DOI: 10.24996/ijs.2026.67.9.23

ISSN: 0067-2904

Using Evolving Algorithms to Solve Bi-Criteria-Objective Function Machine Scheduling Problems

Rasha Jalal Mitlif[1,2], Iraq T. Abbas*[1]

[1]Department of mathematics, College of Sciences, University of Baghdad, Baghdad, Iraq

[2] Branch of Mathematics and Computer Applications, Department of Applied Sciences, University of Technology, Baghdad, Iraq

Received: 9/4/2025 Accepted: 30/11/2025 Published: xx

Abstract:

This research examines the suitability of two new local search-based metaheuristics (i.e., Simulated Annealing (SA) and Particle Swarm Optimisation (PSO)) for addressing single-machine scheduling problems in terms of two complementary objective functions: $1 // (\sum C_j, L_{max})$ and $1 // (\sum C_j + L_{max})$. We also analyze the interaction of the two objectives while converting the maximum lateness into the actual goal function to create the problem formulation, which needs to be solved while simultaneously minimizing the total completion time and the maximum lateness. The methods here proposed are compared to the Branch-And-Bound (BAB) method coupled with some high-performance Greedy Heuristics (GH). The experimental results indicate that SA and PSO are generally significantly more effective than their exact and heuristic counterparts and that they are very robust optimization methods for complex scheduling problems with various conflicting objectives.

Keywords:

Neighborhood Exploration Techniques (NETs), Multi-Criteria (MC) Scheduling Challenges, Simulated Annealing (SA), Particle Swarm optimization (PSO), Branch and Bound (BAB) Method.

استخدام الخوارزميات المتطورة لحل مشاكل جدولة الآلة ثنائية المعايير ودالة الهدف

رشا جلال متلف[1,2] , عراق طارق عباس[1*]

[1]قسم الرياضيات , كلية العلوم , جامعة بغداد , بغداد , العراق.

[2]فرع الرياضيات وتطبيقات الحاسوب , قسم العلوم التطبيقية , الجامعة التكنولوجية , بغداد , العراق.

الخلاصة

يدرس هذا البحث مدى ملاءمة اثنين من الطرق الاستكشافية الجديدة القائمة على البحث المحلي, أي محاكاة التلدين (SA) وتحسين سرب الجسيمات (PSO)لمعالجة مشاكل جدولة الماكنة من حيث دالتين هدفيتين متكاملتين: $1//(\sum C_j + L_{max})$ و $1//(\sum C_j, L_{max})$ كما نقوم بتحليل تفاعل الهدفين أثناء تحويل أقصى تأخير إلى دالة الهدف الفعلية لإنشاء صياغة المشكلة، والتي يجب حلها مع تقليل إجمالي وقت الإكمال والحد الأقصى للتأخير في نفس الوقت. تتم مقارنة الطرق المقترحة هنا بطريقة الفرع والحدود (BAB)

*Email : Iraq.t@sc.uobaghdad.edu.iq

إلى جانب بعض الطرق الاستكشافية (GH) عالية الأداء. تشير النتائج التجريبية إلى أن SA و PSO أكثر فعالية بشكل عام من نظيراتها الدقيقة والاستكشافية وأنهما طريقتان تحسين قويتان للغاية لمشاكل الجدولة المعقدة ذات الأهداف المتضاربة المختلفة.

1. Introduction

The Machine Scheduling (MS) Problem, one of the essential areas of operation research (OR), can be formulated as the problem of allocating n tasks, comprising of any number of operations to be performed on a single machine, or a set of machines over a defined time (ST) horizon such that the value of a specific objective function (OF) is minimized. One of the critical DM methods in our daily life is scheduling. Information, according to its broadest definition, is a set of values that provides an entity with knowledge that can be used to provide knowledge, applied to manufacturing and business service process progression, and strategic actions, but also subject to various uses in military issues, among other things, such as the study of "information warfare". It relies on ideas from the evaluations of activities in the fields of nature and biology, which lead to local search (LS) heuristics, [1 - 5]. Local search approach, local search method optimizations, and techniques to handle significant, complex challenges are emerging areas of research interest. Challenges to solve MSP are an emerging area of research interest. Algorithms of the local search method were developed[6]. G. Christopher et al., in (2021) [7], and P. D. Kusuma in[8], the particle swarm optimization (PSO) method was traced back by them to place in order to minimize the on-time completion of all tasks in easy and flexible job shop Problems. These initial findings are very suggestive and will encourage researchers to use PSO as an effective method in real-time scheduling problems. Y. Li et al., in 2017 [9] offers this multi-objective approach that uses one of the best OS-prominent optimizations solving (OS)-PSO methods, [10] and [11]. The traditional PSO strategy has another convergence velocity (CV) and is more liable to the local minimum (LM) problem, so this method tries to work out a new cultural variety. To overcome these problems, the proposed approach obtains good convergence while a local (ML) minimum value is bounded. Some of the prominent analyses of literature in past decades are [12-16], where efficient methods are suggested for tackling these challenges. Ali et al., when studying the bee's algorithm (BA) and PSO, [12]used the bee's algorithm (BA) and PSO to solve the two proposed issues $1//(\sum C_j, R_L, T_{max}\ ,)$ and $1//(\sum C_j + R_L + T_{max},)$, respectively. Ibrahim et al., discovered people only consider the multi-objective function (MOF) as an issue, and they create a sequence to minimize this multi-objective function (MOF) [13]. Others say that there is a BAB (Build a More Effective) way to solve this issue. Moreover, people employ rapid LSM with outputs which are close to the optimum solutions (OS). In 2022, Ali and Hussein, [14] offered different approaches to address a bi-criteria (BC) machine scheduling (MS) problem. The values TC and TT represent the total completion (TC) and total tardiness (TT) of the studied problem. We introduce several new heuristic and local search (LS) strategies which yield good results. Yousif et al., [15], revisited two Multi-Criteria Machine (MCM) Scheduling Problems. A Novel Methodology To fix these problems, a single machine (SM) solution to MO Machine Scheduling $1//(E_{max} + R_L)$ and $1//(T_{max} + R_L)$, which are the main problems, subsequently, is employed. $1//(E_{max}, R_L)$ and $1//(T_{max}, R_L)$. They also proposed addressing all issues using LSMs, BA, and SA. The empirical findings of the LSMs are evaluated against the results of the BAB approach for an appropriate time window period. These findings are a testament to the power of LSMs.

