University of Baghdad
College of Engineering
JOURNAL OF ENGINEERING

Journal of Engineering
journal homepage: www.jcoeng.edu.iq
Volume 32 Number 2 February 2026

Local Search Approaches for Solving Triple-Criteria and Triple-Objective Machine Scheduling Problems

Zainab W. Murad [1], **Fadhaa O. Sameer** [2,*]

[1]Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq
[2] Tropical-Biological Research Unit, College of Science, University of Baghdad, Baghdad, Iraq

ABSTRACT

Scheduling problems are central to operations research because of their direct impact on productivity, resource utilization, and system responsiveness. This study addresses a novel tri-criteria and tri-objective model that provides an innovative framework for analyzing complex single-machine scheduling problems of minimizing total completion time ($\sum C_j$), total earliness ($\sum E_j$), and maximum lateness (L_{max}), as well as their aggregated form. The problem remains computationally challenging for large job sizes (up to n = 8000). To evaluate solution performance, three independent approaches were employed: the Branch and Bound (BAB) algorithm and two local search methods, Tabu Search and the Bees Algorithm. While BAB provides precise answers for small cases (n ≤ 19) with low AAE, its computational time increases significantly with problem size. Tabu Search balances solution quality and computational effort to produce near-optimal solutions with modest execution times for medium and large-scale examples (n = 500–8000). While the Bees Algorithm may not match the precision of exact approaches, it offers faster computation and produces a variety of solution patterns. These features make it especially appropriate for large-scale problems or situations with restricted computational time. Moreover, this study offers practical insights to support the selection of suitable solution techniques, taking into account problem size, available computational resources, and the required balance between efficiency and solution quality.

Keywords: Branch and bound, Bees algorithm, Tabu search, Local search, Single-machine scheduling.

1. INTRODUCTION

Multi-objective and multi-criteria scheduling problems (MSPs) are essential topics in operations research because they help improve scheduling efficiency, reduce operational costs, and better utilize resources **(Luo, 2023; Zhang, 2025; Costa et al., 2025; Abbas and Ghayyib, 2024)**. Single-Machine Scheduling Problems (SMSPs) are among the most important models used in the real world, including service systems, computing

*Corresponding author
Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2026.02.11

Article received: 16/09/2025
Article revised: 12/01/2026
Article accepted: 19/01/2026
Article published: 01/02/2026

environments, and industrial operations **(Pinedo, 2008; Abdulqader and Ali, 2023; Hassan et al., 2022)**. As scheduling environments become more complex, with changing conditions, diverse constraints, and conflicting operational goals, traditional single-objective models are becoming less useful for solving real-world problems **(Motair, 2017; Chachan and Hameed, 2019; Ali and Jawad, 2020; Sameer, 2023).**

Recent studies have highlighted a clear research gap in the single-machine scheduling literature regarding models formulated with tri-criteria and tri-objective models, particularly in terms of applying and comparing exact and local search solution approaches **(Ibrahim and Ali, 2022; Yousif et al., 2023; Neamah and Kalaf, 2024a; Neamah and Kalaf, 2024b).** This gap motivates the present study, which addresses this limitation by proposing two novel scheduling formulations, namely TC-SCSELm and TO-SCSELm**.** In the TC-SCSELm model, the three objectives are total completion time ($\sum C_j$), total earliness ($\sum E_j$), and maximum lateness (L_{max}) are treated independently, whereas the TO-SCSELm model aggregates these objectives into a single scalarised objective function(F_{TO} in Equation 2) to enable effective comparison. The two models are NP-hard. The objectives are considered without predefined weights to ensure a fair evaluation of all criteria.

This study examines three methods for multi-criteria scheduling due to their computational complexity. The exact Branch-and-Bound (BAB) provides optimal solutions for small instances **(Ahmed and Ali, 2022; Abbass, 2019)**. Two independent local search algorithms are also considered: TLSM-A, using memory-based strategies to escape local optima **(Glover, 1989)**, and BLSM-A, inspired by swarm foraging, balancing global exploration with local exploitation **(Pham et al., 2009; Aurasopon et al., 2025)**. All methods were applied in their original forms, allowing a fair comparison of solution quality, runtime, and stability across problem sizes.

2. MACHINE SCHEDULING PROBLEMS CONCEPTS

In this study, we focus on a single-machine tri-criteria and tri-objective problem. The following notations:

p_j: Represents the processing time required to complete job j.

d_j :Indicates the due date assigned to job j.

s_j: Slack time of job j, s.t. $s_j = d_j - p_j$.

C_j: The completion time of job j, such that $C_j = \sum_{k=1}^{j} p_k$.

$\sum C_j$: Total completion time.

L_j: Lateness time for each j, s.t. $L_j = C_j - d_j$.

L_{max}: Maximum of lateness time, s.t. $L_{max} = max\{L_j\}$.

E_j: Earliest time for each j, such that $E_j = max\{-L_j, 0\}$.

$\sum E_j$: Total earliness time.

