Faisal et al. Iraqi Journal of Science, 2025, Vol. 66, No. 11, pp: 5015- 5024
DOI: 10.24996/ijs.2025.66.11.26

ISSN: 0067-2904

Using Swarm Intelligence to Solve Bicriteria and Biobjective Machine Scheduling Problems

Safanah Y. Faisal*, Manal G. Ahmed, Manal H. Ibrahim and Faez H. Ali
Mathematics Department, College of Science, Mustansiriyah University, Bagdad, Iraq

Received: 12/5/2024 Accepted: 21/8/2024 Published: 30/11/2025

Abstract

In this paper, solving the Bicriteria Machine Scheduling Problems (BCMSP) and Bi-Objective Machine Scheduling Problems (BOMSP) are proposed using Swarm Intelligence (AI) represented by Particle Swarm Optimization (PSO). The discussed BCMSP is a single machine with maximum early job time and range of late jobs $(1//(E_{max}, R_L))$, and the BOMSP is $1//(E_{max} + R_L)$. Comparison results of a simulation for exact (complete enumeration and Branch and Bound), heuristic method, and simulated annealing with proposed PSO has been made. The results prove the good efficiency of PSO in solving the two problems. All the results obtained by constructing simulation programs using MATLAB language.

Keywords: Particle Swarm Optimization, Branch and Bound technique, Range of Lateness Jobs Times, Maximum Early Jobs Time, the Bicriteria Machine Scheduling Problems.

استخدام ذكاء السرب لحل مسائل جدولة الماكنة ثنائية المعايير وثنائية الاهداف

سفانه فيصل يوسف*، منال غسان احمد، منال هاشم ابراهيم وفائز حسن علي
قسم الرياضيات, كلية العلوم ,الجامعة المستنصرية, بغداد, العراق

الخلاصة

في هذا البحث، تم حل إحدى مسائل جدولة الماكنة ثنائية المعايير (BCMSP) وثنائية الاهداف (BOMSP) بأستخدام ذكاء السرب الممثلة بـ (PSO). مسألة (BCMSP) موضوع البحث هي القيمة العظمى لوقت الاعمال المبكرة ومدى زمن الاعمال المتأخرة $(1//(E_{max}, R_L))$, ومسألة (BOMSP) هي $1//(E_{max} + R_L)$. تمت مقارنة نتائج الطرق الدقيقة (العد التام والتفرع والتقيد) والطرق التقريبية وطريقة محاكات التلدين مع نتائج PSO المقترحة. اثبتت النتائج ان الطريقة PSO لها كفاءة جيدة بحل المسألتين. جميع النتائج تم الحصول عليها من خلال بناء برامج محاكاة باستخدام لغة الماتلاب.

1. Introduction

In many applications, such as industrial design, engineering, and commercial activities, where the goals are to optimize profitability, performance, and efficiency or to minimize costs and energy consumption, scheduling is a crucial decision-making process. Numerous manufacturing and service sectors use it. For instance, a smart scheduling algorithm can be used to reduce the cost of production in an industrial operation so that the businesses can

*Email: safanafaysal77@uomustansiriyah.edu.iq

remain competitive [1]. The problem of machine scheduling (MSP), is this: in order to minimize the given objective function, n jobs must be assigned, each requiring the scheduling of one or more activities on one or more machines within a predetermined time frame [2].

In this paper we will try to solve the nonlinear problems $1//(E_{max}, R_L)$ and $1//(E_{max} + R_L)$ using one of the swarm intelligence local search methods (LSMs); particle swarm optimization (PSO). Yousif and Ali [3] discuss these two problems and some theorems and special cases for the suggested problems, then they solve them using exact (complete enumeration method (CEM) and Branch and Bound (BAB)), some heuristic methods. Then they solve the same two problems using two LSM's (Simulated Annealing (SA) and Bees Algorithm (BA)).