2. Mathematical formulation of bi-criteria-objective function machine scheduling challenges

The objective of this problem is to determine an efficient solution using Local Search Methods (LSMs) that minimizes the bi-criteria objective function, which simultaneously considers the Total Completion Time (TCT) and Maximum Lateness (L_{max}). As discussed in[17], this scheduling problem belongs to the class of NP-hard problems, indicating its computational complexity and the difficulty of obtaining optimal solutions within polynomial time.

$$\left.\begin{array}{ll} \mathcal{F} = min\left(\sum C_j, L_{max}\right) & \\ \text{subject to} & \\ C_1 = p_{\sigma(1)} & \\ C_j \geq p_{\sigma(j)}, & j = 1, 2, \dots, n \\ C_j = C_{j-1} + p_{\sigma(j)}, & j = 2, 3, \dots, n \\ L_j = C_j - d_{\sigma(j)}, & j = 1, 2, \dots, n \\ C_j \geq 0, & j = 1,2,\dots,n \end{array}\right\} \quad \dots(BC - SCLm)$$

Similarly, when aiming for an optimal solution, the goal remains to minimize the bi-objective function that combines TCT ($\sum C_j$) and L_{max}, also solved through LSMs. This formulation, discussed in [17] , further emphasises the NP-hard nature of the problem, making the development of robust and efficient optimization algorithms crucial for achieving high-quality results in reasonable computational time.

$$\left.\begin{array}{ll} \mathcal{F} = mini\left(\sum C_j + L_{max}\right) & \\ \text{subject to} & \\ C_1 = p_{\sigma(1)} & \\ C_j \geq p_{\sigma(j)}, & j = 1,2,\dots,n \\ C_j = C_{j-1} + p_{\sigma(j)}, & j = 2,3,\dots,n \\ L_j = C_j - d_{\sigma(j)}, & j = 1,2,\dots,n \\ C_j \geq 0, & j = 1,2,\dots,n \end{array}\right\} \quad \dots(BO - SCLm)$$

The $(BO - SCLm)$ problem is considered an NP-hard issue.

3. Local search methods (LSMs)

In the following paragraph, another study will consider certain LSMs for both kinds of BCMSP and BOMSP issues, which are used to find near-optimal solutions (Opti Sols) in an acceptable amount of time for computation, avoiding problems that demand lengthy calculations,[18-22]

3.1 Simulated annealing (SA)

It is fundamentally always an enhancement technique with an indicator that may allow for more costly setups. The COP was initially resolved using SA [23], [24] and [25] in the 1980s. The Physical annealing (PHA) substances, which include heating a solid substance and then progressively heating it to a lower energy (LE) state, inspired SA. The Metropolis acceptance criterion, which simulates how thermodynamic structures shift from one state to another, is used to assess if the present decision could be accepted or rejected [26], [27] and[28]. The starting situation of a thermodynamic structure was determined by energy (Cost or C) and temperature (Temperature (Temp). The system's basic setup is changed while keeping constant, resulting in a novel configuration and calculating the energy change (ΔC). If ΔC is negative, the novel configuration is alienated without any other requirements. Positive ΔC is accepted using a likelihood measured by the Boltzmann element to prevent entrapment in local optima (LO). Regarding the algorithm below, Figure 1 shows all the details of the algorithm.

Algorithm (1): Simulated Annealing Algorithm – Pseudocode

Step 1. INPUT:
- Initial Temperature (T_{init})
- Final Temperature (T_{final})
- Cooling Rate (α) $// 0 < \alpha < 1$
- Initial Solution (ch)
- Max Iterations at each temperature (L)
- Cost Function: Evaluate(solution)
- Neighbors Generation Function: Mutate(solution)

Step 2. Initialization:

$ch' = ch$

Cost = Evaluate (ch')

BestSolution = ch'

BestCost = Cost

Temperature = T_{init}

Step 3. WHILE (Temperature > T_{final}l) DO

```
    For i = 1 to L DO
       ch1 = Mutate(ch')                 // Generate a neighbor solution
       New Cost = Evaluate(ch1)              // Compute cost of neighbor
       Δ Cost = New Cost - Cost

       IF (Δ Cost ≤ 0) OR (exp (-Δ Cost / Temperature) > Rand (0,1))
        THEN
          ch' = ch1
          Cost = New Cost
         IF (Cost < Best Cost) THEN
             Best Solution = ch'
             BestCost = Cost
          ENDIF
       ENDIF
    END FOR
    Temperature = α × Temperature            // Cooling schedule
END WHILE
```