- EDD Rule **(Khusna and Prabowo, 2025, Peng et al., 2021; Dos Santos et al., 2024):** The $1//\ L_{max}$ *The* problem is solved by sequencing jobs according to the earliest-due-date (EDD) rule, arranging them in ascending order of their due dates d_j.
- SPT Rule **(Jamalabadi and Schwiegelshohn, 2023; Yang et al., 2025):** The $1//\ \Sigma C_j$ The scheduling problem can be optimally solved by arranging the jobs in ascending order of their processing time, following the Shortest Processing Time (SPT) rule.

3. TRIPLE-CRITERION and TRIPLE-OBJECTIVE MATHEMATICAL FORMULATION

This section presents the mathematical model of the Triple-Criteria Single-Machine Scheduling Problem (TC-SCSELm) and it scalarized Triple-Objective formulation (TO-SCSELm). The goal is to minimize three conflicting objectives simultaneously: total completion time ($\sum C_j$), total earliness ($\sum E_j$) and maximum lateness (L_{max}).

$$\left.\begin{aligned} & F_{TC} = min\left(\sum C_j, \sum E_j, L_{max}\right) \\ & s.t. \\ & C_1 = p_{\sigma(1)} \\ & C_j = C_{j-1} + p_{\sigma(j)}, && j = 2,3,,,n \\ & L_j = C_j - d_j, && j = 1,2,,,n \\ & E_j \geq L_{j,} && j = 1,2,,,n \\ & E_j \geq 0 \\ & C_j \geq 0 \end{aligned}\right\} \quad (1)$$

The model described by Eq. (1) is strongly NP-hard for several reasons. First, the number of possible job sequences grows factorially with the number of jobs *n*, leading to a rapidly expanding solution space. Second, adding the early objective ($\sum E_j$) introduces another layer of combinations, making the problem even harder to solve. The scalarized objective function in Eq. (2) combines the three objectives into one measure of how well they do. In this formulation, the objectives are not given any weight. Instead, a simple summation is used to combine them, not a weighted sum.

$$\left.\begin{aligned} & F_{TO} = min\left(\sum C_j + \sum E_j + L_{max}\right) \\ & s.t. \\ & C_1 = p_{\sigma(1)} \\ & C_j = C_{j-1} + p_{\sigma(j)}, && j = 2,3,,,n \\ & L_j = C_j - d_j, && j = 1,2,,,n \\ & E_j \geq L_{j,} && j = 1,2,,,n \\ & E_j \geq 0 \\ & C_j \geq 0 \end{aligned}\right\} \quad (2)$$

4. PROPOSED SOLUTION TECHNIQUES FOR TC-SCSELM AND TO-SCSELM MODELS

Since both the TC-SCSELm and TO-SCSELm models are computationally intensive. This section presents three solution methods; each applied to both models separately. To ensure a fair comparison of the performance of the TC-SCSELm and TO-SCSELm models across varying problem sizes, all three algorithms are executed independently on each model. Comprehensive explanations of each method are presented below.

4.1 Branch and Bound (BAB)

The Branch and Bound (BAB) algorithm is a classical, exact optimization method widely used in scheduling problems **(Ahmed and Ali, 2022; Abbass, 2019).** It systematically explores the solution space by dividing it into subproblems and eliminating non-promising branches using bounding rules. In the TC-SCSELm model, BAB optimizes each objective separately to produce Pareto-efficient schedules. For TO-SCSELm, it minimizes a single aggregated objective,

F_{TO}. Although BAB provides optimal solutions, its high computational cost confines its use to small instances, serving mainly as a benchmark for other methods

Algorithm 1: Branch and Bound (BAB) for TC-SCSELm and TO-SCSELm

σ**:** Partial or complete job sequence.
δ**:** Set of non-dominated sequences (Pareto-optimal solutions).
LB (σ)**:** Lower bound of sequence σ.
UB (σ)**:** Upper bound of sequence σ.

Procedure:
Step 1**:** Input $p_j, d_j n$, for $j = 1,2, \dots, n,\ \delta = \emptyset$.
Step 2: For each partial sequence σ:
a. Compute $F_{TC}(\sigma) = (\sum C_j(\sigma), \sum E_j(\sigma), L_{max}(\sigma))$.
b. Estimate Upper Bound (UB):
- If the model is TC-SCSELm, $UB_{TC} = F_{TC}(\sigma)$
- If the model is TO-SCSELm, generate the SPT sequence and compute
$UB_{TO} = F_{TO}(\sigma) = (\sum C_j(\sigma) + \sum E_j(\sigma) + L_{max}(\sigma))$,
Step 3: For each node and partial sequence σ;
a. Compute LB (σ) = Cost of scheduled jobs + Cost of remaining jobs (SPT rule).
b. For TO-SCSELm: include both sequenced and un-sequenced jobs.
Step 4: Pruning: If LB (σ) ≥ UB (σ): Discard branch # Remove non-promising sequences.
Step 5: Terminal nodes: If σ is complete and non-dominated: Add σ to δ # Store non-dominated solution. Remove dominated solutions from δ # Keep only the best solutions.
Step 6: Continue steps 2–4 until no active nodes remain.
Step 7: Output δ.