In the last 20 years, many researchers have been solving many different MSPs. Abdul-Razaq and Ali (2015) [4], in their paper attempt to minimize a function of three cost criteria for scheduling jobs on a single machine $1//(\sum C_i + \sum T_i + T_{max})$. They suggested some LSM's to solve the multiobjective problem. Abdul-Razaq and Mohammed (2016) [5], in their paper considering the problem of multiobjective scheduling jobs $1//(\sum F_i + E_{max} + T_{max})$, they proposed Tree Type Heuristic (TTH) Method and Variable Neighborhood Descent (VND) algorithm to solve the problem efficiently. Tariq and Akram (2017) [6] solve the problem $1//\sum C_j + \sum T_j + \sum E_j + T_{max} + E_{max}$, by applying the LSM; descent method (DM) and simulated annealing method (SA) for $n \leq 5000$ jobs. Abdul–Razaq and Motair (2018) [7] they consider a single MSP to minimize four cost functions (multi-criteria); $1//(\sum C_j, \sum T_j, T_{max}, E_{max})$, they minimized the four cost functions simultaneously and proposed a local search algorithm to find the set of efficient solutions for the discussed problem.

Ibn Al-Haitham First International Scientific Conference

Chachan and Hameed (2019) [8] studied the problem $1//\sum_{j=1}^{n} \sum (C_j + T_j + E_j + V_j)$ by using the genetic algorithm (GA) and the PSO. Chachan and Jaafar (2020) [9] present BAB method to minimize the following MSP with unequal release dates: $1/r_j/\sum C_j + \sum T_j + \sum E_j + \sum U_j + \sum V_j$, they proposed six heuristic methods for account Upper Bound (UB). Mahmood and Khalil (2021) [10] studied the problem $1//\sum(C_j + E_{max} + L_{max})$, they applied two methods of LSM which are DM and SA to get the near optimal solution for this problem. Abbas (2022) [11] studied the problems $1//F(\sum U_i, \sum T_i, T_{max})$ and $1//(\sum U_i + \sum T_i + T_{max})$ for single machine, which is NP-hard problem, she used some LSMs; DM, SA and GA.

In this work, the MSP concept is discussed in section 2. The PSO is discussed in section 3. In section 4 we introduce the mathematical formulation of the BCMSP and BOMSP for two suggested functions, the maximum earliness (E_{max}) and the range of lateness (R_L). In section 5, applying the PSO for the BCMSP and BOMSP problems and compare the results with exact and heuristic methods. The analysis and discussion of the results are introduced in section 6. Lastly, the suggestions and recommendations are introduced in section 7.

2. Machine Scheduling Problem Concept

The first part of this section had some significant notes and concentrated only on performance metrics, omitting any information about the system environment. Presumably, there are n jobs. Here, we simply describe the notations that are applied to a single machine, jobs j, $(j = 1, \ldots, n)$ has [12]:

p_j : the processing time of job j.
d_j : A due date, or the date by which the job should be completed; completion of the job after its due date is permitted, but a penalty is imposed.
s_j: $s_j = d_j - p_j$, its called the slack time of job j.
C_j : The time at which the processing of a job j is completed is called the completion time, s.t. $C_j = \sum_{k=1}^{j} p_k$.
Now let's have the sequence σ of jobs we have:

- The lateness $L_j = C_j - d_{\sigma(j)}$.
- The earliness $E_j = max\{-L_j, 0\}$.
- $R_L = L_{max} - L_{min}$ where $L_{max} = \max_{1\le j\le n}\{L_j\}$, $L_{min} = \min_{1\le j\le n}\{L_j\}$.
- $E_{max} = \max_{1\le j\le n}\{E_j\}$.

In our paper, we need the following rules:

Definition (1): Shortest Processing Time (SPT) rule [13]: The problem $1//\sum C_j$ is solved by sequencing all jobs in non-decreasing order of the processing times (p_j) i.e. $(p_1 \le p_2 \le \cdots \le p_n)$.

Definition (2): Minimum slack Time (MST) rule [14]**:** The problem $1//E_{max}$ is solved by sequencing all jobs in non- decreasing order of slack time (s_j) i.e. $(s_1 \le s_2 \le \cdots \le s_n)$.

