for fuzzy flow shop scheduling. Computers & Industrial Engineering, 72, 131–141.
[16] Tsujimura, Y., Gen, M., & Kubota, E. (1997). Flow shop scheduling with fuzzy processing time using genetic algorithms. Computers & Industrial Engineering, 33(3-4), 521–524.
[17] Utama, D. M., Primayesti, M. D., Umamy, S. Z., Kholifa, B. M. N., & Yasa, A. D. (2023). A systematic literature review on energy-efficient hybrid flow shop scheduling. Cogent Engineering, 10(1), 2206074.
[18] Wang, L., & Zheng, D. Z. (2003). An effective hybrid heuristic for flow shop scheduling with fuzzy processing times. The International Journal of Advanced Manufacturing Technology, 21(7), 513–522.
[19] Yeh, W. C. (2002). A simple algorithm for multi-stage flowshop scheduling problems with fuzzy processing times. *The International Journal of Advanced Manufacturing Technology, 19(*10), 763–770.