4.2 Bees Algorithm

Nature-inspired metaheuristics efficiently address complex scheduling problems by adapting search strategies. The Bees Algorithm **(Pham et al., 2005)** imitates honeybee foraging, combining global exploration by scout bees with local search by workers to solve combinatorial challenges **(Li et al., 2011; Huang et al., 2025).** In this study, the Bees Local Search Method Algorithm (BLSM-A) is applied to single-machine scheduling with three objectives: total completion time ($\sum C_j$), total earliness ($\sum E_j$), and maximum lateness (L_{max}). Each feasible solution represents a complete job sequence, evaluated within the optimization framework.

- For the TC-SCSELm (multi-objective) model, solutions are compared using Pareto dominance, and non-dominated schedules are retained.
- The TO-SCSELm (aggregated) model combines objectives into a scalar fitness function to guide schedule improvement.

BLSM-A drives bees to search more intensively for possible solutions, while scout bees explore new areas to maintain diversity. Adjusted algorithm parameters balance convergence speed and local optima avoidance. Multi-objective and aggregated models can be searched for more efficiently using the update technique, thereby improving solution quality.

Algorithm 2: Bees Local Search Method Algorithm (BLSM-A)
Input:
1. Number of bees B.
2. Number of elite sites E.
3. Number of selected sites (S).
4. Neighbourhood size NS.
5. Maximum iterations (MaxIter)
6. Job set $j = \{1,2,\ldots,n\}$ with processing times p_j and deadlines d_j.
Output: Best-found schedule σ*.
Procedure:
Step 1**:** Start with B random job sequences.
Step 2: For each solution σ:
a. Evaluate $F_{TC}(\sigma)$or $F_{TO}(\sigma)$ based on the model.
Step 3: Identify elite sites (E) and selected sites (S) based on:
a. $F_{TC}(\sigma)$in multi-objective mode
b. $F_{TO}(\sigma)$ in scalar mode
Step 4: Set σ^*As the top solution in the original population.
Step 5: For iter = 1 to MaxIter do:
a. Elite Site Search: For each elite site:
i. Recruit multiple bees
ii. Explore the neighborhood of size NS.
iii**.** Evaluate the solution using $F_{TC}(\sigma)$or $F_{TO}(\sigma)$.
iv. Retain the best solution at each elite site.
b. Selected Site Search: For each selected site:
i. Recruit fewer bees.
ii. Explore a reduced neighborhood.
iii. Evaluate the solution using $F_{TC}(\sigma)$or $F_{TO}(\sigma)$.
iv**.** Retain the best solution at the chosen site.
c. Scout Bee Search: Maintain population diversity by randomly searching the surviving bees.
d. Update Global Best: Change σ^* To the new solution, if it is superior σ.
Step 6: End
Step 7: Return σ^*.

4.3 Tabu Search

The Tabu Local Search Method Algorithm (TLSM-A), derived from Fred Glover's Tabu Search, is widely employed in combinatorial optimization and has proven effective in scheduling, particularly in flexible job-shop and complex production settings **(Fekih et al., 2020; Zhang, 2025; Xie et al., 2019; Hajibabaei and Behnamian, 2021; Dabah et al., 2019)**. This research utilizes TLSM-A to address a single-machine scheduling problem with three objectives: total completion time ($\sum C_j$), total earliness ($\sum E_j$), and maximum lateness (L_{max}). The algorithm has two ways to work:

1. Multi-objective (TC-SCSELm): Each goal is optimized on its own, and Pareto dominance is used to find solutions that aren't dominated and weigh the pros and cons.
2. Aggregated (TO-SCSELm): All goals are put together into one scalar function to optimize everything at once.

Algorithm 3: TLSM-A (Tabu Local Search Algorithm)
σ_0: Initial job sequence.
σ: Current solution.
σ*: The optimal solution discovered during the inquiry.
σ': A potential solution based on σ (neighborhood).
Tabu List: Memory structure to avoid revisiting recent solutions.
Procedure:
Step 1: Input initial solution σ_0, Tabu tenure T, Maximum iterations MaxIter
Step 2: Initialize the current solution: Assign $\sigma = \sigma_0$.
Step 3: Assess the goal function (F (σ) either $F_{TC}(\sigma)$ or $F_{TO}(\sigma)$) and designate $\sigma*$ to the present solution σ.
Step 4: Initialize the Tabu List as empty
Step 5: For iteration = 1 to MaxIter, do:
a. Generate the neighborhood N(σ) (e.g., via adjacent job swaps)
b. For each candidate $\sigma' \in N(\sigma)$, where σ' is a neighboring solution generated from the current solution σ.
c. If σ' is not in TabuList OR F(σ') is better than F(σ*), mark σ' as admissible
d. Select the best admissible solution and assign it to σ best
e. Update the current solution: set σ equal to σ _best
g. If F(σ) is better than F(σ*), update the best solution: set σ*equal to σ
h. Update the TabuList with σ, removing expired entries
Step 6: End.
Step 7: Return (Best-found solution σ*)