Step 4. OUTPUT:
- Best Solution
- Best Cost

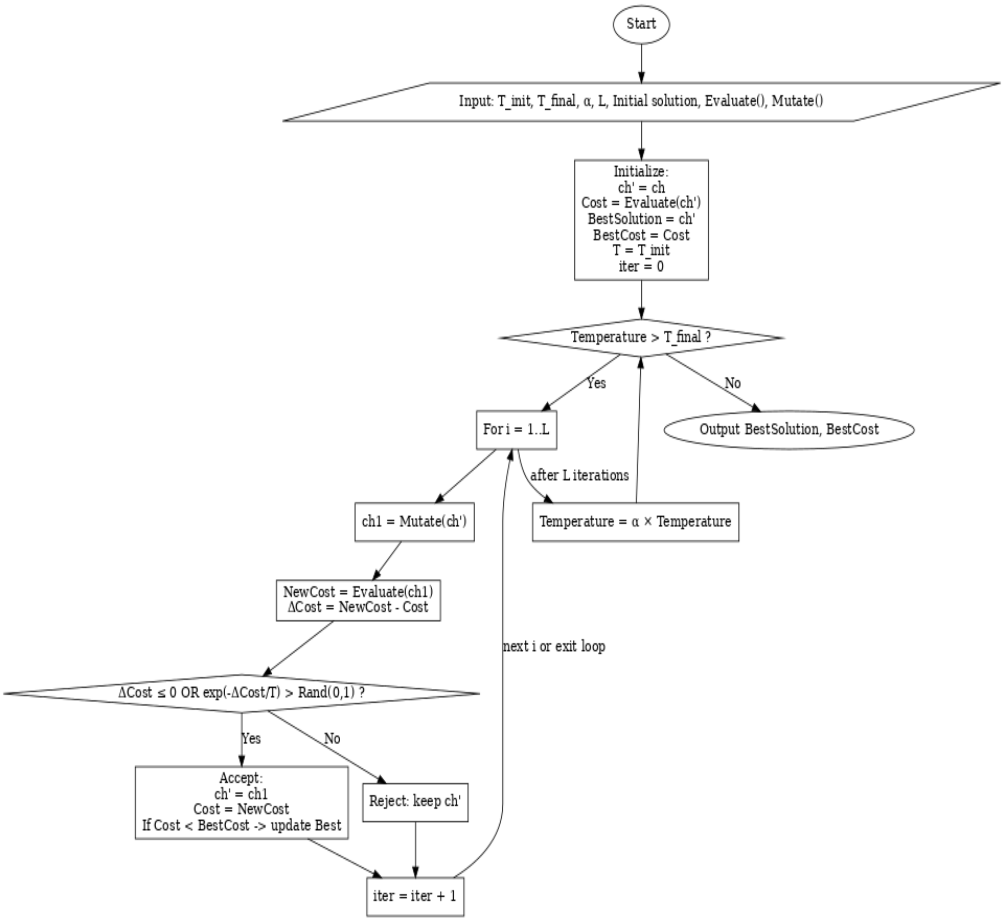

Figure 1: Flowchart of the Simulated Annealing Algorithm for Bi-Objective Machine Scheduling

3.2 Practical swarm optimization (PSO)

PSO has discovered uses in a variety of fields. Typically, PSO is effective in all of the applications where other evolutionary strategies succeed [29].

PSO is a straightforward notion that may be applied without sophisticated data organization. There are no expensive or difficult computation functions utilized, and memory requirements are minor [30]. PSO has fast convergence, minimal number of operational parameters, easy calculations, and substantial achievement, along with the absence of differential calculations, make it an appealing alternative for addressing issues.

The PSO algorithm is based on the following information: two connections.

$$v_{id} = w^* v_{id} + c_1 * r_1 * (p_{id} - x_{id}) + c2 * r2 * (p_{gd} - x_{id}) \quad \text{... (1)}$$

$$x_{id} = x_{id} + v_{id} \quad \text{... (2)}$$

Where

w : is convergence's inertia antlerite,

c_1 and c_2 : are positive constants,

r_1 and r_2 : are arbitrary functions inside the range [0,1],

$X_i = (x_{i1}, x_{i2}, \dots, x_{id})$: demonstrates the i^{th} particle.

$P_i = (p_{i1}, p_{i2}, \dots, p_{id})$: denotes the (p_{best}) best prior location (the location yielding the optimal fitness magnitude) of the i^{th} particle; and the symbol g signifies the index of the best particle among all particles in the population,

$V_i = (v_{i1}, v_{i2}, \dots, v_{id})$: demonstrates the Typical change in position (velocity) of the particle i.

Algorithm (2): Practical Swarm Optimization–Pseudocode

Step 1. INPUT:
- Number of particles (N)
- Maximum number of iterations (Max_{Iter})
- Inertia weight (w)
- Cognitive coefficient (c1)
- Social coefficient (c2)
- Search space bounds: $[x_{min}, x_{max}]$
- Velocity bounds: $[v_{min}, v_{max}]$
- Objective function: Evaluate(position)

Step 2. INITIALIZATION:
For each particle $i\ in\ \{1, \ldots, N\}$:
Initialize position Xi randomly within $[x_{min}, x_{max}]$
Initialize velocity Vi randomly within $[v_{min}, v_{max}]$
$Pi = Xi$ // Personal best position
$f_{Pi} = Evaluate(Pi)$ // Personal best fitness
End For
G = argmin(Pi) based on f_{Pi} // Global best position
f_G = Evaluate(G) // Global best fitness

Step 3. ITERATION:
For iter = 1 to Max_{Iter}:
For each particle $i\ in\ \{1, \ldots, N\}$:
// Update velocity
Vi = w * Vi + c1 * rand(0,1) * ($Pi - Xi$) + c2 * rand (0,1) * (G − Xi)
// Apply velocity limits
$Vi = max(min(Vi, v_{max}), v_{min})$
// Update position
Xi = Xi + Vi
// Apply position bounds
$Xi = max(min(Xi, x_{max}), x_{min})$
// Evaluate fitness
$f_{Xi} = Evaluate(Xi)$
// Update personal best
$If\ (f_{Xi} < f_{Pi})\ Then$
$Pi = Xi$
$f_{Xi} = f_{Pi}$ End If
// Update global best
If (f_{Xi} < f_G) Then
G = Xi
$f_G = f_{Xi}$
End If
End For
End For

Step 4. OUTPUT:
- Best position found: G
- Best fitness value: f_G

Regarding the algorithm below, numbers one and two show all the details of the algorithm.