3. Particle Swarm Optimization

Originally created in 1995 by Kennedy and Eberhart, so originated from previous work with algorithms that recreated the "flocking behavior" that several bird species are known for. When a bird is attracted to a roosting location, it will first begin to fly in spontaneously formed flocks and randomly unless one of the birds crosses the roosting region. In the field of computational intelligence, PSO has grown in popularity. This optimization process is included in the category of soft computing, which also includes evolutionary and genetic computing algorithms [15] [16].
The following two relationships are the focus of the PSO algorithm's core procedure:

$$v_{id} = w * v_{id} + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id}) \quad (1)$$

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

Where
w : is the inertia weight for convergence,
c_1 and c_2 : are positive constants,
r_1 and r_2 : are random functions in the range [0,1],
$X_i = (x_{i1}, x_{i2}, \dots, x_{id})$: represents the i^{th} particle;
$P_i = (p_{i1}, p_{i2}, \dots, p_{id})$: represents the (pbest) greatest prior standing (the standing that had the highest fitness value) of the i^{th} particle. The best particle index among all the particles in the population is represented by the symbol g.
$V_i = (v_{i1}, v_{i2}, \dots, v_{id})$: is the components of the velocity or speed of the particle i [17].

PSO algorithm

Step(1). Generate the initial states of each particle in the population with random velocities and positions with d-dimensions according to dimension of the problem space.
Step(2). The main operations of PSO includes:
a. Evaluate the fitness function in d dimension for each variable for each particle in the population.

b. Compare the fitness value (fv) with its $pbest$. If fv is better than $pbest$, then $pbest = fv$, and $p_{ij} = x_i$ which the current location.
c. Find the best particle in the population, then assign it with g.
d. Update the position and velocity for each particle using equations (1) and (2).
Step(3). Return to step (2) until the end criterion is satisfied.

4. Mathematical Formulation of the BOMSP and BCMSP for MSP

The BCMSP can use the two fields to process a single job at a time. categorizing, the discussed BCMSP denoted by $1//(E_{max}, R_L)$ .and BOMSP $1//(E_{max} + R_L)$. In this paper, the set of the efficient (optimal) solutions wants to be found for the BCMSP (BOMSP), which can be composed for a specific schedule $\sigma = (1,2, \dots, n)$ as:

$$\left.\begin{array}{ll}
F = min\ (E_{max}, R_L) \text{ or } (E_{max} + R_L) & \\
\text{s.t.} & \\
C_1 = p_{\sigma_{(1)}} & \\
C_j = C_{j-1} + p_{\sigma_{(j)}}, & j = 2,3, \dots, n \\
L_j = C_j - d_{\sigma_{(j)}}, & j = 1,2, \dots, n \\
R_{L(\sigma)} = L_{max(\sigma)} - L_{min(\sigma)} & \\
E_j \geq 0, & j = 1,2, \dots, n \\
E_{max}(\sigma), R_L(\sigma) \geq 0 &
\end{array}\right\} \quad \dots (ER)$$

Notice that $E_{max}(\sigma)$ can solved by MST rule [2] and $R_L(\sigma)$ is NP-hard problem [18], then BCMSP-ER and BOMSP-ER are NP-hard too.

5. Using PSO for Solving BCMSP-ER and BOMSP-EPR

In this section, we will compare the results of PSO with methods like CEM, BAB and MST-SPT-ERL Method, and compare them with Simulated Annealing (SA) [3].
The CEM and BAB are exact methods to obtain efficient (optimal) solution(s) for the two problems, but they are limited to some number of jobs (n).

The MST-SPT-ERL is an approximation method that depends on MST and SPT rules to generate a new solution with shifting.
SA is a trajectory-based optimization technique called Simulated Annealing, or SA. It is simply a ongoing development technique with an occasional acceptance requirement for higher cost configurations. SA is considered to be a good tool for imprecise optimization problems. [19] [20].
We generate the values of p_j and d_j for all example randomly s.t. $p_j \in [1,10]$ and,

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

With condition that must $d_j \geq p_j$, for $j = 1,2, \dots, n$.
We use the following notations:

AAE	:	Average Absolute Error.
Av	:	Average.
R	:	$R \in (0,1)$.
T	:	Time averaged in per seconds.
f	:	Value of the Objective Function for BC-ER.
g	:	Value of the Objective Function BO-EPR.