5. APPLICATION OF THE PROPOSED SOLUTION TECHNIQUES TO TC-SCSELM AND TO-SCSELM MODELS

This part assesses how well the proposed solution techniques perform by reporting the average execution time (in seconds) across all test cases. Execution time is a key performance indicator because it directly shows how much work, efficiency, and scalability the algorithms have as the problem size grows. Given the combinatorial nature of the TC-SCSELm and TO-SCSELm models, runtime represents a consistent and practically meaningful basis for comparing the exact BAB method with the two local search–based procedures (BLSM-A and TLSM-A). For every test case, the MSP generates a random processing time p_j and d_j s.t., $p_j \in [1,10]$ and

$$d_j \in \begin{cases} [1,30], & 1 \le n \le 29 \\ [1,40], & 30 \le n \le 99 \\ [1,50], & 100 \le n \le 999 \\ [1,70], & otherwise \end{cases}$$

subject to condition $d_j \geq p_j$, for j = 1, 2, …, n. The generated instances simulate real-world single-machine scheduling scenarios, such as manufacturing lines or service operations, ensuring practical relevance. All computational experiments were executed on a workstation equipped with a 13th Gen Intel® Core™ i9-13900HX CPU (2.20 GHz) and 32 GB of RAM, operating on Windows 11. To guarantee reliability and stability, each problem instance of size n was executed five times independently, using the same settings each time. The average of these runs reduces the impact of random variability common in local search processes, yielding more accurate performance measures.

The average runtime across all experiments ranged from R to 1800 seconds, depending on the instance's complexity. This shows that each algorithm can be used in real life: BAB works best for small cases (n ≤ 19), TLSM-A finds the right balance between accuracy and cost for medium and large instances (up to n = 8000), and BLSM-A works quickly with acceptable solution quality for significant, time-sensitive problems. The following short forms are used in this part:
ANES: The average number of solutions that work.
AEE: Average Absolute Error.
AT/S: The average number of seconds.
R: A real number between 0 and 1 that shows minimal runtime values or fractional averages.

6. COMPARATIVE ANALYSIS OF BAB, BEES, AND TABU ALGORITHMS FOR MULTI-OBJECTIVE SCHEDULING

This section compares three algorithms: Branch and Bound (BAB), Tabu Local Search (TLSM-A), and Bees Local Search (BLSM-A), using experimental data reported in **Tables 1 to 4.** The analysis divides problems into three sizes: small (n= 10-19), medium, and large (n = 500-8000). It evaluates solution quality (F_{TC} and F_{TO}), absolute average error (AAE), and computing efficiency (AT/S).

6.1 Small Instances (n=10–19)

In **Tables 1 and 3**, the following can be seen:

- **Solution Quality:** BAB achieves the lowest aggregated objective value (572.9), confirming it as the most accurate and reliable method. TLSM-A provides near-optimal results (F_{TO}=617.7), while BLSM-A shows slightly higher values (F_{TO}=636.7).
- **AAE:** TLSM-A generally has lower AAE than BLSM-A (44.8 vs. 63.8), indicating higher accuracy.
- **Computational Efficiency:** BAB run faster than TLSM-A and BLSM-A, which remains costly even after normalization (average AT/S = 72.7).

Implication: BAB is the best choice for small instances in terms of accuracy; TLSM-A balances accuracy and runtime, while BLSM-A is suitable for faster but less precise solutions.

6.2 Medium to Large Instances (n=500–8000)

In **Tables 2 and 4** we can see the following:

- **Solution Quality:** TLSM-A outperforms BLSM-A in combined objectives (average F_{TO}=64,007,879), offering better overall precision.
- **ANES:** TLSM-A maintains lower ANE than BLSM-A (2.8 vs. 3.6), showing more accurate solutions.
- **Computational Efficiency:** BLSM-A runs significantly faster (AT/S = 38 vs. 1210 for TLSM-A at n=8000), making it preferable for rapid computations.

Implication: TLSM-A is best for high-accuracy answers, and BLSM-A is best for fast, approximate outcomes.

Key Insights:

• The BAB method works best for small cases (n≤19), but it doesn't work well for larger ones.
• TLSM-A strikes a good balance between accuracy and runtime for all sizes.
• In scenarios with a lot of objects or big ones, BLSM-A is faster but less accurate.
• AAE reveals that TLSM-A is usually more accurate than BLSM-A.

