International Journal of Applied Mathematics and Numerical Research

Compare Between Exact Solution and Local Search Solution to Solve Minimize Multi-Objective Function

Mustafa Talal Kadhim
Ministry of Education Directorate, Babil, Iraq

* Corresponding Author: **Mustafa Talal**

Article Info

Impact Factor (RSIF): 8.62
ISSN (Online): 3107-7110
Volume: 02
Issue: 02
Received: 03-01-2026
Accepted: 05-02-2026
Published: 03-03-2026
Page No: 35-38

Abstract
This paper deals with a branch and bound algorithms to solve flow shop scheduling problem, to obtained minimize multi objective function under fuzzy environment. Where a processing time is a triangular fuzzy number, the objective of this paper obtained optimal sequence to minimized total cost tardiness time and max completion time, so we used three upper bound and two lower bounds which on a branch and bound algorithm on two machines. And compere between branch and bound and complete enumeration method, Local search and compare results with complete enumeration method from time of processing in MATLAB program and number of nodes.

DOI: https://doi.org/10.54660/IJAMNR.2026.2.2.35-38

Keywords: Flow Shop Scheduling, Branch and Bound Algorithm, Triangular Fuzzy Number (TFN)

1. Introduction

In the real life application the flow shop exists when a processing of all jobs are same order on all machines. A branch and bound algorithm mostly used to obtained the optimal sequence to solve multi NP-hard problem. In the scheduling problem a branch and bound technique was a first applied by Lomnicki, Ignall and Schrage in 1965 [1]. Brooks and White where developed branch and bound technique in 1965. McMahon and Burton where applied branch and bound technique in 1967. Brucker, Jurisch and sievers were found optimal sequence by used branch and bound technique in 1994 [2]. Koulamas was studied NP-herd problem formulated by F| | ΣTi for m ≥3 [3]. McCahon and Lee were modified algorithm of Campbell, Dudek and Smith (CDS) to applied of NP-hard problem where a processing time is trapezoid fuzzy number [4]. Ishii et al. was studied scheduling problem where a due date is fuzzy number [5]. Celano et al. was used genetic algorithm to solve NP-hard problem of fuzzy flow shop [6]. Singh et al. was solved a fuzzy flow shop problem of two machine by some heuristic [7]. Huanget al solved NP-herd flow shop problem where a processing time is fuzzy number by using genetic algorithm [8]. Feng-Tse Lin, Chieh-Hung Huang was used distance ranking of fuzzy numbers to solve flow shop scheduling problem with fuzzy processing time [9]. In [10] where found algorithm to solve NP-herd problem of total cost earliness and total tardiness of fuzzy processing time. S. Muruganandam and K. Hema were used branch and bound method to solved the fully fuzzy assignment problem and found optimal sequence [11]. Izzettin temiz and serpil erol were solve minimized completion time (C_{max}) with fuzzy processing time by used branch and bound [12]. N. Selvamalar and V. Vinoba were solved objective function of minimized of makespan and mean flow shop where processing time is fuzzy number by using branch and bound method [13]. *G. Ambika* and *G. Uthra* were studied fuzzy objective function to minimized the total elapsed time with triangular membership functions [14]. The object of this paper applied a branch and bound algorithms to obtained optimal sequence of NP-hard flow shop scheduling problem of two machine with fuzzy processing time to minimized total cost earliness , tardiness and total completion time formulated by F_2 /$\tilde{p}_j$=TFN / $\sum(T_j) + C_{max}$,we used three upper bound and two lower bounds and compare results with complete enumeration method from time of processing in matlap program and number of nodes.