NES : No. of efficient solutions.
$ANES$: Average of No. of efficient solutions.
Each example of n is revered for 5 experiments.
.
In all results tables, we compare the results of PSO with CEM, BAB, SPT-MST-ER and SA for the two problems MCMSP-ER and MOMSP-ER.
For BC-ER, the results of PSO which are compared with results of CEM for $n = 4{:}\,11$ are introduced in Table 1

Table 1: Results of PSO compared with CEM for $n = 4{:}\,11$ for the BC-ER.

n	CEM		PSO		
	f	T	f	T	AAE
4	(9.2,9)	R	(9.2, 9)	R	(0,0)
5	(5.8,14.2)	R	(5.8,14.2)	R	(0,0)
6	(5.6,17.2)	R	(5.6, 17.2)	R	(0,0)
7	(6.4,18.2)	R	(6.4, 18.2)	R	(0,0)
8	(3.6,18.6)	R	(3.6,18.6)	R	(0,0)
9	(3.4,20)	6.8	(3.4,20)	R	(0,0)
10	(2.4,33)	85.5	(2.4, 33)	R	(0,0)
11	(6,37.4)	1010.1	(5.2, 38.2)	R	(0.8,0.8)
Av	**(5.3,20.9)**	**137.8**	**(5.2, 21.1)**	$\boldsymbol{R}$	**(0.1,0.1)**

For BO-ER, the results of PSO are compared with results of CEM for $n = 4{:}\,11$ are described in Table 2.

Table 2 : Results of PSO compared with CEM for $n = 4{:}\,11$ for the BO-ER.

n	CEM		PSO		
	g	T	g	T	AAE
4	18.2	R	18.2	R	0
5	20	R	20	R	0
6	22.8	R	22.8	R	0
7	24.6	R	24.6	R	0
8	22.2	R	22.2	R	0
9	23.4	6.8	23.4	R	0
10	35.4	85.5	35.4	R	0
11	43.4	1010.1	43.4	R	0
Av	**26.2**	**137.8**	**26.2**	$\boldsymbol{R}$	**0**

For BC-ER, the results of PSO are compared with results of BAB for $n = 20{:}\,10{:}\,110$ are introduced in Table 3.

Table 3: Results of PSO are compared with BAB for the BC-ER, $n = 20{:}\,10{:}\,110$.

n	BAB		PSO		
	f	T	f	T	AAE
20	(2.8,89.2)	R	(2.8, 89.2)	R	(0,0)
30	(2.8,127.8)	2.3	(3.8, 127.8)	3.4	(1,0.8)
40	(1.6,194.2)	4.4	(1.6, 195)	4.1	(0,0.8)
50	(2,233.6)	9.5	(2.2, 236.4)	6.2	(0.2,2.8)
60	(1,294.8)	16.5	(2.1, 298.8)	9.2	(1.1,4)
70	(2,350.2)	20.4	(2.4, 354.7)	9.3	(0.4,4.5)
80	(1,399)	27.6	(1.1, 403.2)	10.1	(0.1,4.2)
90	(1.2,469.6)	68.6	(1.2, 470.2)	12.6	(0,0.6)
100	(0.8,505.2)	145.9	(1.2, 470.8)	14.6	(0.4,34.4)
110	(1.4,573.6)	173.4	(1.4, 574)	16.8	(0,0.4)
Av	**(1.6,323.7)**	**46.8**	**(2,322)**	**8.6**	**(0.3,10.4)**

For BO-ER, the results of PSO are compared with results of BAB for $\mathrm{n} = 20{:}\,10{:}\,110$ are described in Table 4.

Table 4: Results of PSO are compared with BAB for the BO-ER, $n = 20{:}\,10{:}\,110$.

n	BAB		PSO		
	g	T	g	T	AAE
20	92	R	92	R	0
30	130.6	2.3	130.6	2.5	0
40	195.8	4.4	196.2	4	0.4
50	235.6	9.5	236.4	5.4	0.8
60	295.8	16.5	295.8	8.5	0
70	352.2	20.4	352.6	8.3	0.4
80	400	27.6	403.6	8.5	3.6
90	470.8	68.6	472	12.6	1.2
100	506	145.9	508	15.3	2
110	575	173.4	579.4	13.5	4.4
Av	**325.3**	**46.8**	**326.7**	**7.9**	**1.3**

Figure (1) shows the closed results between the BAB and PSO for BO-ER for $n = 20{:}\,10{:}\,110$.