Table 1. The comparison results between the (BLSM-A, TLSM-A) and BAB methods

Ex	BAB			BLSM-A			TLSM-A		
n	AV(F_{TC})	AT/S	ANES	AV(F_{TC})	ANES	AAE	AV(F_{TC})	ANES	AAE
10	(265.2, 26.8, 34.8)	R	55.0	(296.1, 20.7, 41.2)	5.4	(30.9, 6.1, 6.4)	(257.4,43.2,34.7)	2.8	(7.8,16.4,0.1)
11	(314.9,30.7,38.1)	R	43.2	(364.7,20.6,45.7)	5.4	(49.8,10.1, 7.6)	(307.3,41.8,46.5)	2.2	(7.6,11.1,8.4)
12	(337.1,35.5,40.5)	2.4	108.6	(413.9,24.5,45.6)	6.4	(76.8,11.0, 5.1)	(340.5,52.1,47.1)	3.8	(3.4,16.6,6.6)
13	(372.1,27.7,46.4)	R	58.0	(447.1,24.9,50.7)	6.2	(75.0, 2.8,4.3)	(391.6,43.8,49.6)	3.0	(19.5,16.1,3.2)
14	(438.5,37.0,53.1)	2.2	111.0	(551.1,20.5,59.2)	4.8	(112.6,16.5,6.1)	(490.2,41.7,59.0)	2.8	(51.7,4.7,5.9)
15	(431.5,41.3,52.8)	3.8	75.2	(639.5,27.4,68.2)	5.4	(208.0,13.9,15.4)	(561.1,46.0,70.8)	3.4	(223.5,4.7,18.0)
16	(572.0,29.7,66.2)	34.2	67.4	(721.5,24.3,72.3)	4.0	(149.5,5.4,6.1)	(644.7,47.7,70.8)	3.2	(72.7,18.0,4.6)
17	(620.5,34.7,68.0)	79.5	105.4	(804.5,21.6,77.7)	4.8	(184.0,13.1,9.7)	(704.8,49.4,78.3)	3.2	(84.3,14.7,10.3)
18	(689.5,32.4,73.5)	267.3	74.6	(883.1,26.3,75.5)	4.8	(193.6,6.1,2.0)	(777.4,35.3,78.6)	3.6	(87.9,2.9,5.1)
19	(851.1,32.1,87.9)	337.6	61.2	(957.2,26.1,87.5)	4.6	(106.1,6.0,0.4)	(879.4,43.1,88.6)	2.8	(28.3,11.0,0.7)
AV	**(489.2, 32.7, 55.7)**	**72.7**	**75.9**	**(607.8, 23.7, 62.4)**	**5.18**	**(118.63,9.1,6.31)**	**(535.44,44.41)**	**3.08**	**(58.67,11.62,6.29)**

Remark (1). In Table 1, AT/S for BLSM-A and TLSM-A is equal to R for n = 10: 19.

Table 2. Results of the comparison of (BLSM-A and TLSM-A) for n = 500:8000.

Ex	BLSM-A			TLSM-A		
N	AV(F_{TC})	AT/S	ANES	AV(F_{TC})	AT/S	ANES
500	(673991.8,43.5,2695.3)	3.6	3.8	(655457.4,113.0,2701.6)	4.5	3.0
1000	(2736856.1,106.7,5491.0)	4.7	5.0	(2699755.7,178.0,5490.4)	6.9	3.0
1500	(6109015.3,97.0,8162.6)	9.8	3.6	(6020509.0,123.0,8153.6)	19.0	3.0
2000	(10937896.9,86.7,11011.3)	7.8	3.2	(10853858.5,181.1,11013.9)	18.1	3.0
2500	(17047935.3,104.3,13713.4)	11.1	3.4	(16951533.0,158.2,13716.6)	47.0	3.6
3000	(24644942.7,114.1,16478.5)	20.7	4.0	(24438046.2,120.5,16481.7)	45.0	2.8
3500	(33571383.8,97.4,19253.2)	20.5	4.2	(33296015.6,157.4,19252.0)	96.0	3.4
4000	(44093442.3,84.1,22096.2)	33.4	3.6	(43790807.3,126.7,22096.6)	99.8	3.6
4500	(55412730.7,91.2,24709.3)	23.4	3.2	(54949158.6,130.4,24705.0)	122.2	2.8
5000	(68707016.1,108.3,27590.2)	21.8	3.6	(68288922.1,163.5,27595.6)	213.2	3.4
5500	(83023989.3,13.0,30277.0)	34.0	3.2	(82503994.2,57.5,30271.7)	331.7	2.6
6000	(98395188.3,21.1,32890.7)	34.0	3.6	(97753233.7,30.0,32886.6)	359.8	2.6
6500	(115386660.0,12.3,35644.4)	38.1	3.2	(114972908.5,21.7,35643.8)	1065.1	2.0
7000	(134156765.4,15.0,38443.2)	17.3	3.2	(133541089.8,27.1,38443.2)	643.5	2.2
7500	(154329166.5,14.2,41281.2)	38.4	3.0	(153705918.8,20.5,41282.4)	591.3	2.6
8000	(174899091.3,15.5,43862.0)	47.0	4.8	(174136166.6,32.9,43861.0)	775.0	2.0
AV	**(64007879.5, 52.6, 23349.9)**	**22.8**	**3.6**	**(62132458.0, 91.2, 23349.7)**	**277.4**	**2.8**

Table 3. A comparison of the BLSM-A, TLSM-A, and BAB approaches for n =10:19.