2. Triangular Fuzzy Number [15]

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

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

3. Average High Ranking [16]

Let processing time is triangular fuzzy number given by $\tilde{A} = (a_1, a_2, a_3)$ than processing time of a job calculated by using Yager's Average High Ranking formula (AHR) given by

$$AHR = \frac{3a_2 + a_3 - a_1}{3}$$

4- Local search

In computer science and artificial intelligence, local search algorithms are employed to address optimization and search issues. Local search algorithms concentrate on enhancing a single candidate solution by making little, gradual adjustments, in contrast to systematic search algorithms that investigate the whole search space. When the search space is very big and it is not feasible to look at every potential solution, these techniques are especially helpful. Starting with an initial solution which could be generated heuristically or randomly and then periodically moving to an adjacent solution that is superior based on the given evaluation or objective function is the basic notion behind local search. A solution that is marginally different from the existing answer is called a neighbor. Until a stopping condition is satisfied or a better surrounding solution cannot be identified, the process keeps on. The fact that local search does not record the route traveled to find the solution is one of its primary features. Local search algorithms are widely used in real-world applications, including scheduling, routing, resource allocation, machine learning, and combinatorial optimization problems such as the traveling salesman problem. Their efficiency and scalability make them a practical choice when obtaining exact solutions is difficult or impossible.

5. Problem Formulation

Let a problem of scheduling n-jobs of flow shop to minimize total cost earliness , tardiness and total completion time where a processing time is p_j is triangular fuzzy number (TFN).the problem formulated by:

$$\left. \begin{array}{ll} F2\ /\tilde{p}j = TFN\ /(Tj)\ + C_{max} & \\ \text{subject to :} & \\ C_j \ge \tilde{p}_j; & j=1,2,\ldots,n \\ C_j = C_{j-1} + \tilde{p}_j; & j=2,3,\ldots,n \\ T_j \ge C_j - d_j & j=2,3,\ldots,n \\ T_j \ge 0 & j=1,2,3,\ldots,n \end{array} \right\} \quad p$$

6. fuzzy branch and bound algorithms

A branch and bound algorithms of three machine flow shop scheduling problem considered by Ignall and Scharge's [17]. In the first step used average high ranking (AHR) to convert a triangular fuzzy processing time to crisp number. The problem appeared as a tree of nodes and each nodes has possibility to issue to partial sequence. The lower bound of each node determent the best partial sequence after calculated the lower bound we chosen the lowest lower bound of node. We continued the procedure till the least one is found. At least all a jobs are scheduled. The object of this algorithms found optimal sequence.

7. Algorithm Simulated Annealing

The physical process of annealing in metallurgy, in which a material is heated and then gradually cooled to minimize flaws and reach a low-energy stable state, served as the inspiration for the probabilistic optimization approach known as simulated annealing (SA). Simulated annealing is a local search process used in computer science and artificial intelligence to locate nearly optimal solutions for intricate optimization problems with wide search spaces. An initial temperature value and an initial solution are the starting points of the algorithm. Every iteration creates an adjacent solution by slightly altering the existing solution at random. The new solution is adopted right away if it offers a better goal (lower cost or more efficiency). However, depending on the cost difference and the present temperature, the new solution may be accepted with a certain probability if it is worse. One of the primary drawbacks of straightforward hill-climbing techniques is local optima, which the algorithm can avoid thanks to its probabilistic acceptance of poorer solutions.

7.1 Algorithm of Simulated Annealing

```
        Set current solution to initial values where i=0 then t = t⁻
        Set the temperature T to its initial value.

        Create neighbor solution while T > Tmin.
        The formula for ΔE is cost(neighbor solution) minus  cost(current
solution).

     if ΔE < 0:
         Alternatively, accept neighbor solution with probability exp(-ΔE /
T).

     Reduce the temperature T

Give back the finest solution.
```

8. Tabu search Algorithms

By preventing local optimizations, Tapu Search is a metatheoretic optimization algorithm that enhances the effectiveness of local search techniques. It was first presented by Fred Glover and is frequently employed to solve complicated, optimization, and synthesis problems. Using adaptive memory structures to direct the search process and stop the algorithm from going back to previously visited solutions is the fundamental principle behind taboo search. In order to determine the optimal course of action, the algorithm repeatedly investigates its neighbors after beginning with a potential starting solution. Taboo search permits unoptimized motions to avoid local peaks, in contrast to basic local search techniques that only accept improved solutions. Recently visited moves or solutions are kept in the taboo list, which forbids the algorithm from returning to them for a certain amount of iterations known as the taboo period, in order to avoid repetition.

8.1 Algorithms

Set up current solution initially.
Set best solution to its initial value.
Set up tabu_list.

when the stopping condition is not satisfied:
Create the present solution's neighborhood
Choose the best non-tabu move (or tabu if the ambition requirement is met).