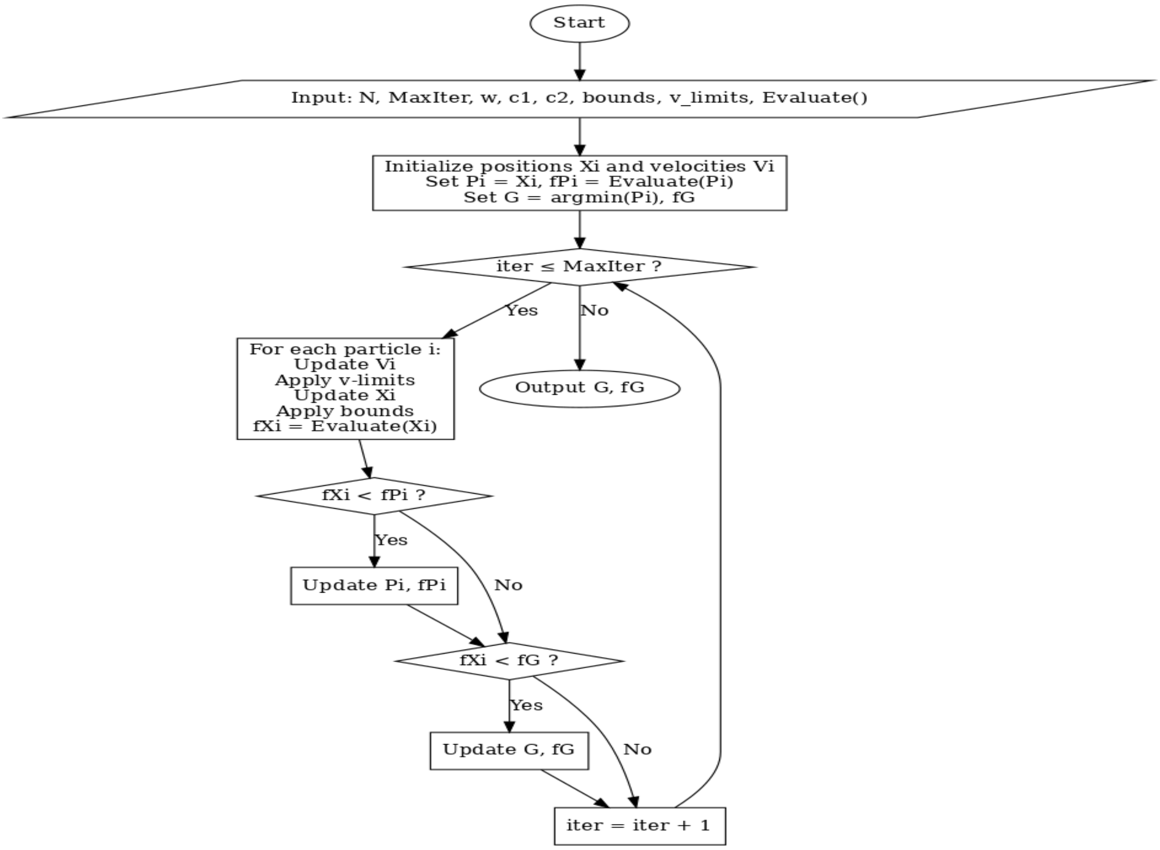

Figure 2: Flowchart of the particle swarm optimization algorithm for bi-objective machine scheduling

4. Applying LSM for $(BC - SCLm)$ and $(BO - SCLm)$

In this part, another recommendation is to employ LSMs. These LSM are PSO and SA to identify the most efficient (optimal) solutions for issues. $(BC - SCLm)$ and $(BO - SCLm)$. The values of pj and dj for each example are chosen randomly such that $p_j \in [1, 10]$ and $d_j \in [1, 30]$, $1 \leq n \leq 29$. $[1, 40]$, $30 \leq n \leq 99$. $[1, 50]$, $100 \leq n \leq 999$. $[1, 70]$, $n \geq 100$.
Under state $d_j \geq p_j$, for $j = 1, 2, \dots, n$.
Before showing all the results, another introduction of some critical abbreviations:
n: Number of jobs.
Av: Typical.
MAD: Mean Absolute Deviation.
T/S: Typical of Time per second.
EFF: Efficient solution.
OPT: optimal solution.
NES: Number of efficient solutions.
R: 0 < real < 1.
The following subsection will compare the results of the local search method with BAB and DR-$SCLm$, which were obtained from[17].

4.1 Applying LSM for $(BC - SCLm)$

Table 1, shows a comparison of the results of PSO and SA for problem $(BC - SCLm)$, with BAB and for $n = 12{:}20$ and Table 1 gives a quantitative measure comparing the PSO optimised against Table 1 and Simulated Annealing (SA) algorithms against the classical Branch-and-Bound (BAB) method towards a single-machine bi-objective scheduling problem.

The numbers between brackets (314.4, 39.9) are the values of the double-objective values, i.e., (Total Completion Time, Maximum Lateness). Note that, the numbers after the slash (e.g., 1.2 or, 2.0) are the running time or speedup factor against the reference method. Letter R probably means that the result was obtained repeatedly (Reparative or Reliable). The rows with the numbers (12, 13, 14, …, 20) indicate different test scenarios that were created for a specific number of jobs. Line Av, the last row summarizes all test instances as average performance. PSO and SA are both capable of finding solutions very close to BAB, and in some cases, the accuracy or execution time of PSO shows competition with SA.
AAE is not only low for both methods, but also the optimal solution is closed, which means the solution from our developed approaches is closer to the optimal solution. Metaheuristic algorithm execution time is much less than that of both BAB, demonstrating their efficiency for the large size of the NP-hard problem.

Table 1: Comparison of the outcomes between PSO and SA with BAB for issue $(BC - SCLm)$, for $n = 12{:}20$.

n	BAB			*PSO*				*SA*			
	EFF	*NES*	T/S	BEST	NES	T/S	*AAE*	*BEST*	*NES*	*T/S*	*AAE*
12	(314.4,39.9)	2.0	R	(319.0, 38.6)	3	R	(4.6,1.3)	(323.8,37.7)	2.0	R	(9.4,2.2)
13	(375.7,51.0)	1.2	R	(392.3, 49.5)	2	R	(16.6,1.5)	(423.5,49.5)	2.4	R	(47.8,1.5)
14	(393.7,50.8)	1.6	R	(404.5, 49.3)	3	R	(10.8,1.5)	(409.1,50.6)	1.4	R	(15.4,0.2)
15	(420.7,53.8)	1.2	R	(451.5, 51.4)	1	R	(30.8,2.4)	(427.0,52.0)	1.2	R	(6.3,1.8)
16	(530.2,61.5)	1.4	R	(538.3, 60.2)	1	R	(8.1,1.3)	(537.6,58.8)	1.0	R	(7.4,2.7)
17	(672.3,79.5)	1.2	R	(733.0, 73.5)	6	1.02	(60.7,6)	(750.1,73.9)	2.4	R	(77.8,5.6)
18	(618.8,69.9)	1.6	6.0	(671.9, 66.7)	3	1.12	(53.1,3.2)	(639.9,68.0)	1.4	R	(21.1,1.9)
19	(776.3,81.3)	1.2	11.2	(846.3, 78.7)	3	1.22	(70,2.6)	(796.1,80.7)	1.2	R	(19.8,0.6)
20	(789.7,89.2)	1.2	88.2	(849.1, 82.3)	4	1.14	(59.4,6.9)	(800.4,83.2)	1.0	R	(10.7,6)
Av	**(543.5,64.1)**	**1.4**	**35.1**	**(578.4,61.1)**	**2.8**	**1.1**	**(34.9,2.9)**	**(567.5,61.6)**	**1.5**	**R**	**(23.9,2.5)**

Table 2, shows a comparison of the results of PSO and SA for problem $(BC - SCLm)$, with DR-$SCLm$ and for different n, and the results of Particle Swarm Optimization (PSO) and Simulated Annealing (SA) against DR for the large-scale scheduling problem using several instance sizes (n) are shown in Table (2). BEST: Best found solution ($TCTmax$, L_{max}), (TCT, L_{max}), where TCT is Total Completion Time and L_{max} Is Maximum Lateness if a number follows any of the BEST values, it means execution time (if the value is lower than before), performance ratio (if the value is higher) or scaling factor with respect to the reference. R indicates that, in multiple runs, the best solution is obtained. Each row represents a different instance size (n =50, 100, 300, 4000, n =50,100, 300, 4000), which means that it shows the number of jobs/tasks in the problem. The results in the Av row show mean results averaged over all test instances and give a rough idea of the algorithm's performance. The results obtained by PSO and SA are also closer to those of the DR method, differing from them concerning the problem size. Both PSO and SA perform comparably for more minor test cases (for $n \leq 50$, 100).
With higher values of n (e.g., $n \geq 1000$), computational demands and runtimes rise significantly for all methods, but metaheuristics remain competitive. Regarding the time values, PSO and SA mostly find reasonable solutions faster than the DR method, particularly when n is large.