Ex	BAB		BLSM-A			TLSM-A		
n	AV (F_{TO})	AT/S	AV(F_{TO})	AT/S	AAE	AV(F_{TO})	AT/S	AAE
10	322.2	R	327.6	R	5.4	324.8	R	2.6
11	380.6	R	393.4	R	12.8	380.6	R	0.0
12	415.4	R	426.6	R	11.2	416.4	R	1.0
13	447.0	R	474.6	R	27.6	456.4	R	9.4
14	527.8	R	575.8	R	48.0	565.0	R	37.2
15	533.0	R	676.4	R	143.4	651.2	R	118.2
16	669.4	R	768.8	R	99.4	739.8	R	70.4
17	720.0	R	828.4	R	108.4	810.4	R	90.4
18	744.4	12.1	906.8	R	162.4	846.2	R	101.8
19	969.4	12.4	988.4	R	19.0	986.6	R	17.2
AV	**572.9**	**2.4**	**636.7**	**R**	**63.8**	**617.7**	**R**	**44.8**

Table 4. Comparison of BLSM-A and TLSM-A for n = 500:8000

Ex	BLSM-A		TLSM-A	
N	AV(F_{TO})	AT/S	AV(F_{TO})	AT/S
500	664425.0	2.0	670734.4	2.9
1000	2710639.4	4.5	2733552.6	13.4
1500	6064092.4	6.8	6100698.8	19.3
2000	10896430.0	10.2	10939815.6	43.5
2500	16990203.8	5.5	17056215.0	45.0
3000	24509032.4	12.4	24639116.4	114.8
3500	33440691.0	15.1	33636960.0	136.3
4000	43791450.2	15.8	44067288.8	122.3
4500	55263616.4	11.3	55368038.6	248.9
5000	68467812.6	17.6	68720970.6	207.6
5500	82624193.0	25.7	82984987.8	360.0
6000	98036955.8	17.6	98492272.8	761.8
6500	115150250.0	26.9	115491380.8	365.7
7000	133690988.8	24.1	133953551.6	750.4
7500	154045452.0	43.5	154348179.2	600.7
8000	174488641.0	38.3	174883814.6	1210.4
AV	**63802179.6**	**17.3**	**64005473.6**	**312.7**

7. CONCLUSIONS

This study compared three approaches to tri-objective (TO-SCSELm) and tri-criteria (TC-SCSELm) single-machine scheduling: BAB, TLSM-A, and BLSM-A. Performance was evaluated using Scalability, runtime, solution quality, and AAE were the metrics utilized to measure performance. BAB provided the best solutions for minor situations (n < 19); however, for larger cases, a lot of processing effort is required. In cases with 8000-8000 variables, TLSM-A achieved superior outcomes with a lower AAE. For jobs that require speed rather than accuracy, BLSM-A is the way to go. New tri-criteria and tri-objective models are the main contribution of the work, which also includes the first complete evaluation of exact and local search strategies. Finding the optimal solution for your problem size may be easier now that you know there is a trade-off between computation cost and precision. Findings apply to production scheduling, cloud computing, and service operations, and provide a foundation for future hybrid approaches combining Tabu Search and the Bees Algorithm to improve solution quality and convergence.

Credit Authorship Contribution Statement

Zainab W. Murad was responsible for generating ideas, designing the research, collecting data, conducting formal analysis, reviewing the work, and writing the manuscript. Fadhaa O. Sameer was in charge of the project's assessment, editing, and supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

Abbas, I.T., and Ghayyib, M.N., 2024. Using sensitivity analysis in linear programming with practical physical applications. *Iraqi Journal of Science*, 65(2), pp. 907–922. https://doi.org/10.24996/ijs.2024.65.2.27.

Abbass, D.A., 2019. Using branch and bound and local search methods to solve multi-objective machine scheduling problem. In Proceedings of 2019 First *International Conference of Computer and Applied Sciences (CAS)*. Doha, Qatar, December 2019. *IEEE*. https://doi.org/10.1109/CAS47993.2019.9075460

Abdulqader, A., and Ali, S., 2023. Diversity Operators-based artificial fish swarm algorithm to solve flexible job shop problems. *Baghdad Science Journal*, 20, pp. 2067–2076. https://doi.org/10.21123/bsj.2023.6810.

Ahmed, M.G., and Ali, F.H., 2022. Exact method with dominance rules for solving scheduling on a single machine problem with multi objective function. *Al-Mustansiriyah Journal of Science*, 33(2), pp. 56–63. https://doi.org/10.23851/mjs.v33i2.1091.

Ali, F.H., and Jawad, A.A., 2020. Minimizing the total completion time and total earliness time functions for a machine scheduling problem using local search methods. *Iraqi Journal of Science*, Special Issue, pp. 126–133. https://doi.org/10.24996/ijs.2020.SI.1.17.

Ali, F.H., Jawad, R.N., and Hussein, W.A., 2022. Solving bi criteria and bi objectives of total tardiness jobs times and range of lateness problems using new techniques. *Al-Mustansiriyah Journal of Science, 33(3), pp. 27–35.* https://doi.org/10.23851/mjs.v33i3.1135.

Androutsopoulos, K.N., and Zografos, K.G., 2008. A decision support system for integrated hazardous materials routing and emergency response. Transportation Research Part C: *Emerging Technologies*, 16, pp. 684–703. https://doi.org/10.1016/j.trc.2008.01.004.