Use move to obtain a new solution.
Modify the tabu_list
If the best solution is improved, update it.

Give back the finest solution.

9. Threshold Acceptance Algorithm

Threshold Acceptance (TA) is an optimization algorithm that falls under the category of metaheuristic techniques and is based on deterministic local search. It was put forth as a substitute for the simulated annealing process with the goal of keeping the capacity to escape local optima while streamlining the acceptance mechanism. Threshold acceptance uses a deterministic threshold value to decide whether or not poorer solutions can be accepted, as opposed to a probabilistic acceptance criteria. The algorithm begins with a threshold value and an initial solution. By making a minor adjustment to the existing solution, an adjacent solution is produced in each repetition. The new solution is accepted right away if it enhances the objective function. Even if the remedy is worse, it can still be approved as long as the cost increase stays below the present level. By allowing a small decline in solution quality, this mechanism enables the search to deviate from local optimality. Similar to the cooling schedule in simulated annealing, the threshold value is progressively lowered as the search goes on in accordance with a predefined timetable. By accepting inferior solutions more readily at the start of the search, a high threshold permits broader exploration of the search space. The algorithm gets more discriminating and concentrates more on intensifying around high-quality solutions as the threshold drops.

9.1 Algorithm

Set current solution to initial values.
Set the threshold T and initialize best solution

Generate neighbor solution while T > Tmin. ΔE = cost (neighbor solution) − cost (current solution)

Accept neighbor solution if ΔE <= 0; otherwise, accept neighbor solution if ΔE <= T.

If the best solution is improved, update it.
Lower the threshold T

provide back the finest solution.

10. Upper bounds

In this paper, two upper bounds that are used in branch and bounds algorithms were used:

1. Used average high ranking (AHR) to convert a triangular fuzzy processing time to crisp number
2. $Sl_j = | \tilde{p}_j - d_j |$
3. Arrangement a sequence non-decreasing order of slack time in step 4 to get the last sequence

11. lower bounds

Two lower bounds that are utilized in branch and bounds algorithms are presented in this study as follows:

$$LB(T_r) = \{EDD(T_j) + SPT_1(C_{ij}) + \sum_{i=1}^{n} n - 1(EDD1)$$

12. Computational results

In this paper used coding of MATLAB to compare between a branch and bound method and complete enumeration method from where time of a processing in MATLAB, and also compare between two lower bounds of a branch and bound method. In the below used computer core i7 and CPU @2.40GHs to run matlab R2017a where n = 3,4,5,6,7,8 as a following:

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

Table 1: Compare between CEM and branch and bounds and Local search

n	EX	UP	TABA	SAA	TCA	CEM	TIME	BAB	TIME
3	1	6.33	6.33	6.33	6.33	6.33	0.02	6.33	0.12
	2	4.45	4.45	4.45	4.45	4.45	0.03	4.45	0.24
	3	12.4	12.4	12.4	12.4	12.4	0.02	12.4	0.34
	4	3.34	3.34	3.34	3.34	3.34	0.02	3.34	0.37
	5	5.56	5.56	5.56	5.56	5.56	0.04	5.56	0.21
4	1	35.45	34.98	34.90	34.45	34.45	0.12	34.45	1.34
	2	56.45	55.90	55.78	55.82	55.67	0,25	55.67	2.33
	3	68.12	67.90	68.16	67.70	67.65	0.21	67.65	2.32
	4	92.34	90.91	92.22	90.92	90.45	0.23	90.45	3.32
	5	101.89	***101***.9	102.01	102.34	101.52	0.12	101.52	4.32
5	1	202.22	201.7	207.02	203.34	201.35	2.34	201.35	5.44
	2	555.12	556.34	559.45	558.45	554.87	3.45	554.87	5.43
	3	289.32	391.11	388.33	390.34	387.98	4.34	387.98	7.43
	4	565.12	563.33	560.98	562.23	560.98	3.44	560.98	6.33