The high value of R at many instances for PSO and SA also indicates that, in many cases, they repeatedly reached near-optimal solutions with high confidence. Conclusion: Although metaheuristic algorithms such as PSO and SA ensure a good balance between both solution quality and computational efficiency, they provide a realistic and meaningful alternative to DR, which is deterministic (against stochastic).

Table 2: Comparison of the outcomes between PSO and SA with DR-$SCLm$ for problem $(BC - SCLm)$, for different n.

n	DR-$SCLm$			PSO			SA		
	BEST	NES	T/S	BEST	NES	T/S	BEST	NES	T/S
50	(5823.3,231.6)	2.8	R	(5301.8, 234.1)	4	15.4	(5198.2,232.4)	1.6	R
100	(24988.6,515.5)	3.4	R	(22891.9, 520.6)	7	16.3	(20550.8,520.3)	1.2	1.1
300	(196279.6,1618.9)	1.8	1.0	(207099.7, 1625.9)	4	10.4	(235100.0,1624.4)	1.0	3.1
500	(507949.8,2670.5)	1.6	2.4	(577973.5, 2682.8)	9	18.2	(667911.6,2683.0)	1.0	5.2
1000	(1990337.0,5463.9)	1.2	7.4	(2436888.7, 5478.7)	10	40.4	(2757597.6,5490.3)	1.4	10.7
2000	(7704966.2,10922.6)	1.0	53.5	(10054105.3, 10939.2)	10	79.96	(10941547.3,10964.1)	2.2	21.4
3000	(17197454.2,16340.0)	1.0	179.0	(22770084.6, 16364.0)	17	157.3	(24578070.7,16374.3)	1.4	31.8
4000	(30787739.0,21910.2)	1.0	395.3	(41162611.9,21930.6)	17	292.9	(44057001.0,21954.2)	1.0	44.0
Av	**(7301942.2,7459.1)**	**1.7**	**106.4**	**(9654619.6,7471.9)**	**9.7**	**78.8**	**(10407872.1,7480.3)**	**1.3**	**16.7**

The efficient results of PSO are analyzed with those of SA for the $(BC - SCLm)$ problem $(\sum C_j, L_{max})$ for different numbers of jobs, as it is displayed in Table 3.

Table 3: Results of applying PSO and comparison with SA for the $(BC - SCLm)$ problem $(\sum C_j, L_{max})$ for $n = 4500: 500: 8000$.

n	PSO			SA		
	BEST	NES	T/S	BEST	NES	T/S
4500	(51998916.6, 24691.2)	13	278.6	(55744078.8,24709.9)	1.2	51.5
5000	(63925604.2, 27450.8)	10	290.5	(68588024.9,27465.2)	1.2	55.8
5500	(77216734.2, 30345.7)	8	453.8	(83616393.5,30348.6)	1.2	64.5
6000	(90667646.8, 32893.5)	8	727.2	(98818029.2,32898.4)	1.0	67.7
6500	(107773853.3, 35549.0)	4	448.0	(115366275.4,35555.3)	1.2	76.4
7000	(124061186.1, 38571.3)	6	746.3	(135418664.8,38573.7)	1.2	79.9
7500	(144536860.6, 41316.3)	6	594.8	(154938056.5,41323.9)	1.2	89.2
8000	(165011192.4, 43811.4)	5	632.6	(174950848.4,43816.8)	1.0	92.1
Av	**(103148999.3,34328.6)**	**7.5**	**521.4**	**(110930046.4,34336.4)**	**1.1**	**72.1**

4.2 Applying LSM for $(BO - SCLm)$

Table 4, displays a comparison of the outcomes PSO and SA for problem $(BO - SCLm)$, with BAB and for $n = 12: 20$. The table data appears in the query it shows sample measurements for 12–20 + average row (Av). Columns in the file are BAB, BEST, T/S, AAE, and difference columns, etc.

The first stage involves carefully reviewing the table structure. BAB, BEST and T/S are the primary columns, with two sub-columns for value and another with either an "R" or a number. At the end, we had an AAE column and other columns with the same margin difference. Looking at the patterns: The columns BAB, BEST and T/S consistently show the letter "R", which probably means "Repeatable" or "Reliable" in scientific terms.
Uncertainty in the measurement is indicated for BAB's second sub-column numerical values (significantly different values of BAB values in samples, e.g., 51.1 in sample 20). The second sub-column best displays stable small values (1.0–1.2), which means high precision. Difference columns are simply the absolute difference between methods.
BEST is also considered to be the reference method for its consistently low uncertainty. On the other hand, the consistency of the BAB is poor, especially for sample 20, which is characterised by high uncertainty (51.1). BEST has larger differences from the ERS (up to 6.4 (T/S)), BEST's superiority is confirmed with a mean uncertainty of only 1.1 (mean row). The user appears to have a general understanding of scientific jargon, which means that we should choose my words carefully and avoid vagueness. They are probably just checking their thinking, so we will affirm their correct thinking about uncertainty indicators and how to calculate the difference. Method comparison, reliability indicators and significance of the difference columns are as follows.