Atiya, B., Bakheet, A.J.K., Abbas, I.T., Bakar, M.R.A., Soon, L.L., and Monsi, M.B., 2016. Application of simulated annealing to solve multi objectives for aggregate production planning. *AIP Conference Proceedings,* 1739. https://doi.org/10.1063/1.4952566.

Aurasopon, A., Takeang, C., and Khamsen, W., 2025. Enhanced local search for bee colony optimization in economic dispatch with smooth cost functions. *Processes,* 13(3), pp. 1–16. https://doi.org/10.3390/pr13030787.

Chachan, H.A., and Hameed, A.S., 2019. Exact methods for solving multi-objective problem on single machine scheduling. *Iraqi Journal of Science*, 60(8), pp. 1802–1813. https://doi.org/10.24996/ijs.2019.60.8.17.

Costa, A., Corsini, R.R., Pagano, D., and Fernandez-Viagas, V., 2025. A machine learning-based optimization method for large-scale scheduling problems. *Computers & Operations Research*, 173, 106849. https://doi.org/10.1016/j.cor.2024.106849.

Dabah, A., Bendjoudi, A., AitZai, A., and Reghioui, M., 2019. Efficient parallel tabu search for the blocking job shop scheduling problem. *Soft Computing*, 23, pp. 13283–13295. https://doi.org/10.1007/s00500-019-03871-1.

Dos Santos, F., Costa, L., and Varela, L., 2024. Multi-objective scheduling optimization in job shop with unrelated parallel machines using NSGA-III. In Computational Science and Its Applications – ICCSA 2024 Workshops. *Lecture Notes in Computer Science (LNCS)*, vol. 14816, pp. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-031-65223-3_25.

Fekih, A., Hadda, H., Kacem, I., and Hadj-Alouane, A.B., 2020. A hybrid genetic tabu search algorithm for minimizing total completion time in a flexible job-shop scheduling problem. *European Journal of Industrial Engineering*. https://doi.org/10.1504/EJIE.2020.112479.

Glover, F., 1989. Tabu search – Part I. *ORSA Journal on Computing*, 1(3), pp. 103–190.

Hajibabaei, M., and Behnamian, J., 2021. Flexible job-shop scheduling problem with unrelated parallel machines and resources-dependent processing times: a tabu search algorithm. *International Journal of Computer Integrated Manufacturing*, 34(11), pp. 1211–1227. https://doi.org/10.1080/17509653.2021.1941368.

Hassan, D.A., Mehdavi Amiri, N., and Ramadan, A.M., 2022. A heuristic approach to minimize three criteria using efficient solutions. *Indonesian Journal of Electrical Engineering and Computer Science*, 26(1), pp. 334–341. https://doi.org/10.11591/ijeecs.v26.i1.pp334-341.

Huang, X., Yin, J., and Deng, G., 2025. A hybrid discrete artificial bee colony optimization algorithm for the no-wait job shop problem with tardiness criterion. *Journal of Algorithms & Computational Technology*, 19. https://doi.org/10.1177/17483026251322104.

Ibrahim, M.H., and Ali, H., 2022. Solving multi-objectives function problem using branch and bound and local search methods. *International Journal of Nonlinear Analysis and Applications*, 13(1). https://doi.org/10.22075/ijnaa.2022.5780.

Jamalabadi, S., and Schwiegelshohn, U., 2023. WSRPT is 1.2259 competitive for weighted completion time scheduling. *arXiv preprint arXiv:2307.07739*. https://doi.org/10.48550/arXiv.2307.07739.

Khusna, A.N., and Prabowo, F.A., 2025. Enhancing customer satisfaction: Exploring the earliest due date method for production scheduling at PT. X. *Journal of Engineering*, 14(2), pp. 385–397. https://doi.org/10.23887/janapati.v14i2.84476.

Li, J., Pan, Q., Xie, S., and Wang, S., 2011. A hybrid artificial bee colony algorithm for flexible job shop scheduling problems. *International Journal of Computers, Communications & Control*, 6(2), pp. 286–296. https://doi.org/10.15837/ijccc.2011.2.2177.

Luo, Z., Liu, X., Tan, S., Xu, H. and Liu, J., 2023. Multi-objective multi-stage scheduling optimization algorithm: minimizing cost, time and maximizing quality. *Processes*, 11(4), 1147. https://doi.org/10.3390/pr11041147.

Motair, H.M., 2017. Solving composite multi-objective single machine scheduling problem using branch and bound and local search algorithms. *Al-Mustansiriyah Journal of Science*, 28(3), pp. 200–208. https://doi.org/10.23851/mjs.v28i3.122.

Neamah, N.M., and Kalaf, B.A., 2024. A hybrid approach for efficiently solving multi-criteria scheduling problems on a single machine. *Iraqi Journal of Science*, 65, pp. 907–922. https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/11215/6471.

Neamah, N.M., and Kalaf, B.A., 2024. Solving tri-criteria: total completion time, total late work, and maximum earliness by using exact and heuristic methods on single machine scheduling problem. *Iraqi Journal of Computer Science and Mathematics*, 5(3), pp. 14–25. https://doi.org/10.52866/ijcsm.2024.05.03.002

Peng, W., Mu, J., Chen, L., and Lin, J., 2021. A novel non-dominated sorting genetic algorithm for solving the triple objective project scheduling problem. *Memetic Computing*, 13, pp. 271–284. https://doi.org/10.1007/s12293-021-00332-x.