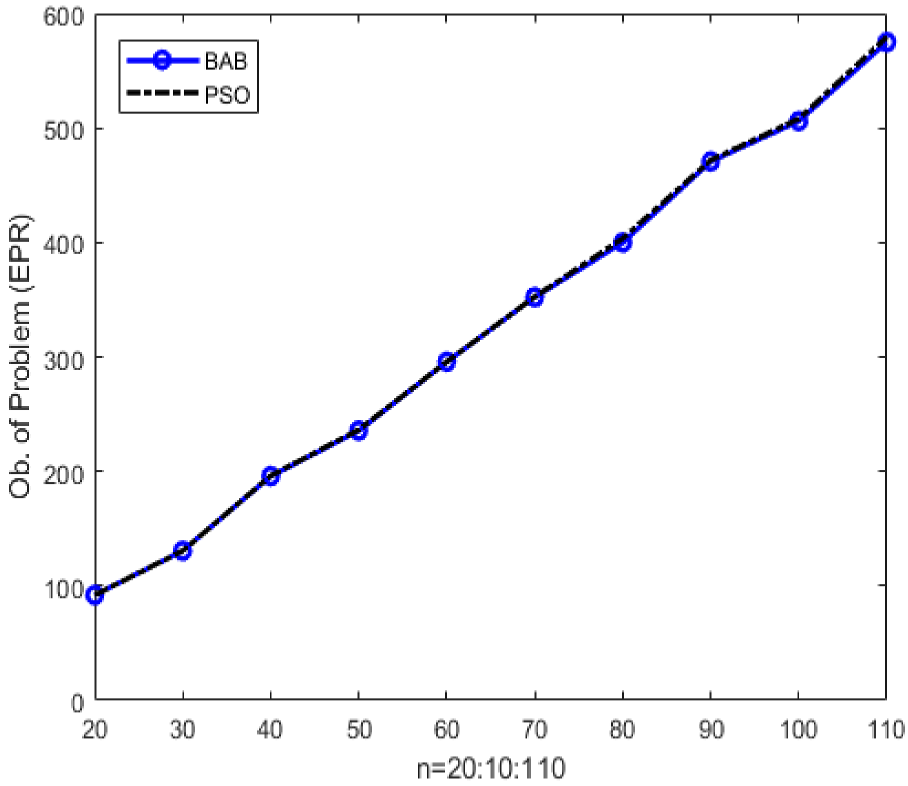

Figure 1: Comparison results between the BAB and PSO for BO-ER for $n = 20{:}\,10{:}\,110$.

For BC-ER, the results of PSO are compared with results of MST-SPT-ERL for $n = n = 4{:}\,11$ are depicted in Table 5.

Table 5: Results of PSO are compared with MST-SPT-ERL for the BC-ER, $n = 4{:}\,11$.

n	**MST-SPT-ERL**		**PSO**		
	f	***T***	***f***	***T***	***AAE***
20	(2.8,91.3)	*R*	(2.9,90.1)	*R*	(0.1,1.2)*
50	(2,235.4)	*R*	(2.2, 236.4)	6.2	(0.2,1)*
100	(0.8,505.6)	*R*	(1.2, 470.8)	14.6	(0.4,34.8)
300	(1,1620.4)	2.2	(2, 1625.1)	53.8	(1, 4.7)
500	(0.2,2680.6)	14.5	(1, 2685.1)	88.6	(0.8, 4.5)
1000	(0.2,5435.6)	15.5	(1.6, 5445.2)	230.1	(1.4,9.6)
2000	(0,10969.8)	122.8	(1.6, 10987.2)	369.3	(1.6,17.4)
5000	(0,27395.8)	1831.6	(2, 27408.7)	508.1	(2,12.9)
Av	**(0.8,6116.7)**	**256.5**	**(1.8,6118.5)**	**158.8**	**(0.9,10.8)**

Note: The sign (*) means the results of PSO is better than other methods.
For BO-ER, the results of PSO are compared with results of MST-SPT-ERL for $n = 20{:}\,5000$ are depicted in Table 6.