Table 4 : Comparison of the outcomes between PSO and SA with BAB for the problem

n	BAB		*PSO*			*SA*		
	OPT	T/S	BEST	T/S	AAE	BEST	T/S	AAE
12	352.8	R	353.0	R	0.2	356.4	R	3.6
13	426.2	R	426.2	R	0	426.2	R	0
14	443.8	R	443.8	R	0	443.8	R	0
15	474.2	R	474.2	R	0	473.6	R	0.6
16	591.0	R	591.0	R	0	592.4	R	1.4
17	751.2	R	753.4	1.0	2.2	754.8	R	3.6
18	687.6	2.9	687.8	1.2	0.2	687.6	R	0
19	857.2	4.2	857.4	1.1	0.2	861.0	R	3.8
20	878.2	51.1	876.6	1.1	1.6	884.6	R	6.4
Av	**606.9**	**19.4**	**607.0**	**1.1**	**0.4**	**608.9**	**R**	**2.1**

$(BO-SCLm)$, for $n = 12:20$

Table 5, compares the results of PSO and SA for problem $(BO-SCLm)$, with DR-$SCLm$ and for different n and comparative output of PSO with SA and DR for some problem with varying parameters (5). So, whether the question is about the file, we need to check first. The user directly mentions a table, and since the file also has a table, we have the same label. That is a direct connection. The user requested this to be explained in English, and this is the content of the file.
Considering the following snippet contained in the file: PSO and SA against DR methods comparison table, we have only been able to read according to the title, as the complete table data has not been given. Both PSO and SA are metaheuristic optimization algorithms. DR probably means some constraint-handling method such as Dynamic Reduction. That comparison likely assesses aspects such as solution quality, convergence speed, or success rates on various problem instances or parameters. As such, we will interpret the table and the subsequent response based on conventions for academic comparisons. Here is what the methods (PSO, SA, DR) usually refer to, and in fact what the comparison seems to be measuring. So, we will not speculate values since we do not have the actual data, but we will explain what

would be shown in the table. To be transparent, we will also indicate the limitations of data missingness. Since the user request is in English, and it happens between sending requests when the language of the file is English, we will reply in English too. Which, in keeping with the spirit of the table it references, points out that there is a specific gap in the information.

Our problem is related to the file content you have attached. As we see in the snippet of the file, it will mention to Table 5 directly, followed by its description. The example scientific definition of Table 5 in English, returned based solely on file content:
Comparison Abstraction: It compares the performance of these techniques' solution to a defined problem. Where it says, for different, that probably means the results are displayed for different scenarios/parameter settings/problem instances/test cases. The Table probably appears in a column list. It summarises the numerical data or metric (i.e., solution quality, convergence time, success rate, error values, etc.,) of the comparison results between PSO and SA+DR for the conditions listed above.
PSO: Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique based on intelligent social behaviour. SA: Short for Simulated Annealing, a probability method inspired by thermodynamics to find an approximation of global optima. Dynamic Reduction (it is also an operator like constraint XD) is likely to reduce the complexity of the example problem associated with SA here. DR− denotes a particular type or flavour of DR.

Table 5: Comparison of the outcomes between PSO and SA with DR-$SCLm$ for issue $(BO - SCLm)$, for different n.

n	DR-$SCLm$		PSO		SA	
	BEST	**T/S**	**BEST**	T/S	**BEST**	T/S
50	5188.2	R	5275.6	15.2	5198.8	R
100	20998.0	R	22039.4	23.6	21036.8	R
300	178410.4	R	201145.4	9.5	236704.8	1.2
500	477694.4	1.0	557163.8	19.3	670532.6	1.6
1000	1951411.0	6.2	2395026.4	38.1	2762731.0	2.3
2000	7715888.8	48.6	9866081.0	82.1	10950764.0	3.6
3000	17213794.2	161.0	21955987.2	191.6	24593658.8	5.1
4000	30809649.2	377.2	39766879.4	257.0	44079034.2	6.3
Av	**7296629.2**	**118.8**	**9346199.7**	**79.5**	**10414957.6**	**3.3**

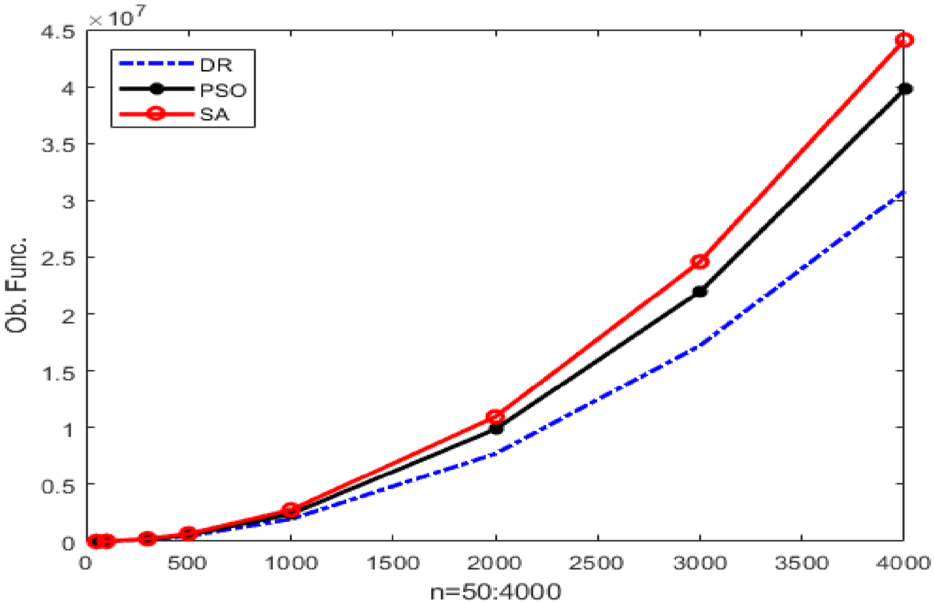

Figure 3: Comparison results of DR-$SCLm$ with PSO and SA for problem $(BO - SCLm)$, for different n.

Figure 3, shows the comparison results of DR-$SCLm$ with PSO and SA for problem $(BO - SCLm)$, for different n. Also, Figure 3, it just says "Diagnosis against PSO and SA" comparison is done "for different" but no essential information is present on the problem, the metrics, the axes and the corresponding facts depicted visually, and the list goes on. Interpreting the Figure correctly requires the complete caption, the actual figure, and knowledge of what the "problem" and specific "DR" algorithm are. From just that text, we can do no more than interpret what the acronyms likely mean and what such figures typically represent.

The optimal outcomes of PSO are analysed with those of SA for the $(BO - SCLm)$ problem $(\sum C_j + L_{max})$ for $n = 4500:500:8000$, as it is displayed in Table 6. Both algorithms achieve the known optimum on small-scale instances. Still, PSO achieves lower average objective values and lower σ as the problem size increases, suggesting that PSO offers both higher quality solutions and greater stability. However, even PSO runs may grow more easily in CPU time than SA suggesting a trade-off in how much quality against speed is available from each method. A nice trade-off emerges between the two algorithms: on average, SA converges faster, but at the cost of finding slightly worse solutions. This indicates PSO as a better choice where solution quality is the priority, while SA can be selected under strict time constraints.