Pham, D.T., Castellani, M., 2009. The bees algorithm: modelling foraging behaviour to solve continuous optimization problems. Proceedings of the Institution of Mechanical Engineers, Part C: *Journal of Mechanical Engineering Science,* 223(12), pp. 2919–2938. https://doi.org/10.1243/09544062JMES1494.

Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., and Zaidi, M., 2005. Bee algorithm: A novel approach to function optimization. *Technical Note*, MEC 0501, The Manufacturing Engineering Centre, Cardiff University.

Pinedo, M.L., 2008. Scheduling: Theory, Algorithms, and Systems. *Springer.* https://doi.org/10.1007/978-0-387-78935-4.

Sameer, S.F.O., 2023. Modified multi-criteria decision-making methods to assess classification methods. *Iraqi Journal of Science*, 64, pp. 893–906. https://doi.org/10.24996/ijs.2023.64.2.34.

Xie, J., Gao, L., and Wang, W., 2019. A tabu search-based memetic algorithm for the multi-objective flexible job shop scheduling problem. *In Genetic and Evolutionary Computation Conference Companion (GECCO'19*), pp. 1522–1525. https://doi.org/10.1145/3319619.3326817.

Yang, S., Meng, L., Ullah, S., Zhang, C., Sang, H. and Zhang, B., 2025. MILP modeling and optimization of three-stage flexible job shop scheduling problem with assembly and AGV transportation. *Chinese Journal of Mechanical Engineering*, 38, article 115. https://doi.org/10.1186/s10033-025-01281-z.

Yousif, S.F., Ali, F.H., and Alshaikhli, K.F., 2023. Using local search methods for solving two multi criteria machine scheduling problems. *Al-Mustansiriyah Journal of Science*, 34(4), pp. 96–103. https://doi.org/10.23851/mjs.v34i4.1430.

Zhang, W., 2025. Metaheuristics for multi-objective scheduling problems in shop environments: A systematic review. *Front. Ind. Eng*. 3:1540022. https://doi.org/10.3389/fieng.2025.1540022.

مناهج البحث المحلي لحل مشكلات جدولة الآلات ذات المعايير الثلاثية والأهداف الثلاثية

زينب ولي مراد[1]، فضاء عثمان سمير[2]*

قسم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق[1]

وحدة الأبحاث البيولوجية للمناطق الحارة، كلية العلوم، جامعة بغداد، بغداد، العراق[2]

الخلاصة

تُعدّ مشكلات الجدولة محورًا أساسيًا في بحوث العمليات نظرًا لتأثيرها المباشر في الإنتاجية، واستغلال الموارد، واستجابة الأنظمة. تتناول هذه الدراسة نموذجًا جديدًا ثلاثيَّ المعايير وثلاثيَّ الأهداف يقدّم إطارًا مبتكرًا لتحليل مشكلات جدولة آلة واحدة المعقّدة، بهدف تقليل زمن الإكمال الكلي ($\sum C_j$)، ومجموع التبكير ($\sum E_j$)، وأقصى تأخير (L_{max})، إضافةً إلى صورتها التجميعية. وتبقى هذه المشكلة ذات تعقيد حسابي مرتفع عند أحجام وظائف كبيرة حتى(n=8000). ولتقييم أداء الحلول، تم اعتماد ثلاث مقاربات مستقلة: خوارزمية الفرع والحد (Branch and Bound – BAB) وطريقتا بحث محلي هما البحث المحظور (Tabu Search)وخوارزمية النحل .(Bees Algorithm) فعلى الرغم من أن خوارزمية BAB توفّر حلولًا دقيقة للحالات الصغيرة (n≤19)مع متوسط خطأ مطلق منخفض(AAE) ، إلا أن زمنها الحسابي يزداد بشكل ملحوظ مع زيادة حجم المشكلة. في المقابل، يحقق البحث المحظور توازنًا بين جودة الحل والجهد الحسابي، إذ ينتج حلولًا قريبة من المثلى مع أزمنة تنفيذ معتدلة للمسائل المتوسطة والكبيرة .(n=500–8000) أما خوارزمية النحل، فرغم أنها قد لا تضاهي دقة الطرائق الدقيقة، فإنها تمتاز بسرعة الحساب وتوليد أنماط متنوعة من الحلول، مما يجعلها مناسبة بشكل خاص للمشكلات واسعة النطاق أو في الحالات التي تكون فيها الموارد الزمنية أو الحسابية محدودة. علاوةً على ذلك، توفّر هذه الدراسة رؤى تطبيقية تدعم اختيار تقنيات الحل المناسبة، مع الأخذ بنظر الاعتبار حجم المشكلة، والموارد الحسابية المتاحة، والتوازن المطلوب بين الكفاءة وجودة الحل.

الكلمات المفتاحية: متفرع ومحدود، خوارزمية النحل، بحث تابو، البحث المحلي، جدولة الآلة الواحدة.