Table 6: Results of PSO are compared with MST-SPT-ERL for BO-ER, $n = 20{:}5000$.

n	MST-SPT-ERL		PSO		
	g	T	g	T	AAE
20	93.4	R	92	R	1.4*
50	237.4	R	236.4	5.4	1*
100	506.4	R	508	15.3	1.6
300	1621.4	2.2	1625.4	46.8	4
500	2680.8	14.5	2682	65.7	1.2
1000	5435.8	15.5	5442.2	177.1	6.4
2000	10969.8	122.8	10973.8	329.4	4
5000	27395.8	1831.6	27402.6	824.4	6.8
Av	**6117.6**	**256.5**	**6120.3**	**183.0**	**3.3**

For BC-ER, the results of PSO are compared with results of SA for $n = 5000{:}8000$ are introduced in Table 7.

Table 7: Results of PSO are compared with SA for BC-ER, $n = 5000{:}8000$.

n	SA		PSO		
	f	T	f	T	AAE
5000	(0.0,27395.8)	3.0	(2, 27408.7)	508.1	(2,12.9)
6000	(0.0,32901.8)	3.4	(0.3, 32904.5)	1104.5	(0.3,5.7)
7000	(0.0,38468.4)	3.9	(0.4, 38469.9)	1599.8	(0.0,1.5)
8000	(0.0,44006)	4.4	(0.5, 44007.4)	2790.4	(0.5,1.4)
Av	**(0,35693)**	**3.7**	**(0.8,35697.6)**	**1500.7**	**(0.7,5.5)**

For BO-ER, the results of PSO are compared with results of SA for $n = 5000{:}8000$ are depicted in Table 8.

Table 8: Results of PSO are compared with SA for BO-ER, $n = 5000{:}8000$.

n	SA		PSO		
	g	T	g	T	AAE
5000	27395.8	3.0	27402.6	824.4	6.8
6000	32901.8	3.4	32902.4	1023.7	0.6
7000	38468.4	3.9	38468.8	577.9	0.4
8000	44006.0	4.4	44007.4	1208.2	1.4
Av	**35693**	**3.7**	**35695.3**	**908.6**	**2.3**

6. Analysis and Discussions Results of BC-ER and BO-EPR For BC-ER:

1. Tables (1) and (3) show that the PSO results are identical to the CEM results, and they are very close to the BAB results.
2. According to table (5), PSO's results are very similar to MST-SPT-ER's in terms of CPU-Time, and in some cases, PSO outperforms MST-SPT-ER.
3. As shown in Table (7), the PSO and SA results are very similar.

For BO-ER:

1. From Tables (2) and (4), we can notice that the results of PSO are identical to the results of CEM and for BAB, they are very closed to others. In CPU-time, we see that PSO is better than BAB.

2.From table (6), the results of PSO are very closed to MST-SPT-ER, and in some cases PSO gives better results from MST-SPT-ER (see $n = 20$ and 50).

3. From Table (8) the results of PSO and SA are very closed to each other's.

7. Conclusions and Recommendations

1. In this paper, we apply the PSO for two problems, first is the bi-criteria MSP and the second is Bi-objective MSP.
2. From all results of tables, we can notice the efficiency of PSO in finding good accuracy results comparing with results of exact, approximate and local search methods.
3. In some tables for BC-ER, we see that the PSO gives more than one efficient solution.
4. We suggest using the more new local search methods to solve the two BC-ER and BO-EPR, like ant colony method.
5. As future work, we suggest studying new MSP and solving them by using PSO, like: $1//(T_{max}, R_L)$ and $1//(\sum_{j=1}^{n} E_j , R_L)$.

References

[1] Z. M. Ali, *Single Machine Scheduling Problem by Using Exact and Local Search Methods,* Baghdad: M. Sc., Mathematics department, College of Science, Mustansiriyah University, 2015.

[2] K. R. Baker and D. Trietsch, *Principle of Sequencing and Scheduling, New Jersey, USA: John Willey and Sons Inc.*, 2009.

[3] F. Y. Safanah and H. A. Faez, “Solving Maximum Early Jobs Time and Range of Lateness Jobs Times Problem Using Exact and Heuristic Methods,” *Iraqi Journal of Science,* vol. 65, no. 2, pp. 923-937, 2024.