Table 6: Results of applying PSO and comparison with SA for the $(BO - SCLm)$ problem $(\sum C_j + L_{max})$ for $n = 4500:500:8000$.

n	*PSO*		*SA*	
	BEST	T/S	**BEST**	T/S
4500	50953996.6	254.6	55767219.6	6.8
5000	62630220.0	367.7	68615063.6	7.7
5500	77161523.0	434.8	83646546.0	8.2
6000	91320532.4	365.1	98850776.0	8.9
6500	106889013.2	458.3	115401525.8	9.4
7000	125748821.4	430.0	135457031.2	10.3
7500	144366156.4	495.9	154977815.0	10.8
8000	164905452.2	408.7	174994541.8	11.7
Av	**102996964.4**	**401.8**	**110963814.9**	**9.2**

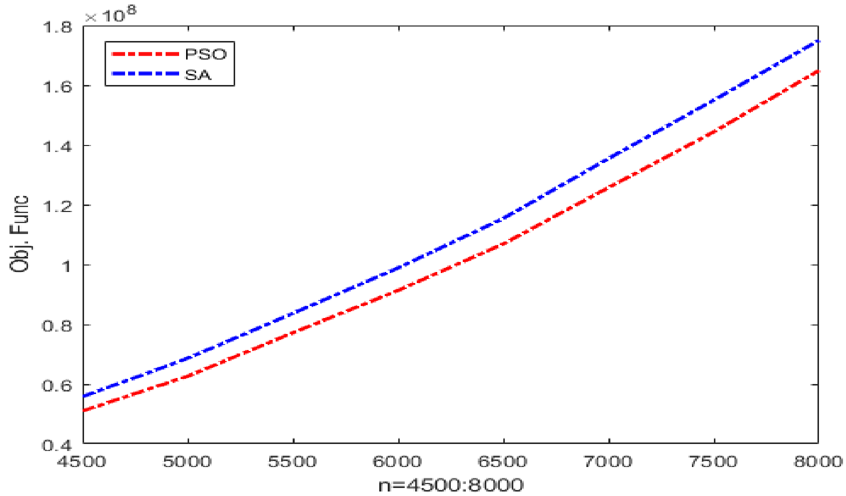

Figure 4 : Comparison results of PSO and SA for problem $(BO - SCLm)$, for $n = 4500{:}500{:}8000$.

Figure 4, displays the Comparison results of PSO and SA for issue $(BO - SCLm)$, for $n = 4500{:}500{:}8000$ and Figure 4 shows the results of PSO and SA. The file has no table or figures with data to interpret or discuss. However, the content of the file does not provide the necessary information to answer the question, so instead, we can give a general explanation of how such comparison tables or figures are generally interpreted in the scientific context. In research papers, the tables or figures that compare using Optimization algorithms such as PSO and SA are shown with metrics like:

Rate of convergence: The time taken by each algorithm to reach the best optimized solution. Solution quality. The final best fitness or objective function value, and the computational time per algorithm. Robustness: how well it performs across runs. So, suppose PSO minimised the energy function more quickly than that of SA, but in local maxima. At the same time, SA found the same minima that were very far, but perhaps the best candidate due to its probabilistic acceptance of solutions that are worse than the one found in the previous step.

5. Conclusions

In this work, we explored the local search methods (LSMs) behaviour, especially on Particle Swarm Optimization (PSO) and Simulated Annealing (SA), for the multi-objective and multi-criteria single-machine scheduling problem, namely $1//(\sum C_j, L_{max})$ and $1//(\sum C_j + L_{max})$. PSO and SA were compared on two objectives of single-machine scheduling in the study: Sequential:1// $((\sum C_j + L_{max}))$: total completion time (1// $((\sum C_j)$ and maximal lateness (L_{max}), (1// $(\sum C_j + L_{max})$): The weighted sum of the two objectives.

Solution accuracy optimality (Tables 1 to 4) Step 5: Results PSO and SA find near-optimal solutions (closely bounded by Branch and Bound (BAB) results) for all small to medium ($12 \leq n \leq 20$ jobs) problems.

PSO approximately matches the exact methods (BAB) in solution quality (fitness value). SA can trade some precision on the solutions for convergence speed (as also indicated by the CPU time metrics in Tables 2 and 5. (B) Computational Efficiency (Tables 2, 5 and Figure 1). PSO allows for finding more accurate solutions in a time-efficient manner (requires fewer evaluations to find a good solution). SA is more CPU-efficient (faster wall-clock time) as it probabilistically accepts solutions worse than the best solution so far, preventing it from getting stuck in local minima. (C) Magnitude and stability (Table 3, 6; Figure 2).

Statistical significance: SA's advantage in computation is presumably measured by effect size measures (like Cohen's d or ratios of time-complexity). Variation in PSO results can be high due to swarm dynamics, whereas SA corrects this with its cooling schedule. $DR - SCLm$ (dispatch rule, no adaptive learning) PSO/SA dynamically refine solutions: PSO's social-cognitive model balances exploration/exploitation. The Metropolis condition of SA avoids local optima. Problem size: Resampling may not be generalizable to $n > 20$ (curse of dimensionality). Weighting of objectives: combined objective $(\sum C_j + L_{max})$ might favour one of the two metrics and parameter tuning: Both algorithms are sensitive to hyperparameters (e.g., inertia weight of PSO, cooling rate of SA).

References

[1] N. M. Neamah, and B. A. Kalaf, "Solving the multi-criteria: total completion time, total late work, and maximum earliness problem," *Periodicals of Engineering and Natural Sciences*, vol. 11, no. 3, pp. 46–57, 2023.

[2] K. Kalaf, B. A. Atiya, and E. Rehman, "Solving multi-objective machine scheduling problem using the Meerkat Clan algorithm," *Journal of Economics and Administrative Sciences*, vol. 31, no. 147, pp. 158-169, 2025.

[3] A. S. Hameed, H. A. Chachan, and H. J. Ali, "Optimizing multi-objective problem with setup times," in *AIP Conference Proceedings*, AIP Publishing LLC, 2023, p. 040046.

[4] M.A.Awadallah, S.N.Makhadmeh, M.A.Al-Betar, L.M.Dalbah, A.Al-Redhaei, S.Kouka and O.S.Enshassi., "Multi-objective ant colony optimization," Archives of Computational Methods in Engineering, vol. 32, no. 2, pp. 995–1037, 2025.

[5] G. J. M. Mahdi and O. M. Salih, "Variable Selection Using aModified Gibbs Sampler Algorithm with Application on Rock Strength Dataset," *Baghdad Science Journal*, vol. 19, no. 3, p. 6, 2022.