	5	490.90	490.78	491.23	492.23	490.87	3.55	490.87	7.54
6	1	702.53	704.44	705.43	703.43	702.43	7.54	702.43	12.34
	2	614.34	621.32	615.34	630.34	613.45	8.66	613.45	11.23
	3	512.45	520.34	520.34	540.34	509.78	7.65	509.78	10.23
	4	782.23	780.68	788.34	782.23	780.67	9.65	780.67	12.34
	5	702.98	701.01	698.54	696.80	696.66	10.45	696.66	11.24
7	1	898.23	900.93	901.23	899.76	897.98	90.43	897.98	22.34
	2	834.33	840.34	804.34	801.91	798.87	95.54	798.87	24.34
	3	920.98	921.32	934.33	922.32	920.76	101.33	920.76	25.54
	4	900.34	902.43	875.44	872.34	865.87	99.33	865.87	30.34
	5	824,43	830.22	811.22	810.32	809.98	103.43	809.98	45.54
8	1	1202.22	1180.21	1180.12	1280.22	1109.98	745.43	1109.98	145.54
	2	995.34	1030.3	1004.3	994.44	990.98	845.45	990.98	201.33
	3	1212.33	1170.22	1334.23	1320.33	1120.65	945.44	1120.65	312.33
	4	1205.34	1202.32	1100.11	1050.33	1009.98	890.44	1009.98	322.33
	5	1190.22	1290.22	1202.32	1120.22	1109.89	1001.34	1109.89	490.44

13. Conclusion

The two machine fuzzy flow shop is solved by using branch and bound method lower bound, upper bounds and local search methods to solved NP-hard problem (P) of multi objective function with compare the result with the complete enumeration method by using coding of MATLAB where $n \geq 8$ through the table (1).

14. References

1. Ignall E, Schrage LE. Application of the branch and bound technique to some flow shop scheduling problems. Operations Research. 1965;13:400–412.
2. Baker KR. Elements of sequencing and scheduling. Amos Tuck School of Business Administration; 1994.
3. Koulamas C. The total tardiness problem: review and extensions. Operations Research. 1994;42(6):1025–1041.
4. McCahon CS, Lee ES. Fuzzy job sequencing for a flow-shop. European Journal of Operational Research. 1992;62:294–301.
5. Ishii H, Tada M, Masuda T. Two scheduling problems with fuzzy due dates. Fuzzy Sets and Systems. 1992;46:339–347.
6. Celano G, Costa A, Fichera S. An evolutionary algorithm for pure fuzzy flowshop scheduling problems. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 2003;11:655–669.
7. Singh TP, Sunita P, Allawalia. Fuzzy flow shop problem on 2-machines with single transport facility—an heuristic approach. Arya Bhatta Journal of Mathematics & Informatics. 2009;1(1–2):38–46.
8. Huang CS, Huang YC, Lai PJ. Modified genetic algorithms for solving fuzzy flow shop scheduling problems and their implementation with CUDA. Expert Systems with Applications. 2012;39(5):4999–5005.
9. Lin FT, Huang CH. Flow-shop scheduling problem with imprecise processing times based on distance ranking of fuzzy numbers. In: International Conference on Machine Learning and Cybernetics; 2016; Jeju, South Korea. IEEE.
10. Kadhum SJ, Aswhad AA, Jasim NS. Fuzzy scheduling problem on two machines. International Journal of Mathematics Trends and Technology. 2016;34(1).
11. Muruganandam S, Hema K. Solving fully fuzzy assignment problem using branch and bound technique. Global Journal of Pure and Applied Mathematics. 2017;13(9):4515–4522.
12. Temiz I, Erol S. Fuzzy branch and bound algorithm for flow shop scheduling. Journal of Intelligent Manufacturing. 2004;14:449–454.
13. Selvamalar N, Vinoba V. Fuzzy branch and bound solution for a flow-shop scheduling problem. Journal of Computer and Mathematical Sciences. 2017;8(3):85–95.
14. Ambika G, Uthra G. Branch and bound technique in flow shop scheduling using fuzzy processing times. Annals of Pure and Applied Mathematics. 2014;8(2):37–42.
15. Dubois D, Prade H. Fuzzy sets and systems. Academic Press; 1980.
16. Yager RR. A procedure for ordering fuzzy subsets of the unit interval. Information Sciences. 1981;24:143–161.