[4] T. S. Abdul-Razaq and Z. M. Ali, “Minimizing the Total Completion Times, the Total Tardiness and Maximum Tardiness,” *Ibn Al-Haitham Journal for Pure & Applied Science,* vol. 28, no. 2, pp. 155-170, 2015.

[5] T. S. Abdul-Razaq and H. A. Mohammed, “Exact and Approximation Approaches for Solving Multiple Objective Function,” *Al-Mustansiriyah Journal of Science,* vol. 27, no. 3, pp. 89-97, 2016.

[6] T. S. Abdul-Razaq and A. Akram, “Local Search Algorithms for Multi-criteria Single Machine Scheduling Problem,” in *Ibn Al-Haitham First International Scientific Conference, 13-14 Dec.*, Baghdad, Iraq., 2017.

[7] T. S. Abdul-Razaq and H. M. Motair, “Solving Four Cost Multi-Objective Scheduling Problem Simultaneously,” *Journal of Kerbala University,* vol. 16, no. 1, pp. 77-88, 2018.

[8] A. S. Hameed and H. A. Chachan, “Exact methods for solving multiobjective problem on single machine scheduling,” *Iraqi Journal of Science,* vol. 60, no. 8, p. 1802–1813, 2019.

[9] H. A. Chachan and H. A. Jaafar, “Exact Solutions for Minimizing cost Function with Five Criteria and Release Dates on Single Machine,” *Ibn Al-Haitham Jour. For Pure and Applied Science,* vol. 33, no. 3, pp. 140-157, 2020.

[10] A. A. Mahmood and K. N. M., “Local Search and Optimal Algorithms for Simultaneous Multi-Criteria Scheduling Problem on a Single Machine,” *Advance in Mechanics,* vol. 9, no. 3, pp. 1376-1390, 2021.

[11] D. A. Abbas, “Search Method for Solving Multicriteria Scheduling Problem,” *International Journal of Nonlinear Analysis and Applications,* vol. 13, no. 1, pp. 1709-1720, 2022.

[12] M. Pinedo, Scheduling: *Theory, Algorithms and Systems, New York, USA: Springer*, 2008.

[13] W. E. Smith, “Various Optimizers for Single Stage Production,” *Naval Research Logistics Quarterly,* vol. 3, no. 1, pp. 59-66, 1956.

[14] J. A. Hoogeveen, “Minimizing Maximum Earliness and Maximum Lateness on a Single Machine,” in *Proceedings of the 1st Integer Programming and Combinatorial Optimization Conference, pp:283–295*, Amsterdam, 1990.

[15] J. Kennedy and R. C. Eberhart, “Particle Swarm Optimization,” in *Proceedings of IEEE International Conference on NN, Piscataway, pp:1942-1948*, 1995.

[16] Y. Wang, “Research of improved particle swarm optimization algorithm Based on Big Data,” in

International Conference on Robots & Intelligent System (ICRIS), pp:287-290, 2019.

[17] I. K. Ali, *Intelligent Cryptanalysis Tools Using Particle Swarm Optimization,* Baghdad, Iraq: Ph. D., Dept. of Computer Science, University of Technology, 2009.

[18] C. Liao and R. Huang, "An Algorithm for Minimizing the Range of Lateness on a Single Machine," *The Journal of the Operational Research Society,* vol. 42, no. 2, pp. 183-186, 1991.

[19] A. M. Mohsen, "Annealing Ant Colony Optimization with Mutation Operator for Solving TSP," *Hindawi Publishing Corporation, Computational Intelligence and Neuroscience, Article ID:8932896,* vol. 2016, no. 1, 2016.

[20] B. Atiya, A. K. Bakheet, I. T. Abbas, R. Mohd, Abu Bakar, L. S. Lee and B. M. Mansor, "Application of simulated annealing to solve multi-objectives for aggregate production planning," in *Innovations Through Mathematical and Statistical Research: Proceedings of the 2nd International Conference on Mathematical Sciences And Statistics (Icmss2016) 1739, 020086*, Kuala Lumpur, Malaysia, 26–28 January 2016.