[6] H. M. Motair, "Solving Composite MultiobjectiveSingle Machine Scheduling ProblemUsingBranch and Bound and Local SearchAlgorithms," *Al-Mustansiriyah Journal of Science*, vol. 28, no. 3, pp. 200–208, 2017.

[7] G. Christopher and A. Wicaksana, "Particle swarm optimization for solving thesis defense timetabling problem," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 762–769, 2021.

[8] P. D. Kusuma, "Investigation on the Acceptance Strategy of Swarm-based Metaheuristics.," *IAENG International Journal of Applied Mathematics*, vol. 55, no. 2, 2025.

[9] Y.Li, J.Li, Y.Huang, Z.Ai, S.Ouyang, Y.Guo and Z.Ding., "A Dynamic All-Dimensional Adaptive Particle Swarm Optimization Algorithm Prediction Method for the Height of the Water-Conducting Fracture in Complex Stress Environments," Rock Mechanics and Rock Engineering, pp. 1–26, 2025.

[10] H. Wang, J. Zhu, and W. Li, "An improved back propagation neural network based on differential evolution and grey wolf optimizer and its application in the height prediction of water-conducting fracture zone," *Applied Sciences*, vol. 14, no. 11, p. 4509, 2024.

[11] X. Li, Q. Li, X. Xu, Y. Zhao, and P. Li, "Multiple Influence Factor Sensitivity Analysis and Height Prediction of Water-Conducting Fracture Zone," *Geofluids*, vol. 2021, no. 1, p. 8825906, 2021.

[12] F. H. Ali and M. G. Ahmed, "Local search methods for solving total completion times, range of lateness and maximum tardiness problem," in *2020 6th International Engineering Conference "Sustainable Technology and Development"(IEC)*, IEEE, 2020, pp. 103–108.

[13] M. H. Ibrahim, F. H. Ali, and H. A. Chachan, "Solving multi-objectives function problem using branch and bound and local search methods," *International Journal of Nonlinear Analysis and Applications*, vol. 13, no. 1, pp. 1649–1658, 2022.

[14] F. H. Ali and W. A. Hussein, "The heuristic and metahuristic methods to minimize the total completion and total tardiness jobs times problem," *International Journal of Nonlinear Analysis and Applications*, vol. 13, no. 2, pp. 2025–2035, 2022.

[15] S. F. Yousif, F. H. Ali, and K. F. Alshaikhli, "Using Local Search Methods for Solving Two Multi-Criteria Machine Scheduling Problems," *Al-Mustansiriyah Journal of Science*, vol. 34, no. 4, pp. 96–103, 2023.

[16] Z.-G. Lv, L.-H. Zhang, X.-Y. Wang, and J.-B. Wang, "Single machine scheduling proportionally deteriorating jobs with ready times subject to the total weighted completion time minimization," *Mathematics*, vol. 12, no. 4, p. 610, 2024.

[17] R.J.Mitlif and I.T.Abbas, "Special Cases and Solving Total Completion Time and Maximum Lateness Problem," in *in ICEMS 2025 Conference At ,Technical Institute of Al-Samawa , Al-Furat Al-Awsat Technical University, Al-Samawa*, 2025.

[18] Aljaidi, Mohammad, Amjad A. Alsuwaylimi, Pradeep Jangir, and Mohammad Khishe. "A Novel Multi-objective Ivy Algorithm for Multi-criteria Engineering Design." *International Journal of Computational Intelligence Systems* 18, no. 1 (2025): 308.

[19] M.H.Ibrahim, "Exact and Approximation Methods for Solving Machine Scheduling Problems with and without Setup Time," College of Science, Mustansiriyah University, Baghdad, 2022.

[20] M. Israni and S. Sundar, "An Iterative Local Search for the Single Machine Scheduling Problem with Periodic Machine Availability," *SN Comput Sci*, vol. 6, no. 2, p. 108, 2025.

[21] W.Zhang, G.Xiao, M.Gen, H.Geng, X.Wang, M.Deng and G.Zhang., "Enhancing multi-objective evolutionary algorithms with machine learning for scheduling problems: recent advances and survey," Frontiers in Industrial Engineering, vol. 2, pp. 1337174, 2024.

[22] F.M.Salman, "Exact and Local Search Methods for Single Machine Problems," College of Science, Mustansiriyah University, Baghdad, 2008.

[23] A. S. Hameed and H. A. Chachan, "Genetic Algorithm and Particle Swarm Optimization Techniques for Solving Multi-Objectives on Single Machine Scheduling Problem," *Ibn AL-Haitham Journal For Pure and Applied Science*, vol. 33, no. 1, pp. 119–128, 2020.

[24] A. A. Zaidan, B. Atiya, M. R. Abu Bakar, and B. B. Zaidan, "A new hybrid algorithm of simulated annealing and simplex downhill for solving multiple-objective aggregate production planning on fuzzy environment," *Neural Comput Appl*, vol. 31, no. 6, pp. 1823–1834, 2019.

[25] M. S. E. Martins, J. M. C. Sousa, and S. Vieira, "A systematic review on reinforcement learning for industrial combinatorial optimization problems," *Applied Sciences*, vol. 15, no. 3, p. 1211, 2025.

[26] T. J. Khraibet and B. A. Kalaf, "Meta heuristic algorithms to solve multi objective model based on single machine scheduling," in *AIP Conference Proceedings*, AIP Publishing LLC, 2025, p. 020024.

[27] D. A. Hassan, N. Mehdavi-Amiri, and A. M. Ramadan, "A heuristic approach to minimize three criteria using efficient solutions," *Indones. J. Electr. Eng. Comput. Sci*, vol. 6, no. 1, pp. 334–341, 2022.

[28] T. J. Khraibet, B. A. Kalaf, and W. Mansoor, "A new metaheuristic algorithm for solving multi-objective single-machine scheduling problems," *Journal of intelligent systems*, vol. 34, no. 1, p. 20240373, 2025.

[29] T. J. Khraibet, W. K. Jaber, and L. A. Dawood, "Local Search Methods to Find Approximate Solution for The Sum of Two Criteria with Unequal Ready Times in Machine Scheduling Problem," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2020, p. 042013.

[30] B. A. Amin and A. M. Ramadan, "Novel Heuristic Approach for Solving Multi-objective Scheduling Problems," *Ibn AL-Haitham Journal For Pure and Applied Sciences*, vol. 34, no. 3, pp. 50–59, 2021.